



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS QUIXADÁ**  
**CURSO DE GRADUAÇÃO EM REDES DE COMPUTADORES**

**LUCAS MARTINS DA SILVA**

**ANALISANDO O EFEITO DE CARGAS DE TRABALHO NO AMBIENTE VIRTUAL  
DE APRENDIZAGEM MOODLE COM O JMETER NA AWS**

**QUIXADÁ**  
**2025**

LUCAS MARTINS DA SILVA

ANALISANDO O EFEITO DE CARGAS DE TRABALHO NO AMBIENTE VIRTUAL DE  
APRENDIZAGEM MOODLE COM O JMETER NA AWS

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Redes de computadores  
do Campus Quixadá da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de tecnólogo em Redes de computadores.

Orientador: Prof. Dr. Emanuel Ferreira  
Coutinho.

QUIXADÁ

2025

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

---

S581a Silva, Lucas Martins da.

Analisando o efeito de cargas de trabalho no ambiente virtual de aprendizagem Moodle com o JMeter na AWS / Lucas Martins da Silva. – 2025.  
94 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Redes de Computadores, Quixadá, 2025.

Orientação: Prof. Dr. Emanuel Ferreira Coutinho.

1. Moodle. 2. AWS. 3. Apache JMeter. 4. Avaliação de desempenho. 5. Computação em nuvem. I. Título.  
CDD 004.6

---

LUCAS MARTINS DA SILVA

ANALISANDO O EFEITO DE CARGAS DE TRABALHO NO AMBIENTE VIRTUAL DE  
APRENDIZAGEM MOODLE COM O JMETER NA AWS

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Redes de computadores  
do Campus Quixadá da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de tecnólogo em Redes de computadores.

Aprovada em: 08/07/2025.

BANCA EXAMINADORA

---

Prof. Dr. Emanuel Ferreira Coutinho (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Antonio Rafael Braga  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Michel Sales Bonfim  
Universidade Federal do Ceará (UFC)

Dedico este trabalho à minha família e aos meus amigos, que sempre acreditaram em mim, apoiando-me em todos os momentos da minha vida e tornando-se a razão pela qual sigo em frente de cabeça erguida.

## **AGRADECIMENTOS**

Agradeço a Deus por me acompanhar em minha jornada até hoje. Graças a Ele, que me dá forças todos os dias, tenho conseguido superar incontáveis barreiras. Sua presença sempre me guiou e aconselhou, mesmo nos piores momentos da minha vida, inspirando-me a nunca desistir.

Agradeço profundamente aos meus pais, Manoel Gomes da Silva e Francisca Maria Martins Rodrigues, por tudo o que fizeram por mim em prol da minha educação. Sempre presentes, ajudaram-me de todas as formas possíveis e me incentivaram a persistir, independentemente dos desafios que surgissem em meu caminho.

Agradeço ao meu orientador, Prof. Dr. Emanuel Coutinho, por sua excelente orientação, conselhos, correções, ideias e alternativas que me apresentou, além da paciência que teve comigo durante todo esse processo.

Aos colegas que conquistei durante minha vida acadêmica, que estiveram ao meu lado nos momentos felizes e difíceis, oferecendo apoio incondicional: a presença de cada um de vocês em minha vida é algo que levarei comigo para sempre.

"Já tentou, já falhou, não importa. Tente de novo, falhe de novo, falhe melhor." (Samuel Beckett.)

## RESUMO

Logo após o surgimento da internet na década de 1990, começou o avanço de novas tecnologias para o benefício humano. Entre elas, surgiram os Ambientes Virtuais de Aprendizagem (AVA), visando ensinar alunos por meio da internet. Contudo, esses ambientes demandam muitos recursos para apresentar resultados satisfatórios. Por isso, uma alternativa amplamente discutida foi o uso da computação em nuvem, devido à sua flexibilidade, gerenciamento de recursos e baixo custo. No entanto, apenas a computação em nuvem não resolve todas as questões relacionadas ao uso eficiente dos recursos e à manutenção do funcionamento adequado. Para isso, avaliações de desempenho são necessárias para determinar limites e permitir correções futuras. Neste contexto, este trabalho analisará os efeitos de cargas de trabalho no Ambiente Virtual de Aprendizagem (AVA) Moodle, hospedado na plataforma de nuvem Amazon Web Services (AWS), utilizando a ferramenta Apache JMeter para testes de desempenho. O objetivo principal é avaliar a capacidade do Moodle em diferentes cenários de carga, identificando gargalos e determinando a configuração ideal para suportar demandas variáveis. Foram criadas três instâncias (M7i-flex, M6a e M5n) no serviço EC2, que possuem propriedades alinhadas à proposta do trabalho, onde serão hospedados os servidores do Moodle. Em seguida, foram selecionados os testes necessários para avaliar o desempenho da aplicação, como testes de estresse, carga e fumaça, além do planejamento de cenários que simulam atividades como login, navegação e interação em fóruns. A carga de trabalho foi dividida em cinco testes (entre 150 e 3000 threads), executados pelo Apache JMeter, enquanto o desempenho foi monitorado pelo serviço CloudWatch, utilizando métricas como utilização de CPU, latência e taxa de erros. Os resultados revelaram que o servidor M7i-flex apresentou o melhor desempenho, suportando até 500 threads simultâneas sem falhas críticas. Cargas superiores a essa geraram gargalos de desempenho e até indisponibilidade temporária. Já os servidores M6a e M5n mostraram menor resiliência, com falhas frequentes e maior sensibilidade a sobrecargas. A latência aumentou proporcionalmente à carga, especialmente em testes acima de 1000 threads. O estudo demonstrou que a escolha da instância AWS impacta significativamente o desempenho do Moodle. Configurações adequadas podem melhorar a escalabilidade e a disponibilidade, essenciais para períodos de alta demanda acadêmica.

**Palavras-chave:** Moodle; AWS; Apache JMeter; avaliação de desempenho; computação em nuvem.



## ABSTRACT

Shortly after the emergence of the internet in the 1990s, advancements in new technologies for human benefit began. Among these, Virtual Learning Environments (VLEs) emerged, aiming to teach students through the internet. However, these environments require significant resources to deliver satisfactory results. For this reason, a widely discussed alternative has been the use of cloud computing due to its flexibility, resource management, and low cost. Nevertheless, cloud computing alone does not address all issues related to efficient resource usage and maintaining proper functionality. Therefore, performance evaluations are necessary to establish limits and enable future improvements. In this context, this study will analyze the effects of workload on the Moodle Virtual Learning Environment (VLE), hosted on the Amazon Web Services (AWS) cloud platform, using the Apache JMeter tool for performance testing. The main objective is to evaluate Moodle's capacity under different load scenarios, identifying bottlenecks and determining the optimal configuration to handle variable demands. Three instances (M7i-flex, M6a, and M5n) were created on the EC2 service, with properties aligned with the study's objectives, where the Moodle servers will be hosted. Next, the necessary performance tests were selected, including stress tests, load tests, and smoke tests, along with the planning of test scenarios simulating activities such as login, navigation, and forum interactions. The workload was divided into five tests (ranging from 150 to 3000 threads), executed by Apache JMeter, while performance was monitored using the CloudWatch service, with selected metrics such as CPU usage, latency, and error rates. The results revealed that the M7i-flex server performed the best, handling up to 500 simultaneous threads without critical failures. Higher loads led to performance bottlenecks and even temporary unavailability. Meanwhile, the M6a and M5n servers showed lower resilience, with frequent failures and greater sensitivity to overloads. Latency increased proportionally with the load, particularly in tests exceeding 1000 threads. The study demonstrated that the choice of AWS instance significantly impacts Moodle's performance. Proper configurations can improve scalability and availability, which are essential during periods of high academic demand.

**Keywords:** Moodle; AWS; Apache JMeter; performance evaluation; cloud computing.

## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 – Quatro eixos de um AVA. . . . .  | 21 |
| Figura 2 – Tela do fórum do Moodle 1.0. . . . .   | 23 |
| Figura 3 – Ambiente de computação em nuvem. . . . .   | 26 |
| Figura 4 – Modelo de serviços da Nuvem. . . . .   | 28 |
| Figura 5 – Modelo de Implatação de serviços da Nuvem. . . . .                                   | 29 |
| Figura 6 – Modelo de nuvem privada. . . . .   | 29 |
| Figura 7 – Modelo de nuvem publica. . . . .   | 30 |
| Figura 8 – Modelo de nuvem comunidade. . . . .  | 30 |
| Figura 9 – Modelo de nuvem hibrida. . . . .   | 31 |
| Figura 10 – Mapa da zonas e regiões da AWS. . . . .   | 32 |
| Figura 11 – Categoria de serviços da AWS. . . . .   | 33 |
| Figura 12 – Criação de instância no serviço EC2 da AWS. . . . .                                 | 34 |
| Figura 13 – Catálogo de métricas da Amazon CloudWatch. . . . .                                  | 35 |
| Figura 14 – Tecnicas de avaliação de desempenho. . . . .  | 37 |
| Figura 15 – Exemplo de teste de carga no Apache JMeter. . . . .                                 | 39 |
| Figura 16 – Infraestrutura montada para os experimentos. . . . .                                | 41 |
| Figura 17 – Metodologia de avaliação de desempenho. . . . .                                     | 42 |
| Figura 18 – Arquitetura de software dos LMSs testados. . . . .                                  | 44 |
| Figura 19 – Arquitetura do Cluster Virtual Moodle. . . . .                                      | 46 |
| Figura 20 – Procedimentos para a execução do trabalho. . . . .                                  | 49 |
| Figura 21 – Configuração das instancias da AWS. . . . .   | 53 |
| Figura 22 – Píanel de metrcas do servidor do Moodle. . . . .                                    | 54 |
| Figura 23 – Tela de execução do plugin de benchmark do Moodle. . . . .                          | 55 |
| Figura 24 – Quadro de especificações de criação de curso de teste. . . . .                      | 57 |
| Figura 25 – Quadro de especificações de criação de plano de teste. . . . .                      | 58 |
| Figura 26 – Plano de teste do Moodle no Apache JMeter. . . . .                                  | 58 |
| Figura 27 – Arquitetura de execução do projeto. . . . .   | 59 |
| Figura 28 – Configuração dos planos de teste do Moodle. . . . .                                 | 60 |
| Figura 29 – Grafico de requisicoes por etapas - primeira execução do primeiro servidor. . . . . | 61 |
| Figura 30 – Grafico de erros por etapas - primeira execução do primeiro servidor. . . . .       | 62 |

|  |    |
|--|----|
| Figura 31 – Grafico referente as metricas analisadas do teste do Moodle - primeira execu-<br>ção do primeiro servidor. . . . . | 62 |
| Figura 32 – Grafico de requisicoes por etapas - segunda execução do primeiro servidor.   | 63 |
| Figura 33 – Grafico de erros por etapas - segundo execução do primeiro servidor. . . . .                                       | 64 |
| Figura 34 – Grafico referente as metricas analisadas do teste do Moodle - segunda execu-<br>ção do primeiro servidor. . . . .  | 64 |
| Figura 35 – Grafico de requisicoes por etapas - terceira execução do primeiro servidor. .                                      | 65 |
| Figura 36 – Grafico de erros por etapas - terceira execução do primeiro servidor. . . . .                                      | 65 |
| Figura 37 – Grafico referente as metricas analisadas do teste do Moodle - terceira execu-<br>ção do primeiro servidor. . . . . | 66 |
| Figura 38 – Grafico de requisicoes por etapas - quarta execução do primeiro servidor. .  | 66 |
| Figura 39 – Grafico de erros por etapas - quarta execução do primeiro servidor. . . . .  | 67 |
| Figura 40 – Grafico referente as metricas analisadas do teste do Moodle - quarta execução<br>do primeiro servidor. . . . .     | 67 |
| Figura 41 – Grafico de requisicoes por etapas - quinta execução do primeiro servidor. .  | 68 |
| Figura 42 – Grafico de erros por etapas - quinta execução do primeiro servidor. . . . .  | 69 |
| Figura 43 – Grafico referente as metricas analisadas do teste do Moodle - quinta execução<br>do primeiro servidor. . . . .     | 69 |
| Figura 44 – Grafico de requisicoes por etapas - primeira execução do segundo servidor.   | 70 |
| Figura 45 – Grafico de erros por etapas - primeira execução do segundo servidor. . . . .                                       | 70 |
| Figura 46 – Grafico referente as metricas analisadas do teste do Moodle - primeira execu-<br>ção do segundo servidor. . . . .  | 71 |
| Figura 47 – Grafico referente as metricas analisadas do teste do Moodle - segunda execu-<br>ção do segundo servidor. . . . .   | 72 |
| Figura 48 – Grafico de requisicoes por etapas - terceira execução do segundo servidor. .                                       | 73 |
| Figura 49 – Grafico de erros por etapas - terceira execução do segundo servidor. . . . .                                       | 73 |
| Figura 50 – Grafico referente as metricas analisadas do teste do Moodle - terceira execu-<br>ção do segundo servidor. . . . .  | 74 |
| Figura 51 – Grafico de requisicoes por etapas - quarta execução do segundo servidor. .   | 74 |
| Figura 52 – Grafico de erros por etapas - quarta execução do segundo servidor. . . . .   | 75 |
| Figura 53 – Grafico referente as metricas analisadas do teste do Moodle - quarta execução<br>do segundo servidor. . . . .      | 75 |

|   |    |
|---|----|
| Figura 54 – Grafico de requisições por etapas - quinta execução do segundo servidor. . . . .                              | 76 |
| Figura 55 – Grafico de erros por etapas - quinta execução do segundo servidor. . . . .                                    | 76 |
| Figura 56 – Grafico referente as metricas analisadas do teste do Moodle - quinta execução do segundo servidor. . . . .    | 77 |
| Figura 57 – Grafico de requisições por etapas - primeira execução do terceiro servidor. . . . .                           | 78 |
| Figura 58 – Grafico de erros por etapas - primeira execução do terceiro servidor. . . . .                                 | 78 |
| Figura 59 – Grafico referente as metricas analisadas do teste do Moodle - primeira execução do terceiro servidor. . . . . | 79 |
| Figura 60 – Grafico referente as metricas analisadas do teste do Moodle - segunda execução do terceiro servidor. . . . .  | 79 |
| Figura 61 – Grafico de requisições por etapas - terceira execução do terceiro servidor. . . . .                           | 80 |
| Figura 62 – Grafico de erros por etapas - terceira execução do terceiro servidor. . . . .                                 | 81 |
| Figura 63 – Grafico referente as metricas analisadas do teste do Moodle - terceira execução do terceiro servidor. . . . . | 81 |
| Figura 64 – Grafico de requisições por etapas - quarta execução do terceiro servidor. . . . .                             | 82 |
| Figura 65 – Grafico de erros por etapas - quarta execução do terceiro servidor. . . . .                                   | 82 |
| Figura 66 – Grafico referente as metricas analisadas do teste do Moodle - quarta execução do terceiro servidor. . . . .   | 83 |
| Figura 67 – Grafico de requisições por etapas - quinta execução do terceiro servidor. . . . .                             | 83 |
| Figura 68 – Grafico de erros por etapas - quinta execução do terceiro servidor. . . . .                                   | 84 |
| Figura 69 – Grafico referente as metricas analisadas do teste do Moodle - quinta execução do terceiro servidor. . . . .   | 84 |
| Figura 70 – Grafico de indisponibilidade dos servidores do Moodle. . . . .  | 85 |
| Figura 71 – Grafico de comparação de latências dos servidores do Moodle. . . . .  | 87 |

## **LISTA DE TABELAS**

|   |    |
|---|----|
| Tabela 1 – Comparativo entre os trabalhos relacionados e o proposto . . . . . | 48 |
|---|----|

## LISTA DE ABREVIATURAS E SIGLAS

|        |   |
|--------|---|
| AVA    | Ambiente Virtual de Aprendizagem                            |
| AWS    | <i>Amazon Web Services</i>                                  |
| DBMS   | Data Base Management System                                 |
| EC2    | <i>Amazon Elastic Compute Cloud</i>                         |
| LMS    | <i>Learning Management System</i>                           |
| Moodle | <i>Modular Object-Oriented Dynamic Learning Environment</i> |
| NIST   | <i>National Institute of Standards and Technology</i>       |
| TI     | Tecnologia da Informação                                    |

## SUMÁRIO

|              |   |           |
|--------------|---|-----------|
| <b>1</b>     | <b>INTRODUÇÃO . . . . .</b>   | <b>16</b> |
| <b>1.1</b>   | <b>Objetivos . . . . .</b>  | <b>18</b> |
| <b>1.1.1</b> | <b><i>Objetivo Geral . . . . .</i></b>  | <b>19</b> |
| <b>1.1.2</b> | <b><i>Objetivos Específicos . . . . .</i></b>   | <b>19</b> |
| <b>1.2</b>   | <b>Organização do Trabalho . . . . .</b>  | <b>19</b> |
| <b>2</b>     | <b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>  | <b>20</b> |
| <b>2.1</b>   | <b>Ambiente Virtual de Aprendizagem . . . . .</b>   | <b>20</b> |
| <b>2.1.1</b> | <b><i>Moodle . . . . .</i></b>  | <b>22</b> |
| <b>2.2</b>   | <b>Computação em Nuvem . . . . .</b>  | <b>25</b> |
| <b>2.2.1</b> | <b><i>Características essenciais . . . . .</i></b>  | <b>27</b> |
| <b>2.2.2</b> | <b><i>Modelos de serviço . . . . .</i></b>  | <b>27</b> |
| <b>2.2.3</b> | <b><i>Modelos de Implementação na nuvem . . . . .</i></b>   | <b>28</b> |
| <b>2.3</b>   | <b>Amazon Web Services . . . . .</b>  | <b>31</b> |
| <b>2.3.1</b> | <b><i>Amazon Elastic Compute Cloud . . . . .</i></b>  | <b>32</b> |
| <b>2.3.2</b> | <b><i>Amazon CloudWatch . . . . .</i></b>   | <b>34</b> |
| <b>2.4</b>   | <b>Avaliação de Desempenho . . . . .</b>  | <b>35</b> |
| <b>2.4.1</b> | <b><i>Apache JMeter . . . . .</i></b>   | <b>37</b> |
| <b>3</b>     | <b>TRABALHOS RELACIONADOS . . . . .</b>   | <b>40</b> |
| <b>3.1</b>   | <b>Avaliação de desempenho e consumo de energia de um ambiente virtual<br/>de aprendizagem em nuvens privadas . . . . .</b> | <b>40</b> |
| <b>3.2</b>   | <b>Análise de Desempenho do Ambiente Virtual de Aprendizagem na Nuvem<br/>Privada Apache CloudStack . . . . .</b>           | <b>41</b> |
| <b>3.3</b>   | <b><i>Optimization of LMS for Improving user Response Time . . . . .</i></b>  | <b>42</b> |
| <b>3.4</b>   | <b><i>Choosing the right LMS: A performance evaluation of three open-source<br/>LMS . . . . .</i></b>                       | <b>43</b> |
| <b>3.5</b>   | <b>A Study on Moodle Virtual Cluster in Cloud Computing . . . . .</b>   | <b>45</b> |
| <b>3.6</b>   | <b>Comparação entre os Trabalhos Relacionados. . . . .</b>  | <b>46</b> |
| <b>4</b>     | <b>METODOLOGIA . . . . .</b>  | <b>49</b> |
| <b>4.1</b>   | <b>Estudar o AVA selecionado . . . . .</b>  | <b>49</b> |
| <b>4.2</b>   | <b>Selecionar serviços de computação em nuvem . . . . .</b>   | <b>49</b> |

|       |   |    |
|-------|---|----|
| 4.3   | Selecionar quais testes serão utilizados . . . . .                        | 50 |
| 4.4   | Definição dos cenários de testes . . . . .                                | 50 |
| 4.5   | Identificar um conjunto de métricas para medir o desempenho do Moodle     | 50 |
| 4.6   | Projetar a arquitetura para implementação do Moodle . . . . .             | 51 |
| 4.7   | Instalação e configuração do Moodle na AWS . . . . .                      | 51 |
| 4.8   | Elaboração dos planos de teste . . . . .                                  | 51 |
| 4.9   | Realizar os experimentos . . . . .  | 51 |
| 4.10  | Análise dos resultados e impactos causados a partir dos testes realizados | 52 |
| 5     | PROJETO DE EXECUÇÃO . . . . .   | 53 |
| 5.1   | Configurações realizadas nos serviços da AWS . . . . .                    | 53 |
| 5.2   | Configurações realizadas na aplicação do Moodle . . . . .                 | 55 |
| 5.3   | Estudo dos cenários de teste viáveis para execução . . . . .              | 56 |
| 5.4   | Configuração realizada para criação dos planos de teste . . . . .         | 57 |
| 5.5   | Configuração realizada no Apache JMeter . . . . .                         | 57 |
| 6     | RESULTADOS E DISCUSSÃO . . . . .  | 60 |
| 6.1   | Execução do primeiro servidor do Moodle . . . . .                         | 61 |
| 6.1.1 | <i>Primeira execução</i> . . . . .  | 61 |
| 6.1.2 | <i>Segunda execução</i> . . . . .   | 63 |
| 6.1.3 | <i>Terceira execução</i> . . . . .  | 64 |
| 6.1.4 | <i>Quarta execução</i> . . . . .  | 65 |
| 6.1.5 | <i>Quinta execução</i> . . . . .  | 68 |
| 6.2   | Execução do segundo servidor do Moodle . . . . .                          | 69 |
| 6.2.1 | <i>Primeira execução</i> . . . . .  | 69 |
| 6.2.2 | <i>Segunda execução</i> . . . . .   | 71 |
| 6.2.3 | <i>Terceira execução</i> . . . . .  | 72 |
| 6.2.4 | <i>Quarta execução</i> . . . . .  | 73 |
| 6.2.5 | <i>Quinta execução</i> . . . . .  | 75 |
| 6.3   | Execução do terceiro servidor do Moodle . . . . .                         | 77 |
| 6.3.1 | <i>Primeira execução</i> . . . . .  | 77 |
| 6.3.2 | <i>Segunda execução</i> . . . . .   | 78 |
| 6.3.3 | <i>Terceira execução</i> . . . . .  | 80 |
| 6.3.4 | <i>Quarta execução</i> . . . . .  | 81 |



|       |   |    |
|-------|---|----|
| 6.3.5 | <i>Quinta execução</i> . . . . .                | 83 |
| 6.4   | Discussão . . . . .                             | 85 |
| 7     | <b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . . | 89 |
| 7.1   | Considerações finais . . . . .                  | 89 |
| 7.2   | Limitações do trabalho . . . . .                | 90 |
| 7.3   | Trabalhos futuros . . . . .                     | 91 |
|       | <b>REFERÊNCIAS</b> . . . . .                    | 92 |

## 1 INTRODUÇÃO

Logo depois do surgimento da Internet, nos anos 90, começaram a surgir novos produtos e tecnologias para a maioria das áreas do interesse humano, buscando dessa forma obter o máximo de benefícios possíveis (O’Leary; Ramsden, 2002). Dentre as áreas que se beneficiaram dessas novas tecnologias, podemos citar a educação, com os programas educacionais desenvolvidos para o meio virtual, também conhecidos como Ambiente Virtual de Aprendizagem (AVA), cujo objetivo é apoiar alunos e professores na realização de atividades de ensino através da Internet (O’Leary; Ramsden, 2002).

Nesse contexto, as instituições de ensino se beneficiaram conseguindo criar novas oportunidades para serem utilizadas pelos educadores para incentivar os alunos em seus estudos, além de permitir uma melhor organização e trabalho cooperativo entre os envolvidos (Franciscato *et al.*, 2008).

Os AVAs podem ser usados de muitas maneiras para auxiliar na educação, seja de forma remota ou presencial. Se houver conexão com a Internet, os AVAs permitem o uso de mídias e recursos para proporcionar interação entre as partes: alunos e professores, visando atingir determinados objetivos (Franciscato *et al.*, 2008).

Contudo, para utilizar tais ambientes, é necessário ter uma infraestrutura capaz de suportar uma grande demanda de desempenho no sistema, bem como tecnologias que permitam uma comunicação constante entre os componentes que compõem a infraestrutura (Lima *et al.*, 2021). Uma alternativa para solucionar essa questão pode ser o uso de computação em nuvem. A computação em nuvem é um modelo de serviço que oferece sob demanda recursos e serviços de Tecnologia da Informação (TI) através da Internet, que podem ser providos de forma rápida sem a necessidade de interação com o provedor de serviços, além que o custo de cobrança entregue aos usuários será conforme o consumo de recursos utilizados (Mell; Grance, 2011).

A computação em nuvem oferece vantagens como a possibilidade de acessar os serviços de qualquer lugar, facilidade de gerenciamento de recursos, flexibilidade e redução de custos, além de apresentar uma maior segurança sobre os dados (Machida *et al.*, 2011). A utilização da nuvem na educação traz inúmeras vantagens podendo impulsionar a produtividade dos educadores, pois tanto a equipe pedagógica quanto a administrativa contam com a facilidade de atualizar e consultar informações, além de expandir os canais de comunicação. Esse recurso também permite acesso a aulas, sistemas administrativos, documentos e aplicativos de qualquer lugar e a qualquer momento, otimizando e simplificando a execução de tarefas.

No entanto, apenas o uso da computação em nuvem não resolverá questões como assegurar o desempenho de um sistema AVA. É importante considerar gerenciamento dos recursos utilizados, como memória, armazenamento, banco de dados, segurança da aplicação, dentre outros. Além disso, deve-se tomar cuidado com os requisitos exigidos pelas empresas que desenvolvem esses sistemas para aqueles que desejam utilizá-los, pois são essas recomendações que auxiliam no bom desempenho operacional quando a aplicação está sendo utilizada em produção.

Eventualmente, chegará um ponto em que até os requisitos recomendados pelas desenvolvedoras, como Moodle, Teleduc, Solar, entre outras, não serão suficientes para manter um AVA operacional. Sendo assim, torna-se crucial avaliar a eficiência atual da aplicação, para determinar se as configurações atuais atendem à demanda ou se há algum problema causado por problemas como a versão dos componentes desatualizados, a falta de recursos ou até mesmo um grande número de usuários (Fernandes *et al.*, 2017).

Para realizar uma análise desse tipo, é necessário escolher uma técnica de análise de desempenho que se encaixe nas especificações da aplicação, como medição, modelagem analítica e simulação. Dentre as técnicas apresentadas, a medição pode ser uma boa opção, por ser possível simular o uso da aplicação em ambientes controlados, usando qualquer cenário de teste possível, podendo assim identificar problemas de desempenho que possam surgir. Em ambientes de aprendizagem, é comum ocorrerem problemas em determinados períodos, como a época de matrícula ou as avaliações, quando há um aumento significativo nos acessos feitos pelos alunos, resultando em estresse e sobrecarregamento da aplicação (Silva *et al.*, 2019).

Portanto, para minimizar esse problema, existem diversas alternativas possíveis que podem auxiliar no desenvolvimento de melhorias voltadas ao desempenho. Dentre elas, existe a utilização de ferramentas de carga de trabalho que podem ser aplicadas ao ambiente de simulação. Essas cargas de trabalho geradas podem ser utilizadas para analisar *sites*, *software*, sistemas e aplicações na *web* do mundo real. Ao final dessas simulações, relatórios sobre a performance da aplicação podem ser gerados, permitindo obter algumas conclusões, tais como:

- Quantos usuários simultâneos são necessários para derrubar a aplicação?
- Durante o teste houve algum gargalo de desempenho?
- Quantas requisições a aplicação pode suportar em um determinado período?

Dentre as opções possíveis de ferramentas voltadas para realização de testes de carga de trabalho, o Apache JMeter da Apache Software Foundation ganha destaque, por ser uma

ferramenta com uma extensa documentação de cada uma das suas funcionalidades, permitindo uma customização para realização dos seus testes através do uso de plugins, além de ser uma ferramenta gratuita bastante utilizada no mundo quando se trata de avaliação de desempenho.

Embora seja necessário avaliar o desempenho em ambientes virtuais de aprendizagem, uma vez que esses ambientes têm se tornado um importante instrumento para a educação, permitindo a comunicação entre alunos e professores, é preciso conhecer sua verdadeira capacidade Silva *et al.* (2019). No entanto, ao analisar a literatura sobre AVAs, nota-se uma lacuna no que diz respeito à avaliação de desempenho, uma vez que há poucos trabalhos que cumprem essa tarefa, sobretudo quando se trata de uma abordagem que utiliza ambientes de computação em nuvem (Wiechork; Charao, 2020)(Silva *et al.*, 2019)(Lima *et al.*, 2021).

Dessa forma, esse trabalho tem como intuito realizar uma análise de desempenho através do uso de cargas de trabalho para analisar os efeitos gerados em um ambiente virtual de aprendizagem que estará hospedado em um ambiente de computação em nuvem. O AVA selecionado para esse trabalho será o *Modular Object-Oriented Dynamic Learning Environment* (Moodle)<sup>1</sup>, um software livre de apoio à aprendizagem mundialmente utilizado em diversas instituições de ensino.

Para isso, serão utilizados os serviços da *Amazon Web Services* (AWS), como o *Amazon Elastic Compute Cloud* (EC2)<sup>2</sup>, que hospedará a aplicação do Moodle na nuvem, e a ferramenta de teste de carga selecionada para o trabalho, o Apache JMeter<sup>3</sup>, que gerará os testes e os executará em seus respectivos cenários de teste, e o serviço Amazon CloudWatch<sup>4</sup>, que permitirá o uso e o gerenciamento das métricas disponíveis na plataforma de nuvem. Assim conseguindo determinar a qualidade do ambiente Moodle no que pode usar outras instituições de ensino a determinar uma configuração adequada que possa dar suporte ao sistema em diferentes situações e com diversos números de usuários simultâneos.

## 1.1 Objetivos

Nesta seção, serão apresentados o objetivo geral e os objetivos específicos do trabalho.

<sup>1</sup> Moodle — <https://moodle.com/pt/>

<sup>2</sup> Amazon EC2 — <https://aws.amazon.com/pt/ec2/>

<sup>3</sup> Apache JMeter — <https://jmeter.apache.org/>

<sup>4</sup> CloudWatch — <https://aws.amazon.com/pt/cloudwatch/>

### ***1.1.1 Objetivo Geral***

O objetivo geral deste trabalho é avaliar o desempenho do ambiente de aprendizagem Moodle hospedado na AWS através do uso de diferentes cargas de trabalho.

### ***1.1.2 Objetivos Específicos***

- Elaborar um ambiente funcional do Moodle na AWS.
- Definir um conjunto de cargas de trabalho projetado especialmente para avaliar as funcionalidades do Moodle.
- Conduzir uma análise de desempenho no ambiente Moodle hospedado na nuvem utilizando o conjunto de carga de trabalhos definido a fim de determinar a capacidade do ambiente.

## **1.2 Organização do Trabalho**

O restante deste trabalho está organizado nas seguintes seções. No Capítulo 2 será apresentado os principais conceitos abordados nesse trabalho, sendo eles, ambientes virtuais de aprendizagem, análise de desempenho, computação em nuvem entre outros. No Capítulo 3 são apresentados os trabalhos relacionados ao tema abordado, apresentado as suas ideias e o meio de condução. No Capítulo 4 é representado a metodologia elaborada para execução do trabalho, sendo explicado cada etapa do processo de execução. No Capítulo 5 será apresentado em detalhes, mais técnicos sobre a criação do ambiente do Moodle na plataforma de nuvem da AWS, bem com a criação dos testes, que serão usados para avaliação. No Capítulo 6 é apresentado os resultados obtidos na execução dos testes são demonstrados e discutidos em grandes detalhes; Por fim, no Capítulo 7 apresenta as conclusões, contribuições e trabalhos futuros decorrentes do trabalho realizada.

## 2 FUNDAMENTAÇÃO TEÓRICA

nesta seção serão apresentados os principais conceitos utilizados neste trabalho, começando pelo conceito de AVA, apresentando uma breve explicação sobre as principais características que compõe esse ambiente. Em seguida será apresentado o ambiente Moodle onde será apresentado um pouco sobre a história, principais funções da plataforma. Em seguida, serão apresentados conceitos de computação em nuvem como: história, modelos de serviço, tipos de nuvem utilizada atualmente. Logo após uma apresentação sobre o serviço de nuvem da Amazon conhecido como AWS é os serviços utilizados nesse trabalho. E por último será apresentado o conceito de avaliação de desempenho, quais são os tipos de avaliação que existe, além da ferramenta de teste de carga utilizada no trabalho, o Apache JMeter.

### 2.1 Ambiente Virtual de Aprendizagem

Os ambientes virtuais de aprendizagem, também conhecidos como *Learning Management System* (LMS), começaram a ser criados durante a implementação dos programas educacionais através do uso da Internet nos anos 90, quando se tornaram populares no meio educacional (Beluce, 2012). Esses ambientes podem ser definidos através da comunicação entre os alunos e professores de forma síncrona ou assíncrona, para o desenvolvimento da educação através do uso dos recursos disponíveis no ambiente (Meyer, 2022).

Dessa forma, esses ambientes podem reunir em espaços virtuais diversos recursos eletrônicos, tais como: fóruns, bibliotecas, *chats* de bate-papo, questionários, dentre outros recursos que possam ser usados para atingir os objetivos dos educadores de aprimorar o processo de ensino e aprendizagem dos alunos (Rosemann *et al.*, 2014). Além dos recursos que estão presentes nesses ambientes, existem funcionalidades que podem ser usadas para organizar e administrar esses ambientes, que podem ser apresentados sob a forma de ferramentas, que podem variar conforme a necessidade requisitada do ambiente (Gonzalez, 2005).

De acordo com Pereira *et al.* (2007), se forem utilizados corretamente, esses recursos e ferramentas, que estão sendo constantemente desenvolvidos para a educação, podem ser aproveitados para aprimorar a aprendizagem. Segundo a Universidade Federal de Santa Catarina, com a experiência na utilização de AVA, esses recursos tecnológicos são normalmente divididos em quatro eixos, como mostra a Figura 1.

No trabalho de Pereira *et al.* (2007) é descrito cada um desses eixos como uma peça

Figura 1 – Quatro eixos de um AVA.



Fonte: (Pereira *et al.*, 2007)

fundamental para o funcionamento de um AVA, sendo necessário conceituar cada um deles separadamente:

- Informação e documentação: Apresenta todos os tipos de dados sobre a instituição, como informações sobre cursos, conteúdos abordados por eles, além de dados sobre os alunos para os professores para disponibilizar conteúdo didático.
- Comunicação: Permite criar um *link* de comunicação entre alunos e professores de forma síncrona ou assíncrona.
- Gerenciamento pedagógico e administrativo: Possui acesso a dados sobre o desempenho acadêmico dos alunos, bem como a serviços internos da instituição, tais como a secretaria acadêmica e entre serviços.
- Produção: Proporciona a criação de atividades para os alunos, além de resolver problemas internos que possam surgir no ambiente.

É importante destacar que um AVA pode ser composto por todos os eixos mencionados anteriormente, mas não é obrigatório ter eles. Se não houver um, esses ambientes não deixaram de ser um AVA, pois o que determinará um desses ambientes será a qualidade e a aplicabilidade necessária para atingir o objetivo que foi determinado nesse ambiente (Pereira *et al.*, 2007).

Os ambientes virtuais de aprendizagem podem ser utilizados como ferramenta de ensino de várias maneiras diferentes. As maneiras mais comuns são como apoio à educação a

distância, presencial e semi-presencial, também chamado de *blended*. Como meio de ensino à distância será a ponte de comunicação dos alunos e professores para a resolução de dúvidas, além de ser um local de armazenamento para o conteúdo apresentado aos alunos (Fiorio *et al.*, 2011).

Os ambientes em questão apresentam diversos benefícios que as instituições que os empregam utilizam, tais como: facilidade de comunicação, aplicação de atividades e provas, consulta de material sempre que desejar, além de permitir que os alunos aprendam de forma independente e em qualquer lugar (Meyer, 2022). Devido a essas vantagens, existem atualmente diversos AVAs disponíveis no mercado, alguns deles de acesso restrito onde é necessário pagar para utilizar, são ambientes gratuitos acessados pela Internet. Alguns exemplos desses ambientes que podem ser citados Moodle, BlackBoard<sup>1</sup>, SOLAR<sup>2</sup> entre outros (Meyer, 2022). Neste trabalho, optou-se por utilizar o ambiente Moodle como AVA para a avaliação do desempenho em cenários de testes criados de forma específica para essa plataforma. Esses cenários foram desenvolvidos com base em um conjunto de carga de trabalho previamente elaborado.

### 2.1.1 Moodle

O Moodle é uma plataforma *open-source* criada pelo australiano Martin Dougiamas em 2001, logo depois da experiência frustrante com a ferramenta WebCT da Curtin University onde ele resolveu criar a sua própria ferramenta de ensino (Moodle.ORG, 2023). A ferramenta foi criada para oferecer aos educadores, administradores e alunos um único sistema robusto, seguro e integrado para criar ambientes de aprendizagem personalizados. Além disso, oferece diversos recursos para apoiar a aprendizagem, como fóruns, salas de conversação, questionários, matérias, entre outros (Moodle.ORG, 2023).

A plataforma Moodle já está presente em mais de 223 países em 179.000 sites com aproximadamente 316.000.000 usuários em todo mundo, sendo um dos principais líderes mundiais na indústria da educação (Moodle.ORG, 2023). A versão 1.0 do Moodle foi lançada em 20 de agosto de 2002, quando o seu criador Martin Dougiamas anunciou em uma publicação no fórum do seu sistema, como mostrado na Figura 2. Nesta versão veio acompanhada de um novo fórum para discursarão de temas, permitindo a tradução do Moodle para diferentes idiomas, além de permitir a criação de temas (Ribeiro *et al.*, 2007).

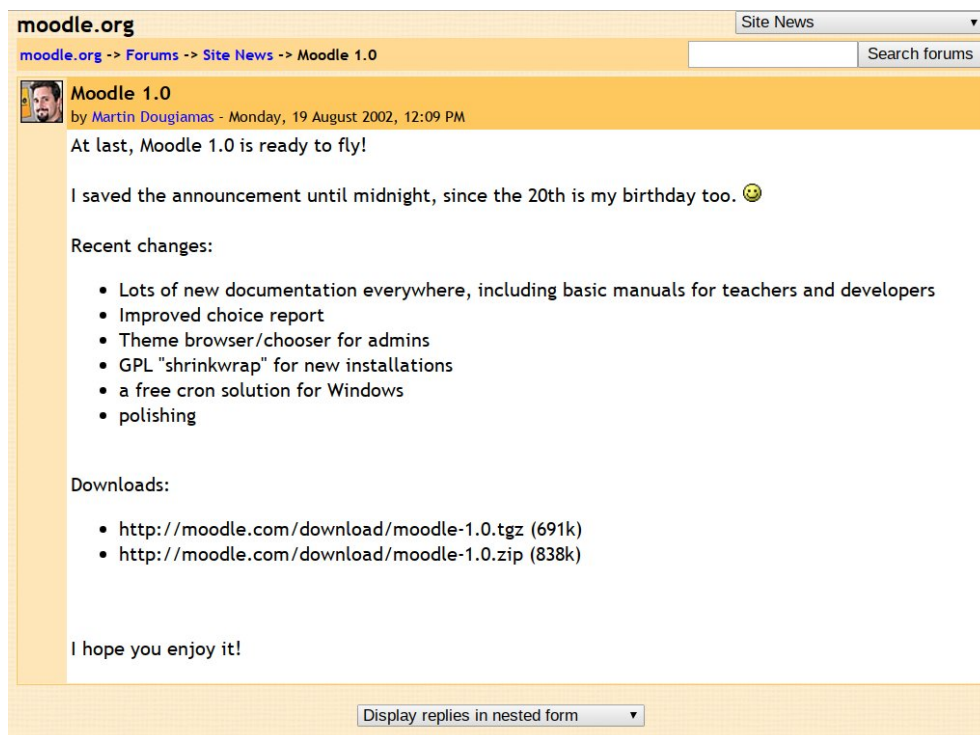
---

<sup>1</sup> BlackBoard — <https://www.blackboard.com/>

<sup>2</sup> SOLAR — <https://solar.virtual.ufc.br/>



Figura 2 – Tela do fórum do Moodle 1.0.



Fonte: (MOODLE, 2017).

Dado o sucesso da primeira versão, o Moodle tem sido constantemente atualizado, melhorando o seu desempenho e trazendo, a cada nova versão, algum recurso novo. Devido a isso, diversas organizações pelo mundo utilizam o Moodle como o seu sistema padrão. Além do fato de ser gratuito e possuir uma grande comunidade que auxilia na correção de erros e discussões sobre possíveis melhorias (Ribeiro *et al.*, 2007). Em 2003, foi fundada a empresa moodle.com, cujo objetivo seria fornecer suporte adicional aos usuários, lidar com as questões comerciais do Moodle, gerenciar o funcionamento e prestar consultorias pelos serviços oferecidos (Moodle.ORG, 2023).

O Moodle, por ser uma ferramenta de software livre, pode encontrado para *download* em praticamente todas as plataformas (Unix, Linux, Windows, Mac OS) desde que a plataforma em questão possa executar PHP e ter suporte a base de dados do tipo SQL. Sendo desenvolvida em conjunto com sua comunidade virtual ativa, que reúne programadores, professores e usuários de diversos lugares do mundo, tem se desenvolvido conforme as necessidades que os seus usuários têm requisitado ao longo do tempo. Dessa forma, o Moodle pode ser moldado conforme a necessidade do usuário, desde uma página única utilizada por um profissional da educação formado até um sistema de gerenciamento de milhares de alunos em uma instituição de ensino.

Para isso, o Moodle tem diversas características, recursos e ferramentas que auxiliam na criação de atividades e na gestão do próprio sistema, proporcionando uma maior interação e

dinamicidade entre os envolvidos no processo. Os professores ou administradores do sistema podem adicionar ferramentas e recursos conforme os seus objetivos pedagógicos pertinentes. Para organizar esses recursos e ferramentas, o Moodle divide-os em quatro grupos: características gerais, recursos administrativos, recursos de desenvolvimento e gerenciamento de cursos. Alguns desses recursos serão apresentados a seguir.

- Administração de site: O Moodle possui um usuário administrador definido no momento da instalação do Moodle. Há uma variedade de *plugins* disponíveis nesse sistema que podem ser usados para personalizar conforme as especificações do administrador. Alguns *plugins* que existem ajustam as cores, fontes, aparência, módulo de atividade e pacotes de idioma.
- Administração do curso: Neste módulo, o professor tem total controle sobre todos os parâmetros de um curso criado no Moodle, podendo assim escolher qual será o formato do curso, tais como semanal, por tópico ou um formato social centrada na discussão. Dessa forma, é possível selecionar quais atividades este curso pode empregar, tais como fóruns, questionários, tarefas, etc. Os cursos criados no Moodle podem ser exportados em um arquivo zip, utilizando a função de *backup*, o qual pode ser restaurado em qualquer servidor Moodle.
- Módulo tarefa: Esse módulo é usado pelos professores visando orientar a execução de atividades. Após a entrega num prazo determinado, o aluno receberá um *feedback* da atividade com uma pontuação que atingiu. Para auxiliar os professores, é necessário criar um formulário único para a avaliação da turna, no qual o aluno deve submeter um arquivo com as suas respostas ou preencher as respostas em um formulário de texto *online*.
- Módulo chat: Esse módulo é usado para realização de conversar em tempo real através de mensagens de texto, além de permitir o uso de figuras, *emojis*, *URLs*, imagens, etc. para permitir discussão sobre temas abordados nas aulas. As mensagens ficam salvas para caso os participantes queiram ver novamente posteriormente, caso o professor considere necessário.
- Módulo fórum: É um módulo que permite aos participantes realizar discussões sobre temas abordados em aulas de maneira assíncrona. Dessa forma, permitindo debates que favoreçam na troca de conhecimento entre os participantes. Este módulo apresenta algumas variações, tais como: fórum reservado para professores, news, fórum para uso geral, fórum com ações limitadas, onde as postagens realizadas têm a foto do autor anexada. Além

disso, o professor pode personalizar o fórum conforme a sua vontade, limitando o tempo de permanência do fórum ou, até mesmo, forçando a inscrição de todos.

- Módulo questionário: Esse módulo é utilizado pelos professores para criar uma base de dados com questões para avaliar o conhecimento dos alunos, permitindo a aplicação de autoavaliação, lista de exercícios, revisão de conteúdo, teste ou avaliações virtuais. As questões podem ser de múltipla escolha, verdadeira ou falsa, correspondência, resposta dissertativa curta com um prazo limitado para responder, além de serem configuradas para a correção automática.
- Módulo recursos: Esse módulo permite que o Moodle tenha suporte a quase todo tipo de conteúdo eletrônico, Word, PowerPoint, flash, vídeo, sons, etc. Esses arquivos podem ser enviados para o servidor onde também são administrados ou caso o usuário queira ele pode usar formulários *web* interno do sistema para criar os seus próprios arquivos.
- Módulo de base de dados: É um módulo que permite criar, atualizar, pesquisar e exibir um conjunto de dados sobre um tema específico que já existe salvo na base de dados do sistema, que pode ser imagens, arquivos, *URLs*, números e texto, entre outras coisas.

## 2.2 Computação em Nuvem

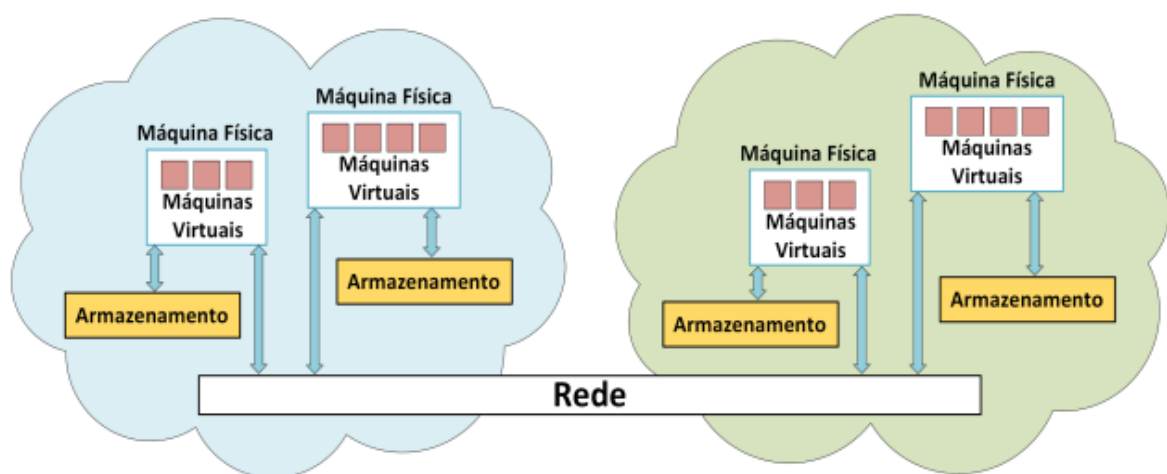
A computação em nuvem, como conhecemos atualmente, surgiu nos anos 1990, quando as companhias de telecomunicações começaram a criar uma rede virtualizada para serem usadas pelos seus clientes, onde ela deveria ser de qualidade e com um baixo custo (blog, 2020). A fim de atingir esse objetivo, várias tecnologias foram combinadas, como: virtualização, computação em grids, computação distribuída, de forma que as empresas começaram a fornecer acesso à infraestrutura de forma compartilhada para os seus clientes da mesma forma que a usavam fisicamente (Fernandes *et al.*, 2017).

Segundo a *National Institute of Standards and Technology* (NIST), a computação em nuvem é um modelo que permite o acesso ubíquo e conveniente a rede, onde um conjunto de recursos, como a rede, armazenamento e os serviços, pode ser alocado e removidos de forma rápida, sem esforço ou nenhuma interação com o provedor do serviço, o que permite que as empresas paguem apenas pelo o que está sendo utilizado, reduzindo os custos de infraestrutura pessoal (Mell; Grance, 2011). Dessa forma, o uso de computação em nuvem oferece vantagens como: facilidade de implantação, alta escalabilidade, redução de custos em equipamentos e manutenção, o que fez com que diversas companhias migrassem os seus sistemas para a nuvem

(Polash *et al.*, 2014).

A infraestrutura de um ambiente de computação em nuvem é composta por centenas ou milhares de computadores físicos de baixo custo em ambientes climatizados em locais fechados, onde eles estão todos conectados através de uma rede, como mostrado na Figura 3. De forma que cada computador pode ter especificações de hardware e software distintas (Soror *et al.*, 2008). Ademais, cada computador terá uma quantidade não especificada de instâncias de máquinas virtuais em execução, com configurações determinada, pela máquina física.

Figura 3 – Ambiente de computação em nuvem.



Fonte: (Coutinho, 2014).

A computação em nuvem está cada vez mais presente nas vidas das pessoas e empresas, o que está aprimorando o desempenho dos serviços prestados, tornando-se necessário a utilização desta nova tecnologia (Arruda, 2016). Essa tecnologia pode ser encontrada em serviços básicos da Internet como armazenamento de imagens, vídeos, entretenimento, sendo oferecido tanto para os usuários quanto para as empresas de forma independente (Moreira *et al.*, 2021).

O objetivo da computação em nuvem é oferecer maiores utilidades a tecnologia do que apenas armazenar dados, isso significa criar novos serviços, flexibilidade e administrar a infraestrutura (Eulálio *et al.*, 2016). Segundo a NIST, no modelo de computação em nuvem, existem cinco características essenciais, três modelos de serviço e quatro modelos de implantação (Mell; Grance, 2011), conforme será apresentado a seguir.

### 2.2.1 Características essenciais

Na definição de computação em nuvem da NIST, existem cinco características essenciais que oferecem vantagens para a tecnologia, onde elas podem juntas definir o conceito de computação em nuvem.

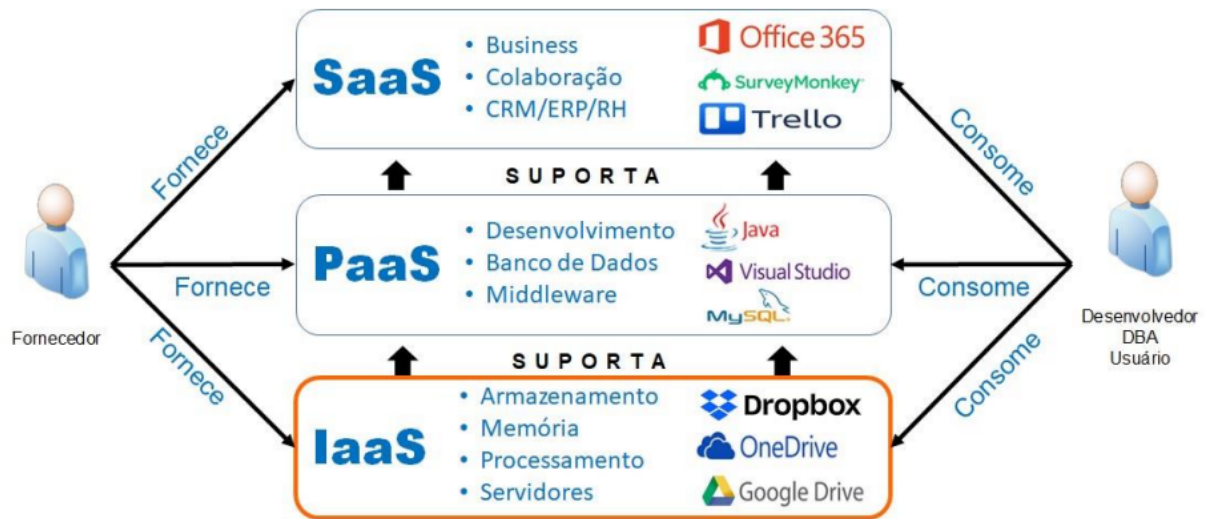
- *Self-service* sob demanda: Permite que o usuário tenha acesso aos recursos computacionais unilateralmente, como a rede e o armazenamento, de forma rápida, sem a necessidade de interação com os provedores de cada recurso alocado.
- Amplo acesso: Os recursos computacionais são acessíveis pela rede por dispositivos padronizados, com acesso à Internet. Tais como dispositivos móveis, computadores, PDAs, etc.
- *Pooling* de recursos: Os serviços são oferecidos aos clientes no modelo multi-inquilino, onde podem ser oferecidos recursos físicos ou virtuais conforme a necessidade do cliente. Além disso, os clientes não possuem informações precisa da localização dos recursos físicos, apenas informações abstratas como país, cidade ou região.
- Elasticidade rápida: Os recursos podem ser adquiridos ou liberados de forma rápida, em alguns casos até mesmo de forma automática. Caso seja preciso, pode-se adquirir mais recursos para atender à demanda. Para os usuários os recursos podem parecer infinitos, podendo ser adquirido qualquer quantidade no momento que quiser.
- Serviço medido: Os sistemas que oferecem serviços de nuvem podem monitorar, reportar e até mesmo otimizar os recursos de forma automática através de uma capacidade de medição que permite uma análise do nível de recursos, como processamento, armazenamento, latência da rede e a quantidade de clientes ativos. Este monitoramento é realizado de forma transparente tanto para o provedor do serviço quanto para o cliente, seguindo um modelo de medição chamado de *pay-as-you-go*, no qual apenas é cobrado o que foi realmente utilizado.

### 2.2.2 Modelos de serviço

O ambiente de computação em nuvem é composto principalmente por três modelos de serviços utilizados para implantação, onde eles são apresentados na Figura 4 (Mell; Grance, 2011) (Services, 2023).

- *Software como um Serviço (Software as a Service — SaaS)*: O modelo SaaS oferece um

Figura 4 – Modelo de serviços da Nuvem.



Fonte: (Fenner, 2019).

conjunto de sistemas de *software*, onde cada um possui um propósito específico, é são disponibilizados para os clientes por meio de interfaces *thin client* como, por exemplo, navegadores web. Nesse modelo, o cliente não tem controle sobre a infraestrutura, como rede, armazenamento, sistema ou características exclusivas do *software*, apenas funções específicas para o uso do cliente.

- *Plataforma como um Serviço (Platform as a Service — PaaS)*: O modelo PaaS oferece ao cliente uma infraestrutura de alto nível, incluindo linguagens de programação, sistemas operacionais e ambiente de desenvolvimento, para poderem ser projetados sistemas de *software*. Da mesma forma que no SaaS, o cliente não tem acesso total a infraestrutura, mas tem acesso às aplicações implementadas e, possivelmente, as configurações de aplicações implementadas na infraestrutura.
- *Infraestrutura como um Serviço (Infrastructure as a Service — IaaS)*: O modelo IaaS fornece aos clientes recursos como rede, armazenamento, servidores e outros recursos computacionais fundamentais para a criação de um ambiente de sistema sob demanda, que pode ser composto por *software* ou aplicações. O cliente não tem acesso à infraestrutura da nuvem, mas tem acesso aos recursos computacionais, podendo selecionar quais componentes de rede desejam usar como *firewalls*.

### 2.2.3 Modelos de Implementação na nuvem

Os principais modelos de implementação na nuvem podem ser divididos em quatro tipos conforme mostrado na Figura 5.

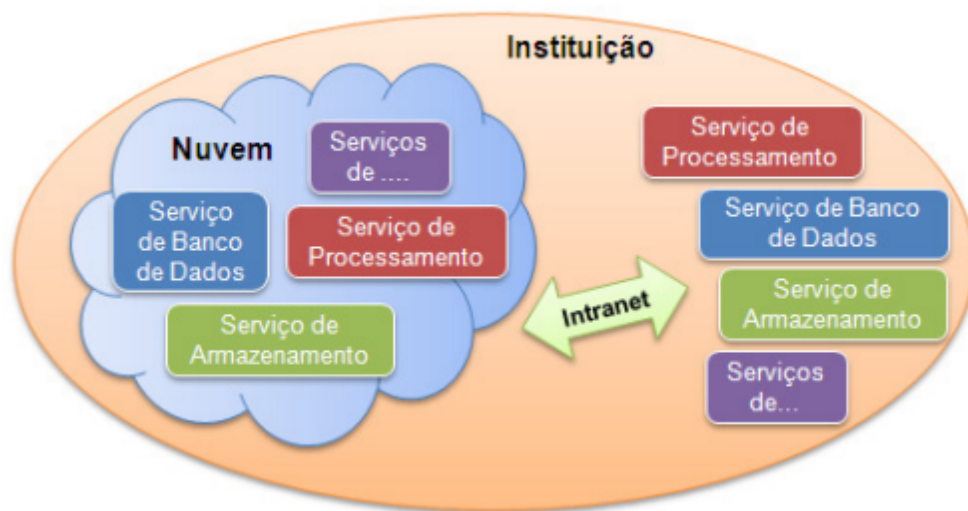
Figura 5 – Modelo de Implatação de serviços da Nuvem.



Fonte: (Coutinho, 2014).

- Nuvem Privada: A infraestrutura de nuvem privada, é usada somente por uma instituição como mostrado na Figura 6, onde ela é administrada pela própria empresa ou por terceiros. Os serviços oferecidos pela empresa estão localizados nas dependências, permitindo, dessa forma, administrar de forma eficiente os recursos locais para os próprios usuários da empresa. Além disso, qualquer acesso aos recursos fora do local não é permitido.

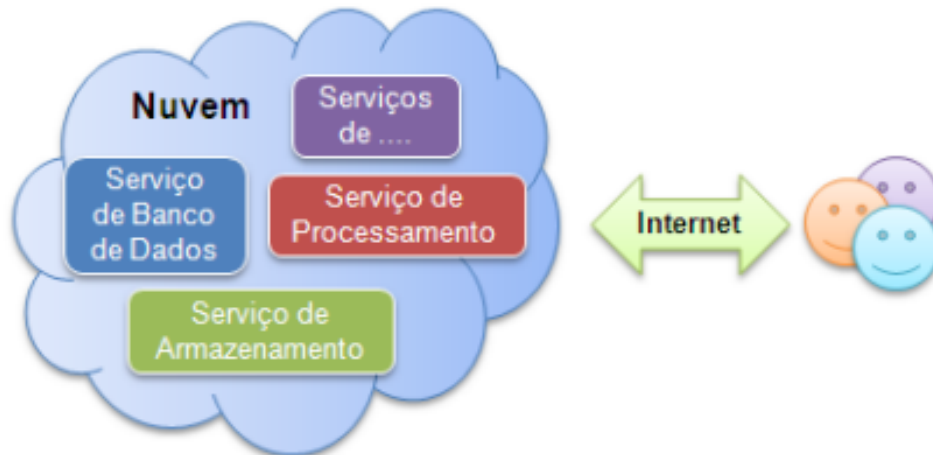
Figura 6 – Modelo de nuvem privada.



Fonte: (Borges *et al.*, 2011).

- Nuvem Pública: A infraestrutura de nuvem pública pertence a empresas especializadas nessa área, como Amazon, Google e Microsoft, que vendem os seus serviços ao público através da Internet, como mostra a Figura 7. Os serviços oferecidos por esse modelo seguem o conceito de pagar apenas pelo que foi realmente utilizado, assim como os clientes não precisam se preocupar com a complexidade dos serviços, uma vez que isso é responsabilidade da organização.

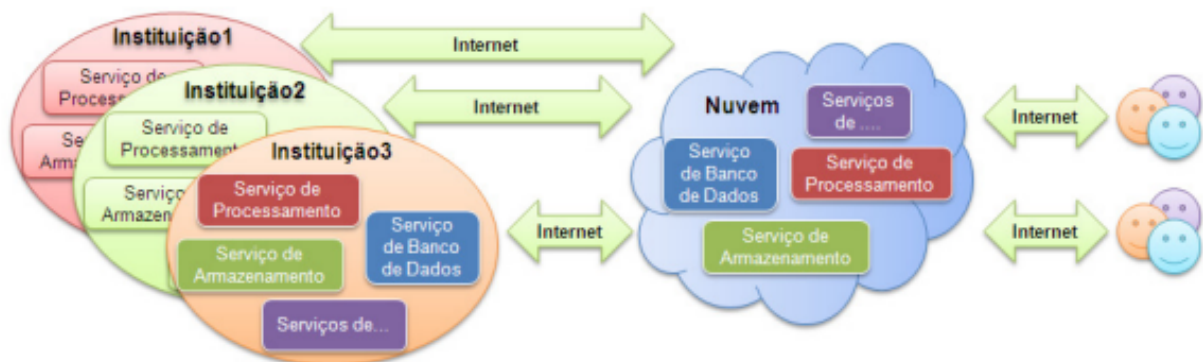
Figura 7 – Modelo de nuvem publica.



Fonte: (Borges *et al.*, 2011).

- Nuvem Comunidade: A infraestrutura da nuvem é compartilhada com um grupo de organizações, com objetivos em comum, como propósito, missão, políticas e requisitos de sistema, como mostrado na Figura 8. Sendo assim, podendo ser gerenciada por uma ou mais organizações, ou até mesmo por terceiros, que pode estar presentes dentro ou fora da organização.

Figura 8 – Modelo de nuvem comunidade.

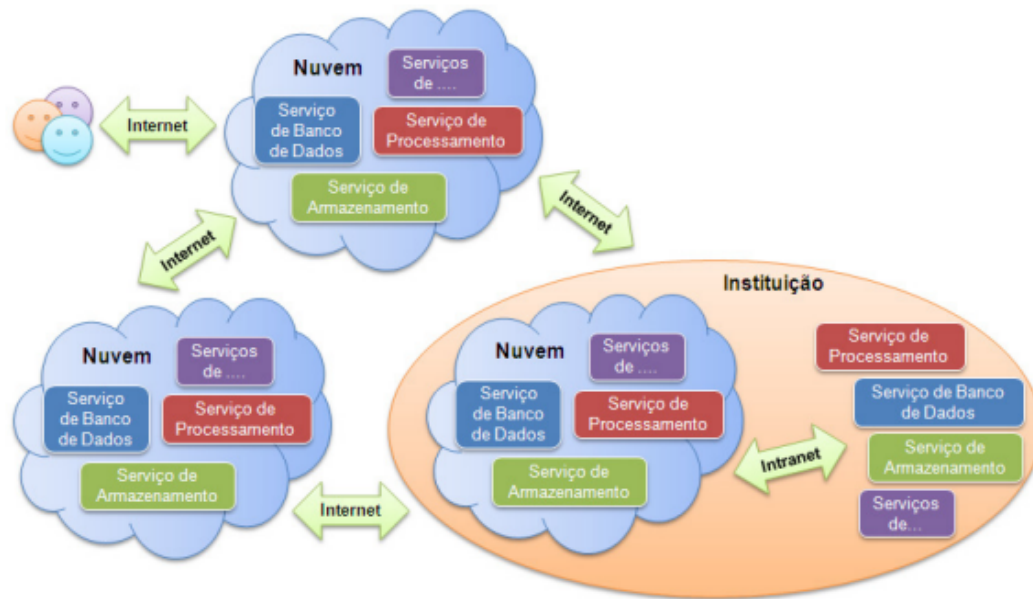


Fonte: (Borges *et al.*, 2011).

- Nuvem Híbrida: A infraestrutura básica desse modelo de nuvem é composta por duas ou mais nuvens, que podem ser públicas, privadas ou comunitárias, conforme demonstrado na Figura 9. Sendo que essas nuvens permanecem como entidades independentes, mas são unidas por uma tecnologia padronizada que existe em comum entre elas, como balanceadores de carga, permitindo a portabilidade dos dados entre as nuvens.



Figura 9 – Modelo de nuvem híbrida.



Fonte: (Borges *et al.*, 2011).

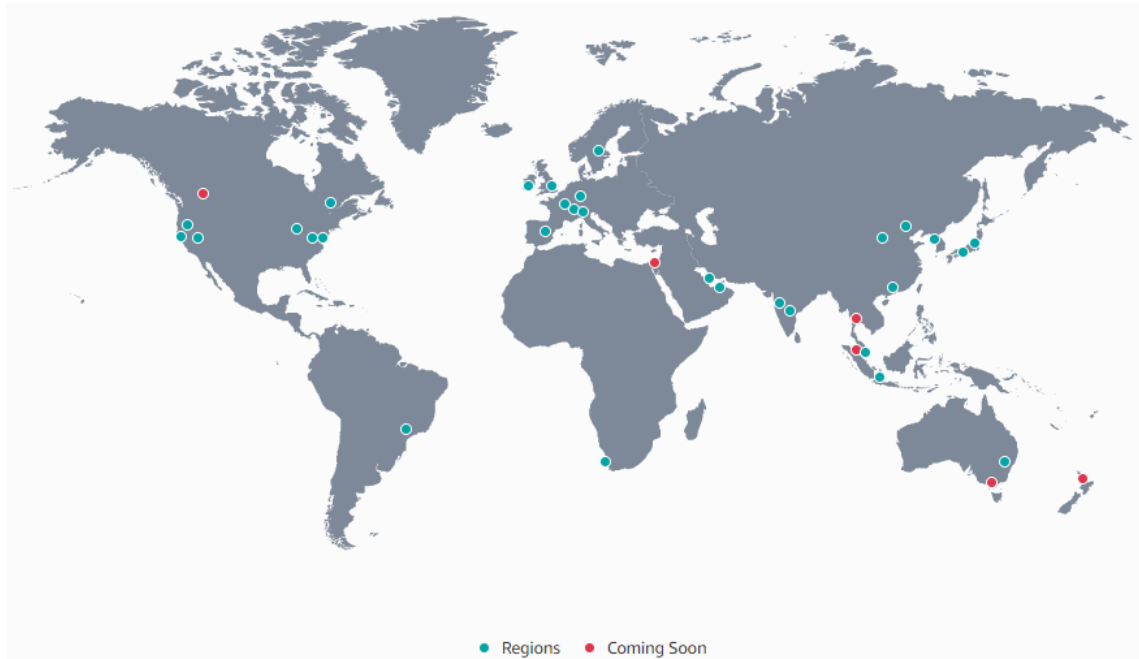
### 2.3 Amazon Web Services

A *Amazon Web Services* é a empresa líder mundial no campo da computação em nuvem, devido às suas vantagens de disponibilidade, desempenho e acessibilidade. Além disso, possui como clientes grandes companhias de diversas áreas, como a Netflix, a Coca-Cola, o McDonald's e a Ubisoft. Além dessas, há mais de 2062 empresas que utilizam os serviços da AWS. Foi lançada no mercado em 2006 pela companhia multinacional de tecnologia norte-americana Amazon, que fornece para as companhias infraestruturas de TI como um serviço disponibilizado para empresas através da *Internet* (Mathew; Varia, 2014).

A AWS oferece mais de 200 serviços completos de *data centers* em todo o mundo, o que permite ter uma grande comunidade ativa com os seus milhares de clientes, que estão espalhados por mais de 190 países pelo mundo. Os *data centers* que oferecerão serviços para empresas são divididos em zonas chamadas de AZs (*availability zones*), onde a empresa que contratará os serviços da AWS deve escolher qual a melhor zona que se adéque às suas necessidade, tendo em mente que cada zona terá uma cobrança diferente. A AWS tem atualmente cerca de 86 zonas de disponibilidade divididas em 26 regiões pelo mundo, com intenção de criar novas zonas em mais regiões pelo mundo, conforme apresentado na Figura 10 (Services, 2023).

A AWS oferece mais de 18.000 produtos, que estão divididos em diversas categorias, como soluções para computação, IoT, jogos e desenvolvimento. Como mostrado na Figura 11, alguns desses serviços costumam não exigir que o cliente tenha um conhecimento aprofundado

Figura 10 – Mapa da zonas e regiões da AWS.



Fonte: (Services, 2023).

sobre como gerenciar o funcionamento, por serem intuitivos e trabalham geralmente de forma automática, como no serviço de criação de instância, o EC2. Já outros exigem um conhecimento prévio para trabalhar, como os serviços que oferecem ferramentas de desenvolvimento que trabalham com linguagens de programação como Java, .NET, PHP, Node.js, Python, Ruby, Go (Đorđević *et al.*, 2014). Por esses motivos apresentados que a plataforma de nuvem da Amazon foi selecionada por ter uma grande gama de serviços é por não ser necessário um conhecimento aprofundado para utilizar os serviços disponíveis.

### 2.3.1 *Amazon Elastic Compute Cloud*

O serviço *Amazon Elastic Compute Cloud* ou EC2 é o serviço central da plataforma de nuvem AWS, que está no nível de serviço IaaS, oferecendo aos clientes infraestrutura, permitindo que os clientes aluguem computadores virtuais, também chamados de instâncias pela própria AWS, que podem ser usados para hospedar suas aplicações. Um usuário da AWS que usa o serviço EC2 pode aumentar ou diminuir os recursos computacionais da instância conforme a necessidade em poucos minutos, usando a interface da Amazon EC2. Os recursos computacionais utilizados nas instâncias do serviço EC2 podem ser de natureza variada, com configurações distintas provenientes de locais diferentes em diferentes *data center*, onde os clientes podem selecionar os recursos, mas, além disso, podem fornecer redundância em caso de

Figura 11 – Categoria de serviços da AWS.



Fonte: (Services, 2023).

problemas.

O serviço EC2 foi desenvolvido pelo engenheiro Chris Pinkham e Benjamin Black, em 2006, quando a Amazon anunciou um teste beta para um público limitado do serviço EC2. No ano seguinte, a Amazon adicionou dois novos tipos de instâncias para o serviço, os tipos (*Large e Extra-Large*). Já na edição de 2008, foram introduzidos os endereços IP estáticos, zonas de disponibilidade e *kernels* selecionáveis pelo usuário. Além disso, também foram introduzidos dois novos tipos de instâncias: *High-CPU Medium e High-CPU Extra Large*. O serviço Amazon EC2 iniciou a sua produção em outubro do mesmo ano, quando foi atualizado com o acordo de nível de serviço específico para o EC2, além da introdução do serviço de armazenamento persistente Elastic Block Store (EBS).

O serviço Amazon EC2 funciona em volta do seu console de gerenciamento, onde é mostrado as informações das instâncias criadas pelos clientes, tais como: armazenamento, rede, sistema operacional, grupo de segurança, além da informação sobre o *status* da instância. Se o cliente não tiver instancias criadas, é possível criar uma nova do zero, selecionado a opção iniciar instância. Após isso, será apresentado a série de passos necessários para criar uma instância, conforme mostrado na Figura 12. O serviço EC2 também permite que o serviço CloudWatch possa monitorar as aplicações hospedadas nas instâncias criadas, o que possibilita definir alarmes para serem exibidos em gráficos. Dessa forma, o serviço EC2 será utilizado neste trabalho para

criar as instâncias para hospedar o Moodle para poder ser realizado a análise de desempenho.

Figura 12 – Criação de instância no serviço EC2 da AWS.

EC2 > Instâncias > Iniciar uma instância

### Iniciar uma instância [Informações](#)

O Amazon EC2 permite criar máquinas virtuais, ou instâncias, que são executadas na Nuvem AWS. Comece a usar rapidamente seguindo as etapas simples abaixo.

**Nome e tags [Informações](#)**

Nome

 [Adicionar mais tags](#)

**▼ Imagens de aplicação e de sistema operacional (imagem de máquina da Amazon) [Informações](#)**

Uma AMI é um modelo que contém a configuração do software (sistema operacional, servidor de aplicações e aplicações) necessária para executar a instância. Pesquise ou navegue pelas AMIs se você não estiver vendo o que está buscando abaixo

**▼ Resumo**

Número de instâncias [Informações](#)

**Imagem do software (AMI)**

Amazon Linux 2023 AMI 2023.0.2...[Ler mais](#)  
ami-0889a44b331db0194

**Tipo de servidor virtual (tipo de instância)**

t2.micro

**Firewall (grupo de segurança)**

Novo grupo de segurança

**Armazenamento (volumes)**

1 volume(s) - 8 GiB

Cancelar **Executar instância** [Revisar comandos](#)

Fonte: Autor.

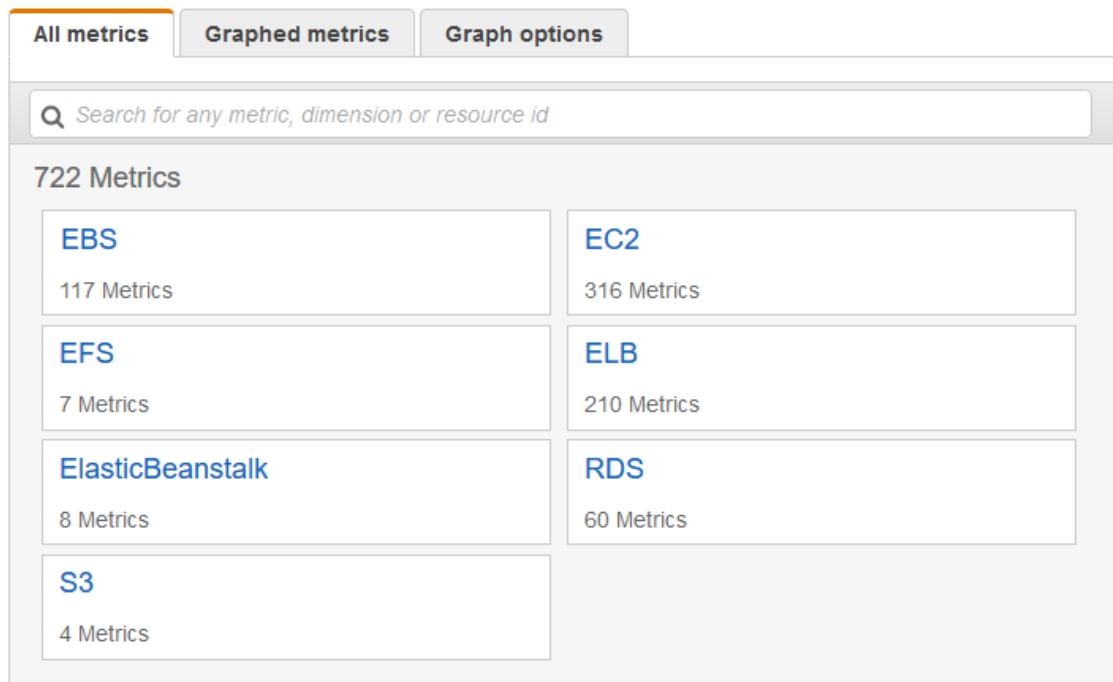
### 2.3.2 Amazon CloudWatch

O serviço Amazon CloudWatch é um serviço da Amazon Web Services que oferece monitoramento de recursos da AWS e aplicações de clientes que utilizam a infraestrutura da Amazon, permitindo que esses recursos sejam monitorados tempo real. Dessa forma, a CloudWatch foi desenvolvida para monitorar aplicações de grande demanda em ambientes de nuvem, o que requer o gerenciamento por meio de métricas, uma vez que isso torna, mas fácil, a recuperação de estatísticas de forma rápida e organizada. Para isso, a CloudWatch permite coletar e rastrear métricas, coletar e monitorar arquivos de log e definir alarmes.

A CloudWatch pode ser usada para monitorar os recursos, tais como as instâncias criadas pelo serviço do EC2, volumes de dados do serviço *Elastic Block Store (EBS)*, *Elastic Load Balancing* e instâncias do *Amazon Relational Database Service (RDS)*. O cliente pode coletar informações e fornecer métricas entre as disponíveis da AWS conforme mostrado na Figura 13. Se for necessário a utilização de métricas para monitorar a aplicação que não esteja na AWS, e possível executar uma solicitação através do método PUT para inserir uma nova métrica na Amazon CloudWatch.

Os usuários que trabalham com a CloudWatch podem acessar as suas funcionalidades através de uma interface de API ou via linha de comando. A interface da CloudWatch permite a visualização das métricas em formato de gráfico. Além disso, é possível criar alerta para os usuários caso algum recurso monitorado ultrapasse um limite determinado. O sistema de

Figura 13 – Catálogo de métricas da Amazon CloudWatch.



Fonte: (Services, 2023).

monitoramento de recursos CloudWatch oferece dois tipos de monitoramento de recursos, o básico ou o detalhado. O primeiro faz o monitoramento a cada cinco minutos para um número limitado de métricas sem nenhum custo, enquanto o segundo faz o monitoramento a cada minuto usando qualquer métrica, mas com um custo adicional.

A arquitetura da CloudWatch tem como objetivo fornecer recursos ou seus dados diretamente na nuvem da AWS, de forma que as métricas, como tempo de resposta, uso de CPU e memória, sejam fornecidas aos usuários da CloudWatch todas as opções possíveis de trabalhar com essas métricas. Dessa forma, a CloudWatch será utilizada para coletar os dados gerados dos testes de carga para serem analisados utilizados as métricas selecionadas para a análise de desempenho.

## 2.4 Avaliação de Desempenho

A avaliação de desempenho é um conjunto de técnicas cujo objetivo é mensurar o desempenho de um sistema, seja ele real ou abstrato. É necessário um certo nível de conhecimento sobre o sistema que está sendo modelado, além de ser considerado com cautela a escolha das técnicas, cargas e ferramentas de análise que serão usadas (Jain, 1990). Para realizar uma avaliação de desempenho, é necessário selecionar um conjunto de métricas de desempenho

que condiz com os objetivos propostos para avaliação, para identificar gargalos no sistema ou planejar melhorias no sistema (Lilja, 2005).

Atualmente existem diversas maneiras de realizar uma avaliação de desempenho, conforme mostrado na Figura 14 (Jain, 1990) (Lilja, 2005). Cada técnica pode ser usada conforme as condições do sistema ou até mesmo por conta própria. Utilizar mais de uma técnica pode permitir um melhor entendimento do sistema para poder tomar a decisão mais adequada. Dentre as técnicas para realizar uma avaliação de desempenho, existem três que se destacam e serão explicadas a seguir (Callou *et al.*, 2011):

- Medição: A medição é uma das técnicas mais comum que existe para avaliar um sistema. Ela consiste em submeter o sistema com uma carga de trabalho, onde serão coletados os resultados para análise, usando métricas previamente definidas. Geralmente, para a execução desse tipo de técnica, costuma ser usado ferramentas de *benchmark*. Essas ferramentas fazem com que sistema realize uma série de ações para monitorar o comportamento do sistema durante o experimento. Existem dois tipos principais de modelo de medição, a prototipação e o monitor ou coleta de dados.
- Modelagem Analítica: A modelagem analítica é uma técnica que utiliza modelos de equações matemáticas para prever o comportamento do sistema. A técnica de modelagem analítica pode ser usada em sistemas que não precisam ser necessariamente criados. Apesar de ser uma técnica flexível, os resultados gerados não são tão precisos em comparação com as outras técnicas que existe atualmente.
- Simulação: A técnica de simulação é composta pelo uso de modelos matemáticos ou dados coletados em sistemas reais, o que é usado para criar um modelo de simulação que terá como informação relevante uma característica do sistema real. Esse modelo será executado por meio de replicações do sistema em diversos computadores para confirmar se o resultado está de fato em conformidade com o sistema real. As simulações podem ser conduzidas por meio de eventos discretos ou contínuos.

As técnicas de medição, apresentadas permitem atingir um dos vários objetivos propostos pela avaliação de desempenho, tais como: comparar alternativas é realizar por meio de uma análise comparativa utilizando as informações dos usuários; análise de impacto, consistem em estudar os impactos causados no sistema e como eles vão se comportar após serem corrigidos, entre outras (Lilja, 2005). Para que isso ocorra é necessário a colaboração entre os envolvidos, tais como: desenvolvedores, testadores, DBAs, administradores de sistema e administradores

Figura 14 – Técnicas de avaliação de desempenho.



Fonte: (Santana *et al.*, 2016).

de rede. A avaliação de desempenho elaborada neste trabalho irá utilizar uma das técnicas de avaliação apresentada que corresponda com a proposta do trabalho.

#### 2.4.1 Apache JMeter

O Apache JMeter é uma ferramenta de teste de carga e medição de desempenho de código aberto desenvolvida pela *Apache Software Foundation*, onde a ferramenta foi criada utilizando a linguagem de programação Java (Jmeter, 2019). A ferramenta foi desenvolvida originalmente para realização de testes em aplicações web, mas foi posteriormente atualizada para realizar além de aplicações *web* agora trabalhando com recursos estáticos e dinâmicos, aplicativos dinâmicos da *Web*, servidores FTP, banco de dados, objetos Java e entre outros (Chandel *et al.*, 2013).

Sendo assim, o JMeter pode ser usado para realizar simulações em servidores utilizando diferentes tipos de cargas de trabalho pesadas para poder analisar e medir o desempenho da aplicação (Jmeter, 2019). O Apache JMeter é uma das principais opções de ferramenta de código aberto que existe atualmente no mercado. Ela possui a maior comunidade ativa para dar suporte em resolução de problemas, a sua interface gráfica é simples de ser utilizada mesmo para usuários com pouca experiência, além de ser bastante customizável, permitindo a utilização de *plugins* de terceiros para utilizar funcionalidades que a ferramenta original não dispõe (Caminero *et al.*, 2013).

Contudo, o JMeter não deve ser comparado a navegador, uma vez que não é capaz

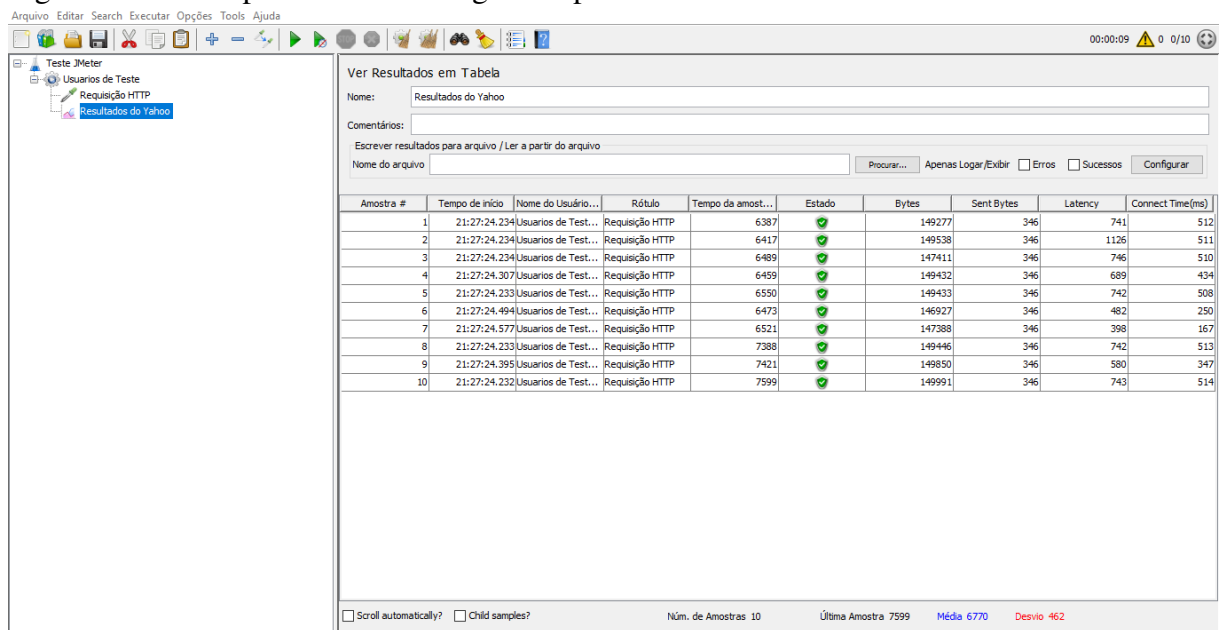
de executar todas as informações presentes em uma página HTML, como, por exemplo, o JavaScript ou muito menos realizar uma renderização da página. Dessa maneira, JMeter oferece a possibilidade de visualizar o que está acontecendo por trás da página, como as respostas entre o cliente e o servidor, por meio da funcionalidade de ouvinte que permite a visualização dos registros das ações.

A sua arquitetura é baseada em um modelo distribuído para realização dos testes de carga, onde o JMeter cria vários usuários virtuais que serão usados para criar a carga de trabalho para medição. Essa carga será enviada para aplicação por meio de requisições ao servidor. Após a realização do teste, é coletado o resultado de cada requisição enviada ao servidor. Com base nesses dados, são elaboradas as estatísticas a partir das métricas de performance permitidas pelo JMeter, como vazão, quantidade de dados enviados por segundo, tempo de resposta, dentre outras. O JMeter tem como objetivo simular cenários de teste que sejam compatíveis com situações reais que possam ocorrer com uma aplicação (Rodrigues *et al.*, 2019) (Matam; Jain, 2017).

Para serem realizados testes no JMeter, conforme apresentado na Figura 15, é necessário realizar uma série de passos na ferramenta, incluindo a adição de diversos tipos de configurações para personalizar os testes. As configurações básicas que devem estar presentes em todos os testes são: criar um plano de teste, adicionar um grupo de usuários, adicionar um modelo de requisição do teste e adicionar ouvintes para capturar as requisições das páginas web. Dessa forma será elaborado os planos de testes, que serão utilizados neste trabalho para gerar carga de trabalho no ambiente do Moodle preparado para os testes de performance.



Figura 15 – Exemplo de teste de carga no Apache JMeter.



Fonte: Elaborado pelo Autor.

### 3 TRABALHOS RELACIONADOS

Nesta seção serão apresentados os trabalhos relacionados ao tema, junto com uma descrição de cada um deles, destacando ao final as diferenças e semelhanças ao que está sendo elaborado neste trabalho.

#### 3.1 Avaliação de desempenho e consumo de energia de um ambiente virtual de aprendizagem em nuvens privadas

No trabalho de Lima *et al.* (2021) é enfatizado a realização de uma avaliação de desempenho e consumo de energia de um AVA, uma vez que, apesar de existirem trabalhos que medem o desempenho, poucos realizam também uma avaliação do consumo de energia. Dessa forma, os autores enfatizam a vantagem de realizar uma avaliação de desempenho e consumo de energia em um AVA, o que pode auxiliar na solucionar problemas que ocorra em fase de implantação ou atualização de algum projeto. O foco do estudo apresentado é avaliar o desempenho e o consumo de energia do Moodle em diferentes ambientes de nuvem privada Apache CloudStack<sup>1</sup> e OpenStack<sup>2</sup>.

Os autores elaboraram uma metodologia para realizar os testes de medição, que se divide em seis etapas: estudar os sistemas em nuvem, instalar e configurar os sistemas, instalar e configurar o serviço, criar um plano de medição, realizar medições e analisar os resultados. Além da metodologia, foi realizado um estudo de caso para demonstrar a aplicação do Moodle em condições reais de uso em ambos os sistemas de nuvem Apache CloudStack e OpenStack, de modo para garantir a fidelidade dos dados.

Como mostra a Figura 16, uma infraestrutura para os testes foi montada, sendo composta por três computadores: (i) uma máquina cliente com JMeter para gerar a carga de trabalho e validar as métricas selecionadas; (ii) uma máquina com sistema do Moodle no Apache CloudStack; e (iii) uma máquina com sistema do Moodle no OpenStack. Para validar os dados, foram selecionadas algumas métricas de desempenho, tais como o uso de CPU, o consumo de memória RAM, o consumo de energia, a vazão e o tempo de resposta. Dessa forma, os testes foram realizados em diferentes cenários, realizando uma atividade específica e considerando o número de clientes no sistema.

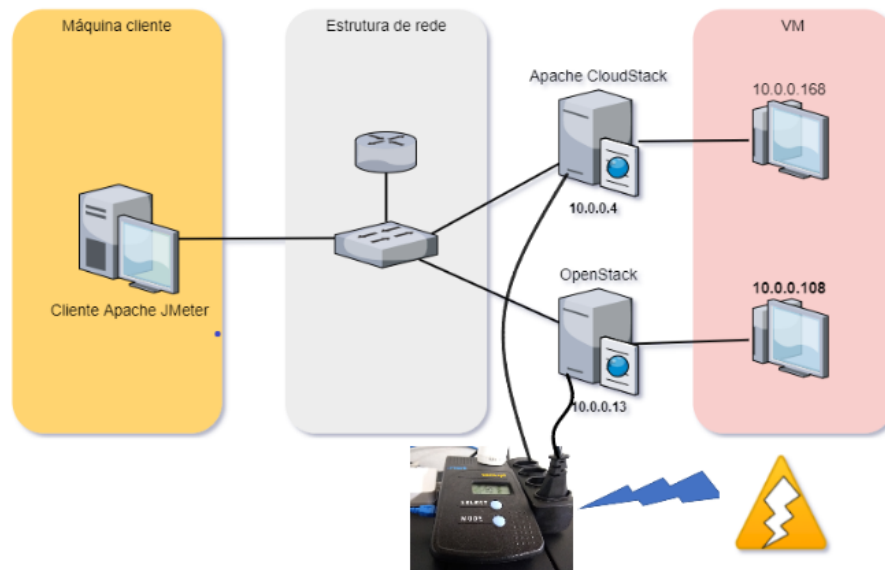
Os resultados mostraram que o desempenho como é o consumo de energia em ambos

---

<sup>1</sup> Apache CloudStack — <https://cloudstack.apache.org/>

<sup>2</sup> OpenStack — <https://www.openstack.org/>

Figura 16 – Infraestrutura montada para os experimentos.



Fonte: (Lima *et al.*, 2021)

os sistemas de nuvem foram diferentes. Sendo possível observar nos experimentos realizados, o OpenStack teve um consumo maior de energia do que o CloudStack. Isso pode ser explicado pelo fato de que o OpenStack requer um maior consumo de CPU. Outro ponto relevante observado foi o tempo de resposta, que, mais uma vez o OpenStack, mostrou-se inferior ao CloudStack, uma vez que houve uma vazão inferior a (60%) em comparação com o OpenStack. Dessa forma, a plataforma CloudStack apresentou o melhor desempenho, demandando um menor consumo de energia, em todos os cenários analisados.

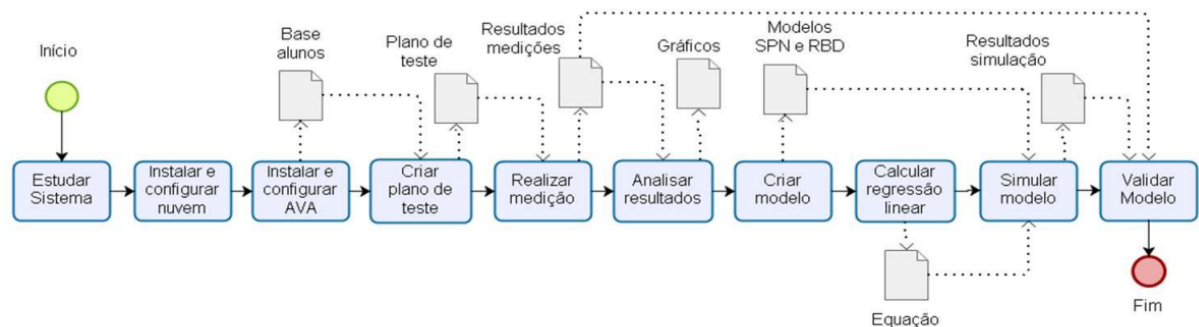
### 3.2 Análise de Desempenho do Ambiente Virtual de Aprendizagem na Nuvem Privada Apache CloudStack

No artigo de Silva *et al.* (2019), foi realizada uma análise de desempenho através de modelos de medição e modelagem de sistema, usado uma rede de Petri estocástica, em um ambiente virtual de aprendizagem Moodle, usado na Universidade Federal Rural de Pernambuco. A infraestrutura de aprendizagem utilizada pela universidade está hospedada na nuvem privada Apache CloudStack. Primeiramente, foi realizado uma análise de desempenho, utilizando algumas métricas, como vazão, tempo de resposta, uso de CPU, memória e taxa de escrita em HD. Os dados gerados por essas métricas, foram utilizados para a avaliação dos modelos de modelagem de sistema propostos, que foram avaliados através de três estudos de caso, simulados em um ambiente real montado em laboratório.

A metodologia utilizada para conduzir esta avaliação de desempenho foi elaborada

com dez atividades de execução, conforme demonstrado na Figura 17. A avaliação de desempenho foi realizada nas seis primeiras atividades, com base no conjunto de métricas estabelecidas. Após isso, na próxima etapa, são criados os modelos de desempenho, utilizando o modelo de rede Petri estocástica (SPN) junto à técnica de regressão linear. Com os modelos criados, foram realizados novamente os mesmos planos de teste para medir todos os modelos propostos, a fim de validar o modelo com as duas estratégias elaboradas.

Figura 17 – Metodologia de avaliação de desempenho.



Fonte: (Silva *et al.*, 2019)

Após a etapa de modelagem, foi realizado o estudo de caso utilizando os três casos propostos. O primeiro deles é uma avaliação da capacidade do sistema usando métricas de vazão e tempo de resposta, utilizando duas VMs com configurações distintas, nas quais os usuários devem apenas acessar o Moodle. O segundo caso apresentado refere-se aos resultados das medições de consumo de CPU e memória e à taxa de escrita no HD, obtido através de uma simulação de uma turma composta por 40 alunos que executavam um conjunto de atividades. No último caso proposto, foi validado o modelo de desempenho proposto na metodologia, com a comparação do tempo de resposta da medição com o tempo da simulação, onde o resultado deve ser o intervalo de confiança entre os dois.

### 3.3 Optimization of LMS for Improving user Response Time

No trabalho de Sabareeshwaran (2015) foi apresentado a ferramenta Moodle, um LMS de código aberto amplamente utilizado no mundo, voltado ao apoio à aprendizagem de forma assíncrona. Segundo os autores, a ferramenta apresenta problemas relacionados ao desempenho quando é utilizada em conjunto com mais de uma função ao mesmo tempo, como fazer um questionário ao mesmo tempo que coleta o *feedback* dos participantes. Os autores

identificaram o problema como sendo um bloqueio do Data Base Management System (DBMS) para atualizar a tabela de sessões do Moodle, por meio de testes de carga.

Dessa forma, os autores propõem uma estratégia para otimizar a inserção de dados na tabela de sessão do banco de dados no Moodle. Além disso, os autores também especificaram alguns requisitos para estruturar a utilização do Moodle. Dessa forma, recomendam que, quando for utilizado recursos como (questionários, provas, fóruns, etc.), evitem o uso de mais de um recurso por vez para evitar a criação de requisições extras desnecessárias.

De acordo com um administrador do Moodle, quando o sistema é utilizado em um servidor dedicado, o número de usuários que o banco de dados suporta é superior ao de uma máquina normal, o que resulta numa alteração nas configurações necessárias para a utilização, resultando em um mecanismo de bloqueia as atualizações dos registros, resultando em um aumento do consumo de memória. Para comprovar isso, foi realizado um teste com 400 usuários simultâneos onde apresentou um consumo de memória de (433,9%) somente em requisições para o MySQL.

Para solucionar o problema de otimização do Moodle, os autores propuseram a remoção desse bloqueio do banco de dados, para permitir que as atualizações realizadas na tabela de sessão do Moodle sejam realizadas de forma contínua. Em função de, quando houver novas consultas, elas permanecerão suspensas até que sejam concluídas as atualizações dos registros pela tabela de sessão.

Para comprovar que a alteração realmente funciona, utilizou-se o Apache JMeter para gerar uma carga com vários usuários simultaneamente, usando um *script* com as informações para o *login* dos usuários. Como consequência, a otimização fornecida pela remoção do bloqueio das tabelas do banco de dados permitiu ao Moodle suportar (30%) mais usuários do que a versão não otimizada.

### **3.4 *Choosing the right LMS: A performance evaluation of three open-source LMS***

O trabalho de Caminero *et al.* (2013) tem como objetivo identificar qual LMS escolher entre as opções disponíveis pode atender às necessidades para funcionamento de uma instituição de ensino. Para isso foram selecionados três LMSs de código aberto (LRN<sup>3</sup>, Sakai<sup>4</sup> e Moodle), e realizado uma avaliação de desempenho entre as três ferramentas selecionadas. Os

---

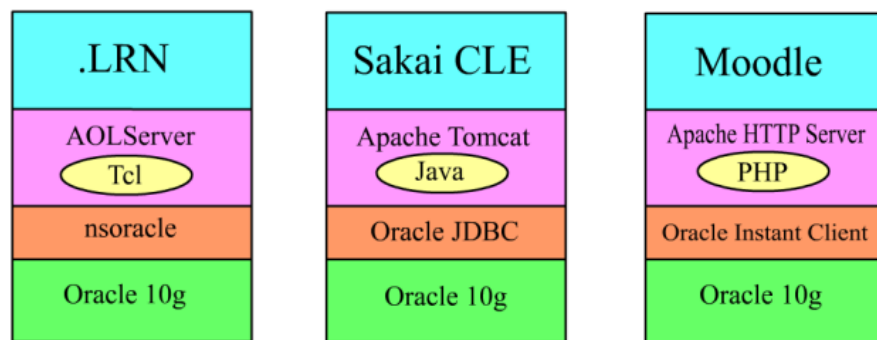
<sup>3</sup> LRN — <http://www.dotlrn.org/>

<sup>4</sup> Sakai — <https://www.sakailms.org/>

experimentos foram conduzidos com usuários virtuais em diferentes cenários executando alguma tarefa e, posteriormente, analisadas com um conjunto de métricas selecionadas como consumo de memória e uso CPU.

Para uma comparação eficiente, as ferramentas foram configuradas de forma semelhante, sendo assim, foram utilizadas as mesmas configurações em todas as ferramentas, conforme apresentado na Figura 18. Além disso, o mesmo banco de dados foi utilizado em todos os LMSs escolhidos, sendo escolhido o Oracle 10g devido à sua compatibilidade com todos os LMSs escolhidos. Os administradores dos sistemas também relataram suas experiências para auxiliar na conclusão do estudo.

Figura 18 – Arquitetura de software dos LMSs testados.



Fonte: (Caminero *et al.*, 2013)

Os autores selecionaram a ferramenta de teste de carga Apache JMeter para conduzir as simulações para avaliar o desempenho dos LMSs neste estudo. Após isso, foi elaborado um conjunto de planos de teste para a execução dos cenários, os quais incluem *login/logout*, postar em um fórum, enviar uma tarefa, baixar um arquivo, agendar um evento usando o calendário e responder a um questionário, no qual cada um desses cenários foi executado dez vezes. Dessa maneira, foram criados cenários para cada LMS, os quais foram reproduzidos automaticamente pelo JMeter, utilizando uma ferramenta chamada JMeter HTTP Proxy Server.

Os experimentos foram divididos em duas partes, sendo que inicialmente, foi realizada uma avaliação de desempenho nos três LMSs, usando os testes elaborados no Apache JMeter. Foram aplicados testes com 90, 100 e 110 usuários virtuais para avaliar as métricas de consumo de CPU, memória, I/O de rede e disco. Os resultados obtidos indicam que o Sakai apresenta um desempenho superior em comparação com outras ferramentas, ficando atrás apenas do LRN e, por fim, do Moodle, que apresentou um maior consumo de recursos, mas também apresentou um desempenho inferior devido ao colapso em alguns cenários.

Após isso, elaborou-se uma avaliação do ponto de vista do administrador, que deve considera as funcionalidades dos LMSs de forma paliativa, com uma pontuação variando de 1 (pior) a 5 (bom). O conjunto de funcionalidades foi definido pelos próprios administradores. Nessa avaliação, o Sakai obteve a pontuação mais elevada, seguido pelo LRN e, por fim, pelo Moodle. Dessa forma, chegou-se à conclusão de que, do ponto de vista do sistema, as ferramentas mais adequadas são Sakai e LRN, devido ao uso eficiente dos recursos da máquina nos experimentos. Já do ponto de vista do administrador do sistema, Moodle e Sakai apresentam classificações semelhantes devido ao suporte a comunidades grandes e ativa, enquanto o LRN não tem o mesmo suporte.

### 3.5 A Study on Moodle Virtual Cluster in Cloud Computing

Em Guo *et al.* (2013) é apresentado uma solução para o problema de saturação causado pelo excesso de trabalho em um único servidor Moodle. Os autores sugerem uma solução que consiste em criar um cluster virtualizado usando computação em nuvem ao invés de utilizar vários servidores físicos, o que só traria problemas como desperdício de energia, além de ser inflexível.

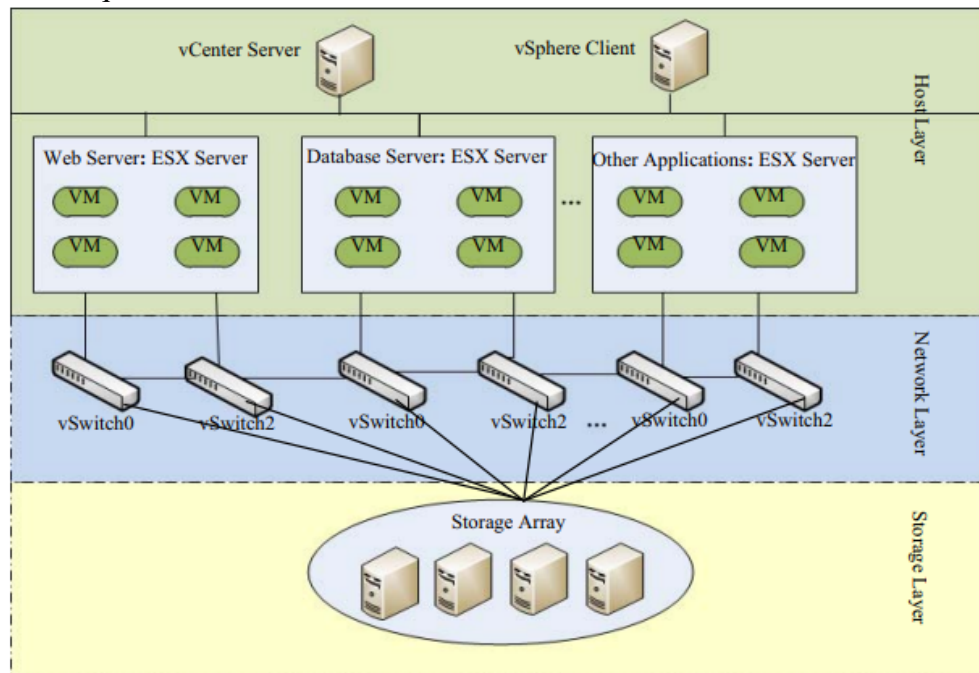
E apresentado inicialmente que o servidor Moodle que foi utilizado pertence a dois campus universitários de Jinhai e Jiashan, cuja configuração não era adequada para suportar a carga de trabalho, sendo necessário, portanto, o desenvolvimento do cluster.

Assim, o cluster criado pelos autores utiliza a plataforma de virtualização VMware vSphere<sup>5</sup>, que foi construída em torno do serviço de nuvem, devido à flexibilidade e elasticidade dos recursos. A arquitetura foi dividida em três camadas: rede, host e armazenamento, como mostra a Figura 19, sendo realizado uma virtualização em cada uma delas.

A avaliação de desempenho proposta para confirmar esse estudo é um teste de pressão realizado através do acesso via web usando a ferramenta de *benchmark* Siege, usado para testes de carga. Essa ferramenta permite o acesso por múltiplos usuários simultaneamente, registrando cada tentativa e repetindo o acesso até o limite desejado. Os testes foram conduzidos num servidor físico e no cluster proposto no trabalho. Inicialmente, os testes começam com 10 usuários simultâneos, com duração de 1 min, até atingir 200 usuários. Os testes são repetidos 10 vezes para poder ter uma maior precisão dos dados. As métricas utilizadas são capacidade média de transação e tempo médio de resposta.

<sup>5</sup> VMware vSphere — <https://www.vmware.com/br/products/vsphere.html>

Figura 19 – Arquitetura do Cluster Virtual Moodle.



Fonte: (Guo *et al.*, 2013)

Os resultados da primeira métrica demonstram que o cluster do Moodle apresenta um desempenho inferior em termos de capacidade de transação em comparação ao servidor único quando a carga de usuários é inferior a 40 devido a um problema de sobrecarga. No entanto, os resultados mudam quando a carga de usuários simultâneos aumenta, apresentando estabilidade em um pico de 160 usuários, ao contrário do servidor único, que apresenta uma queda de (65%) no desempenho.

A segunda métrica revelou que o cluster apresenta um maior tempo de resposta ao longo do tempo, ao passo que o número de usuários aumenta, ao contrário do servidor único, que demanda um tempo maior para responder aos usuários devido ao seu ritmo de trabalho mais lento. Dessa forma, os autores chegaram à conclusão de que o cluster virtual do Moodle consegue resolver o problema de saturação por carga de trabalho em um único servidor físico.

### 3.6 Comparação entre os Trabalhos Relacionados.

A Tabela 1 apresenta os pontos em comum e diferentes entre os trabalhos relacionados com trabalho proposto com base nos critérios apresentados nele.

Os critérios de comparação utilizados na Tabela 1 estão separados da seguinte forma:

(i) Infraestrutura utilizada, que está relacionado ao tipo de ambiente utilizado para execução dos testes, seja eles provedores de serviço de nuvem, virtualização ou até servidores físicos;



(ii) Apache JMeter, referente a ferramenta principal escolhida pelo trabalho proposto para gerar a carga de trabalho; (iii) Moodle como sendo o AVA selecionado pelo trabalho proposto para receber a carga de trabalho; (iv) Ferramenta de coleta de dados é análise métrica, está relacionado ao uso de alguma ferramenta para coletar é analisar os dados gerados dos testes, feito pelo JMeter no Moodle no caso do trabalho proposto refere ao serviço da AWS CloudWatch; (v) Realizam teste de performance, refere ao uso de teste que utilizam carga de trabalho em diferentes condições com objetivo de verificar o comportamento gerado pela carga de trabalho; e (vi) Diversidade de cenários de teste está relacionado aos critérios 3 e 5 onde será criado diversas atividades diferentes dentro do Moodle como, efetuar login/logout, envio de atividade, baixar arquivos entre outras para gerar a carga de trabalho dos testes de performance do trabalho proposto.

Ao analisar os trabalhos apresentados, percebe-se que cada um apresenta uma perspectiva e situações diferentes sobre o tema abordado no trabalho proposto, como no trabalho de Lima *et al.* (2021), mas se assemelha com a proposta deste estudo, pois se trata de uma avaliação do desempenho do Moodle em um ambiente de nuvem, usando o Apache JMeter como ferramenta de teste de carga em diferentes cenários de teste. Contudo, há algumas diferenças, uma vez que o foco do trabalho está principalmente no consumo de energia do ambiente Moodle, o que não é abordado no trabalho proposto. Além disso, o trabalho utiliza ambientes de nuvem privada ao invés de nuvem pública, como é o caso do trabalho proposto, que utiliza a plataforma da AWS. Além disso, não é apresentado o uso de qualquer ferramenta para analisar o desempenho através dos dados gerados, como no caso da CloudWatch, a ferramenta que será utilizada no presente trabalho.

O mesmo se aplica ao trabalho de Silva *et al.* (2019), que não utiliza uma plataforma de nuvem pública para hospedar o Moodle e nem utiliza ferramenta para analisar os dados por meio do uso de métricas, diferente do trabalho proposto que usa o serviço cloudwatch da AWS. Além disso, o trabalho é bastante complexo, pois é realizado o uso de modelagem de sistema com regressão linear, o que é diferente do que é proposto para ser realizado no trabalho proposto.

No trabalho de Caminero *et al.* (2013) é feito um estudo para comparar três LMS bastante utilizados no mercado atualmente, com o objetivo de demonstrar qual é o melhor LMS para ser utilizado em uma instituição. Esse trabalho tem um certo grau de semelhança com o trabalho proposto, assim como ambos os trabalhos anteriores, que usam a mesma ferramenta, o

sistema do Moodle, e até uma grande variedade de cenários, mas, ao mesmo tempo, difere, pois não utiliza nenhuma, plataforma de nuvem, são apenas servidores físicos, o que significa que há riscos de falhas e diferenças significativas nos resultados.

Já os trabalhos Sabareeshwaran (2015) e Guo *et al.* (2013) são trabalhos que não tem como foco principal a análise de desempenho, mas sim apresentam uma solução para algum problema que pode surgir ao usar o Moodle como AVA numa determinada situação. A avaliação de desempenho entra, mas para validar a solução desses problemas. Esses problemas, particularmente do trabalho Sabareeshwaran (2015), pode ser facilmente incorporado como proposta para ser adicionado num trabalho futuro que possa ser realizado no trabalho proposto, uma vez que é algo que já está presente no código-fonte do próprio sistema do Moodle. Sendo assim, caso a solução desse trabalho seja aplicada, pode-se obter resultados diferentes dos quais podemos alcançar com o trabalho proposto.

Tabela 1 – Comparativo entre os trabalhos relacionados e o proposto

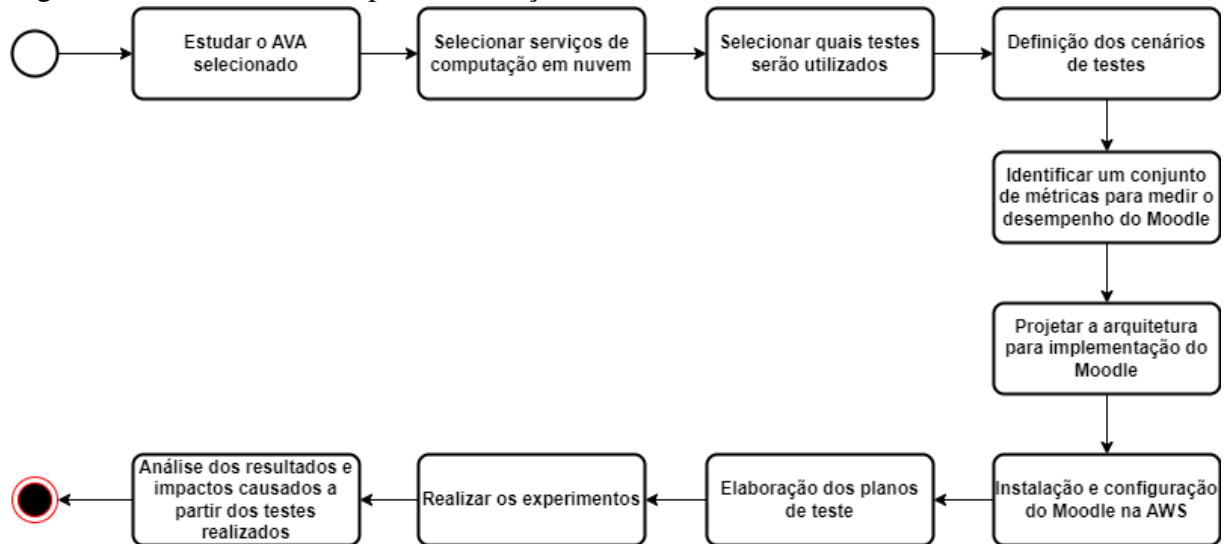
| Trabalho                        | Infraestrutura utilizada      | Apache JMeter | Moodle | Ferramenta de coleta de dados é análise de métricas | Realiza teste de performance | diversidade de cenários de teste |
|---------------------------------|-------------------------------|---------------|--------|---|------------------------------|----------------------------------|
| (Lima <i>et al.</i> , 2021)     | Apache CloudStack e OpenStack | Sim           | Sim    | Não   | Sim                          | Sim                              |
| (Silva <i>et al.</i> , 2019)    | Apache CloudStack             | Sim           | Sim    | Não   | Sim                          | Sim                              |
| (Caminero <i>et al.</i> , 2013) | Servidores físicos            | Sim           | Sim    | Não   | Sim                          | Sim                              |
| (Sabareeshwaran, 2015)          | Servidores físicos            | Sim           | Sim    | Não   | Sim                          | Não                              |
| (Guo <i>et al.</i> , 2013)      | VMware vSphere                | Não           | Sim    | Não   | Sim                          | Não                              |
| Trabalho proposto               | AWS                           | Sim           | Sim    | Sim   | Sim                          | Sim                              |

Fonte: elaborada pelo autor.

## 4 METODOLOGIA

Nesta seção são apresentados os procedimentos metodológicos para a execução deste trabalho. A Figura 20 apresenta os seguintes passos para a execução do trabalho: i) Estudar o AVA selecionado; ii) Selecionar serviços de computação em nuvem; iii) Selecionar quais testes serão utilizados; iv) Definição dos cenários de testes; v) Identificar um conjunto de métricas para medir o desempenho do Moodle; vi) Projetar a arquitetura para implementação do Moodle; vii) Instalação e configuração do Moodle na AWS; viii) Elaboração dos planos de teste; ix) Realizar os experimentos; x) Análise dos resultados e impactos causados a partir dos testes realizados.

Figura 20 – Procedimentos para a execução do trabalho.



Fonte: Elaborado pelo autor.

### 4.1 Estudar o AVA selecionado

O objetivo desta etapa é analisar as características que compõem um ambiente Moodle, desde as funcionalidades até os requisitos de sistema necessários para hospedar a sua própria instância do Moodle. Isso permitirá uma compreensão mais ampla do processo de hospedagem de uma cópia do Moodle na plataforma de nuvem da AWS, bem como a configuração adequada para a realização dos experimentos.

### 4.2 Selecionar serviços de computação em nuvem

Nesta etapa serão definidos os serviços de nuvem necessários para a execução do trabalho proposto. Esses serviços serão tanto para hospedar a aplicação quanto para monitorar

o desempenho para poder ser realizado a análise posteriormente. Os serviços utilizados serão da plataforma de nuvem AWS, inicialmente, devido à sua ampla variedade de serviços. Os serviços escolhidos foram a Amazon EC2 para criação de instâncias aonde serão hospedados os servidores do Moodle e o serviço da CloudWatch para monitoramento dos dados gerados a partir dos testes de carga.

#### **4.3 Selecionar quais testes serão utilizados**

Nesta etapa, serão selecionados os tipos de teste de desempenho dentre as opções disponíveis atualmente que podem ser executadas pelo Apache JMeter, a fim de gerar as cargas de trabalho necessárias para a execução dos testes. Dentre os testes selecionados estão, basicamente, os testes de estresse, teste de carga e o teste de fumaça. Inicialmente, será usado o teste de fumaça para gerar uma pequena carga de trabalho, para confirmar que a aplicação do Moodle usada irá suportar os outros testes.

#### **4.4 Definição dos cenários de testes**

Nesta etapa, serão definidos os cenários de teste, que serão utilizados nos planos de teste do JMeter. Cada cenário corresponde a uma funcionalidade presente no ambiente do Moodle, sendo assim os cenários selecionados para serem utilizados neste trabalho são: fazer *login/logout* no Moodle com uma quantidade  $x$  de usuários, navegar pelas páginas do site, acessar um curso e responder a perguntas que existe no fórum de discussão do curso.

#### **4.5 Identificar um conjunto de métricas para medir o desempenho do Moodle**

O objetivo desta etapa é definir as métricas que serão utilizadas para analisar os resultados dos testes realizados nos cenários apresentados no trabalho. As métricas propostas a serem utilizadas neste trabalho estão relacionadas aos testes selecionados. Dessa forma, as métricas selecionadas para esse trabalho são, utilização de CPU, latência, throughput, tempo de resposta, perda de pacotes, taxa de erro, usuários concorrentes entre outras. Essas métricas serão utilizadas através do serviço CloudWatch da AWS para realizar a análise de desempenho do ambiente Moodle.

#### **4.6 Projetar a arquitetura para implementação do Moodle**

Nesta etapa será definida qual será a configuração do ambiente utilizado para hospedar o Moodle na plataforma da AWS. Para isso, serão usados diferentes tipos de instâncias do serviço EC2, inicialmente instâncias do tipo uso geral próprias para serem utilizadas em servidores web em produção ou para cargas de trabalho mais pesadas, onde será utilizado especificamente instancias do modelo M7i-flex, M6a e M5n para permitir uma variedade de configurações. Cada tipo de instância tem a sua configuração de sistema já definida, o que pode resultar em conclusões diferentes ao fazer a análise de desempenho.

#### **4.7 Instalação e configuração do Moodle na AWS**

Nesta etapa, será realizada a instalação e configuração do ambiente Moodle nas instâncias EC2, que foram preparadas na etapa anterior. Nessas instâncias serão instaladas todas as dependências de software necessárias para o pleno funcionamento, tais como a criação da conta de administrador para o uso dos recursos e a criação de uma base de dados fictícia contendo turmas, atividades e arquivos didáticos, que serão utilizados na criação dos cenários de teste propostos no trabalho, que utilizará carga de trabalho.

#### **4.8 Elaboração dos planos de teste**

Nesta etapa serão elaborados e configurados os planos de testes, que serão executados na ferramenta Apache JMeter para realizar os experimentos no ambiente Moodle hospedado na nuvem. Esses planos de teste são basicamente os cenários de teste em que serão executados. Eles serão criados através da função que permite criar planos de teste no próprio Moodle, sendo necessário definir algumas informações para a criação do plano como, por exemplo, qual curso será utilizado no teste, a quantidade  $x$  de usuários que serão utilizados e a quantidade de execução por usuário, a criação de *endpoint* para gerenciar o controle de usuários, além da criação de ouvintes para capturar os resultados dos testes em tempo real.

#### **4.9 Realizar os experimentos**

Após a conclusão das etapas anteriores serão executados os planos de testes elaborados no Apache JMeter para gerar cargas de trabalho diretamente nas instâncias da AWS que

possuem o Moodle hospedado. Cada plano de teste executado no Moodle será destinado a uma atividade específica definida anteriormente para avaliar o desempenho da funcionalidade do Moodle, além dele todo.

#### **4.10 Análise dos resultados e impactos causados a partir dos testes realizados**

Durante a execução dos experimentos serão coletados dados obtidos por meio da CloudWath, para serem posteriormente interpretados e consolidados para poderem ser gerados gráficos e tabelas através do uso de métricas já definidas que estão presentes no serviço da CloudWatch. A análise de desempenho será realizada com objetivo de avaliar o desempenho do Moodle quando ocorrer dele receber uma determinada carga de trabalho enquanto realizar alguma ação, e como isso afeta no seu desempenho.

## 5 PROJETO DE EXECUÇÃO

Nesta seção serão apresentados as configurações utilizadas em todos os componentes do experimento, ferramentas, como foi realizado a implementação e análise dos resultados obtidos dos testes durante o processo de execução.

### 5.1 Configurações realizadas nos serviços da AWS

A configuração e execução realizadas neste trabalho foram feitas na plataforma de nuvem da *AWS Academy*, mas precisamente na região *us-east-1* onde os servidores da aplicação do Moodle ficarão hospedados. Essa é uma região onde o custo pelo uso dos serviços não é tão alto, além de ser uma região bastante conhecida pelos usuários para tirar o máximo proveito da plataforma da AWS.

Todos os serviços necessários para a execução do trabalho foram gerenciados pelo console de gerenciamento as AWS, visando um acompanhamento mais preciso dos resultados em tempo real. As configurações realizadas no serviço de criação de instância EC2 são compostas por três instancias do modelo de uso geral do tipo M7i-flex, M6a e M5n voltadas para a carga de trabalho em servidores web.

Portanto, dessa forma, era para selecionar inicialmente três tipos de configurações para esses modelos de instâncias, sendo elas (*large*, *xlarge* e *2xlarge*). Essas instâncias apresentam configurações de vCPU e Memória (GiB) inferior, próximo e superior respectivamente às configurações utilizadas na máquina virtual utilizada para hospedar o Moodle utilizado pela Universidade Federal do Ceará no campus de Quixadá. Mas devido às limitações impostas na *AWS Academy* para restringir o uso de recursos de alto valor, só foi possível criar instâncias com o modelo *large*, conforme mostrado na figura da tabela 21.

Figura 21 – Configuração das instancias da AWS.

| Tipo de instancia | Tamanho de instância | vCPU | Memória (GiB) | Armazenamento de instâncias (GB) | Largura de banda da rede (Gbps) | Largura de banda do EBS |
|-------------------|----------------------|------|---------------|----------------------------------|---------------------------------|-------------------------|
| M7i-flex          | m7i-flex.large       | 2    | 8             | Somente EBS                      | Até 12,5                        | Até 10 Gbps             |
| M6a               | m6a.large            | 2    | 8             | Somente EBS                      | Até 12,5                        | Até 10 Gbps             |
| M5n               | m5n.large            | 2    | 8             | Somente EBS                      | Até 25                          | Até 4.750 Mbps          |

Fonte: Elaborado pelo autor.

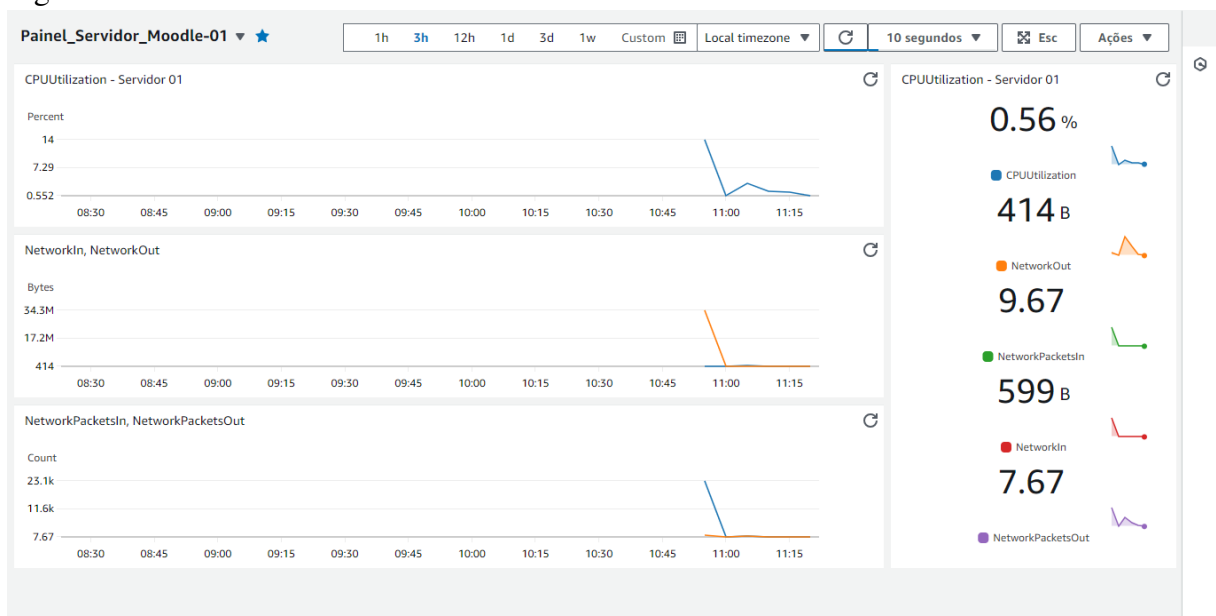
Dessa forma, foi criado um grupo de segurança para ser utilizado nas instâncias, liberando as portas para *http* e *https* para poder ser permitido o acesso externo ao Moodle, bem como também permitindo a passagem do tráfego gerado pela carga de trabalho criada no trabalho, além de uma porta para o *SSH*, o que permitirá a instalação do servidor Moodle nas instâncias e

uma porta para conexão com o banco de dados para os dados gerados pelo servidor do Moodle.

Além disso, foi adicionado um IP elástico em cada instancia para permitir o acesso sempre que se iniciar uma nova sessão no *AWS Academy*, uma vez que o IP que é normalmente utilizado nas instâncias quando a sessão e encerra não é mais possível acessar a aplicação do Moodle. Finalmente, foi estabelecido em cada instância um espaço de armazenamento de 50 GB para garantir o armazenamento necessário para criar todos os cursos de teste necessários para a execução do trabalho.

Finalmente, foram elaboradas as configurações necessárias no serviço da *CloudWatch* para monitorar as instâncias com os servidores do Moodle, a fim de coletar os dados obtidos durante os testes de carga, por meio de métricas pré-selecionadas que serão empregadas para medir o desempenho da aplicação em diferentes cenários de teste. Para isso, foram criados painéis para cada servidor, como mostrado na figura 22, com intuito de organizar todas as métricas que serão utilizadas para facilitar o acampamento do monitoramento durante a execução dos testes.

Figura 22 – Píanel de metricas do servidor do Moodle.



Fonte: Elaborado pelo autor.

As métricas selecionadas são aquelas que já estão disponíveis para o uso do serviço EC2, ou seja, já estão pré-instaladas para qualquer instância criada no serviço. Todas essas métricas serão apresentadas em gráficos separados para uma melhor organização, especialmente os de linha e numéricos, devido à facilidade de compreensão que esses tipos de gráficos oferecem. Além dessas alterações, também foi alterado o tempo de atualização de dados, que, tradicional-



mente, é de cinco minutos. A atualização será a cada dez segundos, para permitir a coleta de dados de forma mais aprofundada para ser analisado e discutido posteriormente.

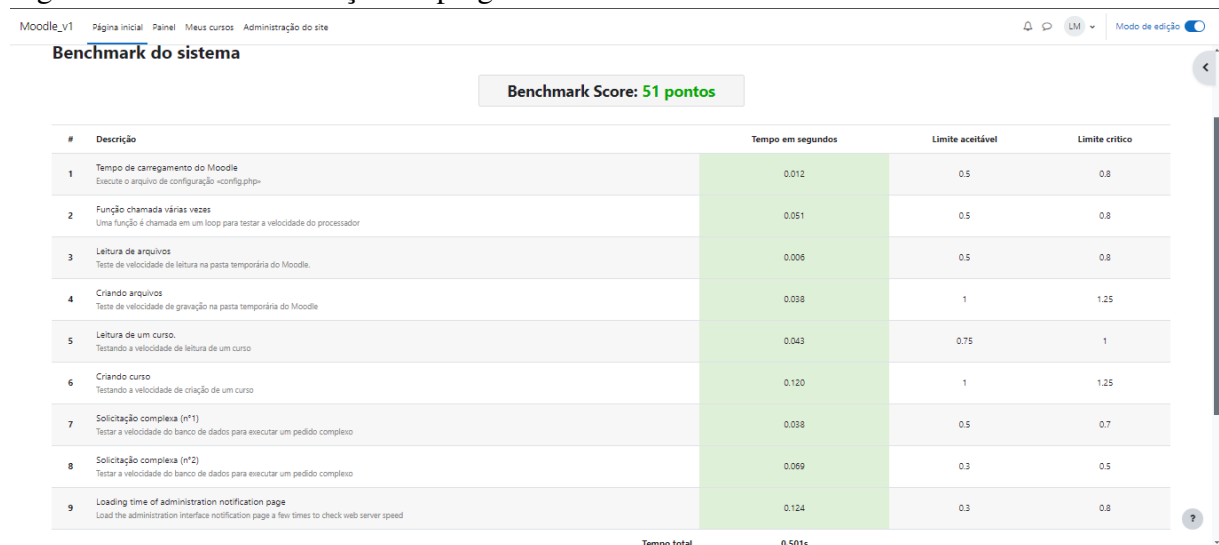
## 5.2 Configurações realizadas na aplicação do Moodle

Primeiramente, foi necessário instalar a aplicação do Moodle nas instâncias do serviço EC2, onde foi necessário instalar todas as dependências necessárias do *PHP* para o funcionamento dos módulos internos do Moodle. Além disso, foi necessário criar um diretório chamado moodledata para armazenar todo conteúdo que será criado pelos usuários, como também foi preciso criar um banco de dados com o nome Moodle e um usuário administrador para o banco.

Após isso, o próximo passo é finalizar a instalação pelo navegador web, informando as configurações para o banco de dados criado anteriormente. Em seguida, é necessário criar uma conta de usuário administrador do Moodle, além de criar um nome para a aplicação do Moodle e concordar com termos de contrato para finalizar a instalação com êxito.

Após o termino da instalação do servidor do Moodle foi instalado o *plugin de benchmark* para verificar a integridade dos módulos do Moodle é ao final informa uma nota que pode ficar entre os dois limites definidos, os quais são: limite aceitável e limite critico, como mostrado na figura 23. Isso será útil para a execução dos testes para determinar como a carga de trabalho afeta os módulos do Moodle, para isso o *benchmark* será executado antes e depois dos testes para poderem ser comparados os resultados.

Figura 23 – Tela de execução do plugin de benchmark do Moodle.



Fonte: Elaborado pelo autor.

Por fim, foi necessário modificar algumas das configurações do Moodle e do PHP em seus diretórios raízes, uma vez que o Moodle por ser um *software livre*, apresenta por padrão configurações predeterminadas abaixo do necessário, bem como outras que não foram adicionadas que impede a execução do trabalho. primeiramente, no diretório do *PHP*, foi necessário aumentar o valor das variáveis: *max-input-vars* para permitir o fim da instalação do Moodle e *memory-limit* para aumentar a velocidade do processamento do *PHP* das páginas. Já no diretório do Moodle, foi adicionado uma variável para criação de senhas dos usuários para os planos de teste do *JMeter*.

### 5.3 Estudo dos cenários de teste viáveis para execução

Tendo em vista que o objetivo deste trabalho é uma análise dos efeitos gerados no Moodle por meio de cargas de trabalho em diferentes cenários, é preciso primeiramente avaliar quais cenários podem ser empregados para atingir esse objetivo. Para isso, foram escolhidos previamente quais cenários de testes seriam utilizados, a fim de verificar a viabilidade de execução.

Dessa forma, foi realizado um estudo de cada possível cenário, lembrando que, apesar de o Moodle ser um *software livre*, os módulos que já estão presentes nele não podem ser alterados conforme a vontade do administrador do sistema, o que pode prejudicar a execução de alguns cenários, podendo até mesmo impedir a execução deles. Por conta disso, nos cenários de *Download e Upload* de arquivo, devido à natureza do Moodle, durante a gravação dos testes através do *plugin do BlazeMeter*, que estava sendo usado para gravar os testes para o *JMeter*, o arquivo utilizados nas duas operações não era capturado, pois o arquivo era enviado diretamente ao banco. Dessa forma, não era possível capturar o arquivo para ser utilizado posteriormente nos testes reais.

No cenário de teste, que consiste em responder um questionário, o formulário de perguntas possuía variáveis com valores dinâmicos, o que dificultava quando o arquivo de teste gerado na gravação era executado, resultado em uma resolução em branco do questionário. Além disso, o Moodle não reconhecia os arquivos csv's criados pelo usuário para enviar as respostas ao questionário.

Os outros testes demonstraram que atendem as expectativas, como o *login/logout*, acessar um curso, navegar entre as páginas e responder a um questionamento em um fórum, devido a uma funcionalidade que existe dentro do Moodle que gera cargas de trabalho para

cursos especialmente criados para medir o desempenho do Moodle, onde são criados diversas tarefas para testar diferentes funcionalidades.

#### 5.4 Configuração realizada para criação dos planos de teste

Este trabalho foi executado utilizando o modulo interno do Moodle para a criação de curso de teste, que serão utilizados no *Apache JMeter*. Para isso, foi necessário habilitar o modo de *debugging*, que está desabilitado por padrão, uma vez que é um módulo usado para avaliar o desempenho da aplicação e não pode ser usado em sites que estão em produção, uma vez que pode causar uma sobrecarga no site, levando a ficar *offline*.

O modulo de criação de cursos de teste possibilita a criação de seis configurações diferentes de um mesmo curso para avaliar o desempenho, conforme mostrado na figura 24. Para este trabalho serão utilizados os cursos do tipo S ao XXL, uma vez que essas configurações são adequadas para gerar a carga de trabalho necessária para este trabalho.

Figura 24 – Quadro de especificações de criação de curso de teste.

| Tipo de Curso | Tamanho - Tempo de Criação | Usuarios | Topicos | Arquivos Pequenos | Arquivos Grandes | Paginas | Fórum Postagem | Tarefa |
|---------------|----------------------------|----------|---------|-------------------|------------------|---------|----------------|--------|
| XS            | 10KB - 1s                  | 1        | 2       | 1                 | 1                | 1       | 2              | 1      |
| S             | 10MB - 30s                 | 100      | 10      | 64                | 2                | 50      | 20             | 10     |
| M             | 100MB - 2min               | 1000     | 100     | 128               | 5                | 200     | 500            | 100    |
| L             | 1GB - 30min                | 10000    | 500     | 1024              | 10               | 1000    | 5000           | 500    |
| XL            | 10GB - 2h                  | 50000    | 1000    | 16384             | 10               | 5000    | 10000          | 1000   |
| XXL           | 20GB - 4h                  | 100000   | 2000    | 32768             | 10               | 10000   | 20000          | 2000   |

Fonte: Elaborado pelo autor.

Após a criação dos cursos de teste, é necessário criar os planos de teste a partir dos cursos criados, utilizando o módulo chamado criar plano de teste *JMeter*. Para isso, é necessário que o Moodle esteja em modo *debugging*. Além disso, é necessário ir ao arquivo de configuração do Moodle chamado *config.php* para adicionar uma linha de comando que permitirá criar as senhas para os usuários cadastrados no curso. Feito isso, resta só selecionar o tipo de curso entre as opções apresentadas na figura 25. Ao finalizar a elaboração dos planos de teste, dois arquivos serão disponibilizados para *download*, um deles contendo o plano de teste e outro contendo um arquivo csv com os *logins* e senhas dos usuários do curso para serem utilizados nos testes.

#### 5.5 Configuração realizada no Apache JMeter

Com todas as configurações feitas na aplicação do Moodle, o que falta ser feito para iniciar a execução do projeto será realizado na ferramenta *Apache JMeter*, utilizando os

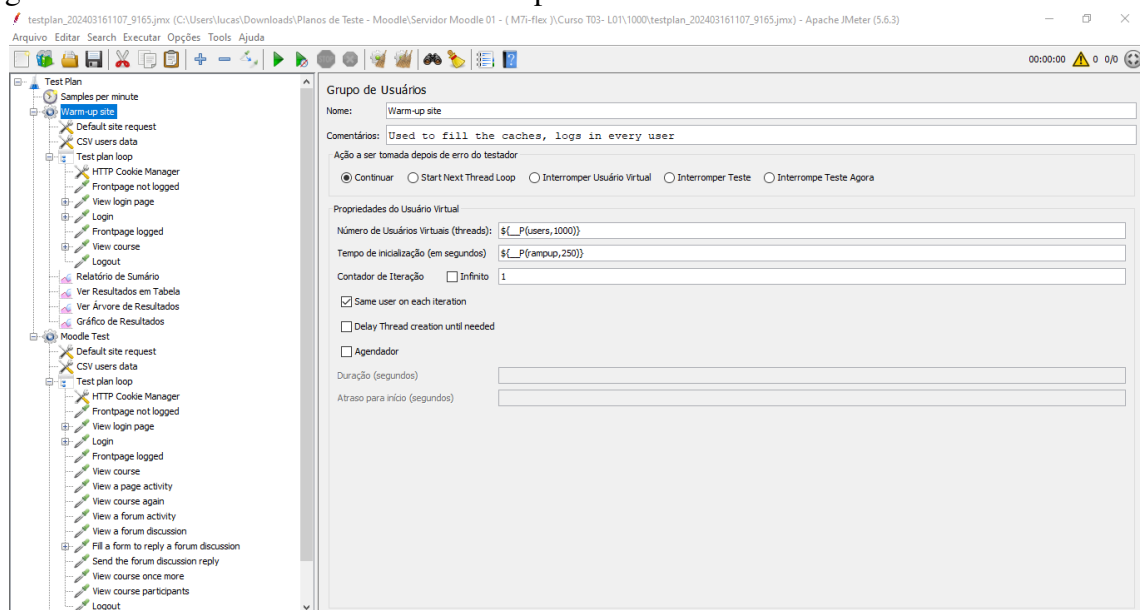
Figura 25 – Quadro de especificações de criação de plano de teste.

| Tipo de Curso | Quantidade de Usuarios | Loops | Periodo de Construção |
|---------------|------------------------|-------|-----------------------|
| XS            | 1                      | 5     | 1                     |
| S             | 30                     | 5     | 6                     |
| M             | 100                    | 5     | 40                    |
| L             | 1000                   | 6     | 100                   |
| XL            | 5000                   | 6     | 500                   |
| XXL           | 10000                  | 7     | 800                   |

Fonte: Elaborado pelo autor.

planos de testes já criados, com o intuito de realizar algumas alterações adicionais utilizando os recursos disponibilizados pela ferramenta. Por meio disso, serão realizadas algumas alterações no componente *Thread Group*, utilizado para adicionar uma quantidade  $x$  de usuários simultâneos, além do número de interações realizadas por cada um dos usuários, bem como temporizadores para controlar o tempo de execução de cada um conforme mostrado na figura 26.

Figura 26 – Plano de teste do Moodle no Apache JMeter.



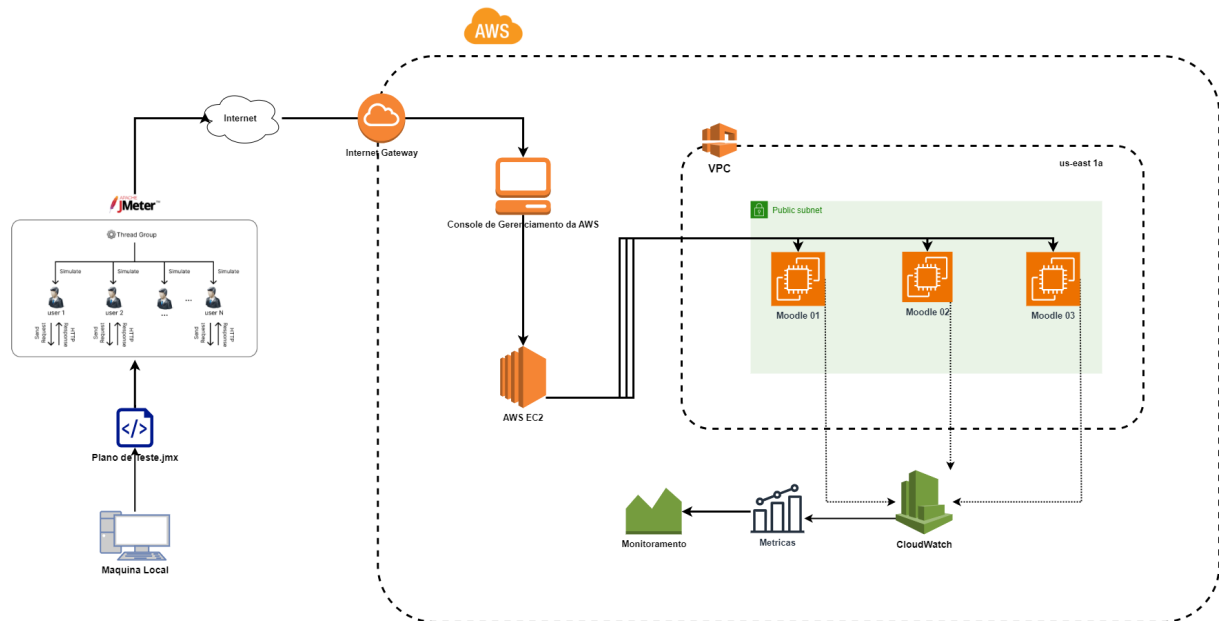
Fonte: Elaborado pelo autor.

Além das configurações realizadas no componente *Thread Group*, foram adicionados mais elementos *HTTP Request*, para aumentar a interação de acesso a páginas para o teste de navegar entre as páginas do site do Moodle, não somente as que contém no curso de teste, mas também as de outros cursos e recursos do Moodle. Para isso, será definido o protocolo, o nome do servidor, a porta, além do caminho ou rota que a aplicação precisará acessar para atender a sua solicitação específica. Finalmente, será adicionado os ouvintes como uma maneira alternativa de

monitorar a execução do projeto pelo *JMeter*, para assegurar os logs de todos os eventos que ocorrerão durante a execução, bem como para identificar qualquer erro que surja no processo.

Ao final de todas essas etapas teremos todo o processo de execução preparado para ser executado seguindo um fluxo definido conforme mostrado na figura 27.

Figura 27 – Arquitetura de execução do projeto.



Fonte: Elaborado pelo autor.

## 6 RESULTADOS E DISCUSSÃO

Nessa seção serão discutidos os resultados obtidos através dos testes elaborados na seção anterior do trabalho que seguirá uma configuração conforme mostrado na figura 28. As configurações foram feitas dessa maneira devido a várias testes feito com antecedência para confirmar que os servidores estão funcionando corretamente, bem como devido às limitações de tempo de hardware disponível pela nuvem da Amazon.

A execução do trabalho foram feitos em três dias seguidos, tendo início no dia 12 a 14 de fevereiro de 2025 no período da noite, tendo o início por volta das 20:00 horas com intervalos de 10 minutos entre as execuções para uma melhor visualização dos resultados nos gráficos criados na *CloudWatch*.

Os planos de teste foram criados para executar uma quantidade definida de requisições dos usuários, que estão listados num arquivo csv onde no primeiro teste ira executar 150 requisições, ao término do teste e feito um intervalo no segundo teste e executado 500 requisições, na terceira será 1000, na quarta 2000 e por último no quinto teste e executado 3000 requisições. Para chegar na quantidade definidas de usuários foi utilizado dois elementos do Jmeter chamados, *loops* e *Threads*, onde a *thread* que representa o usuário irá repetir a solicitação o número de vezes definido em *loops*.

Por último foi defino o elemento *Ramp-up* que representa a quantidade de tempo em segundos em que o Jmeter deve levar para iniciar as *threads* defina no plano onde de forma simples o cálculo e representado *Ramp-up/thread*, dessa forma evitado que o servidor tenha um gargalho após iniciar o teste, no caso desse trabalho os valores definidos são padrões já definidos no próprio modulo do Moodle onde são criados.

Figura 28 – Configuração dos planos de teste do Moodle.

| Teste    | Quantidade | Loops | Ramp-Up | Threads | Tempo de duração |
|----------|------------|-------|---------|---------|------------------|
| Primeiro | 150        | 5     | 6       | 30      | 15 Minutos       |
| Segundo  | 500        | 5     | 40      | 100     | 20 Minutos       |
| Terceiro | 1000       | 1     | 100     | 1000    | 25 Minutos       |
| Quarto   | 2000       | 2     | 100     | 1000    | 30 Minutos       |
| Quinto   | 3000       | 3     | 100     | 1000    | 35 Minutos       |

Fonte: Elaborado pelo autor.

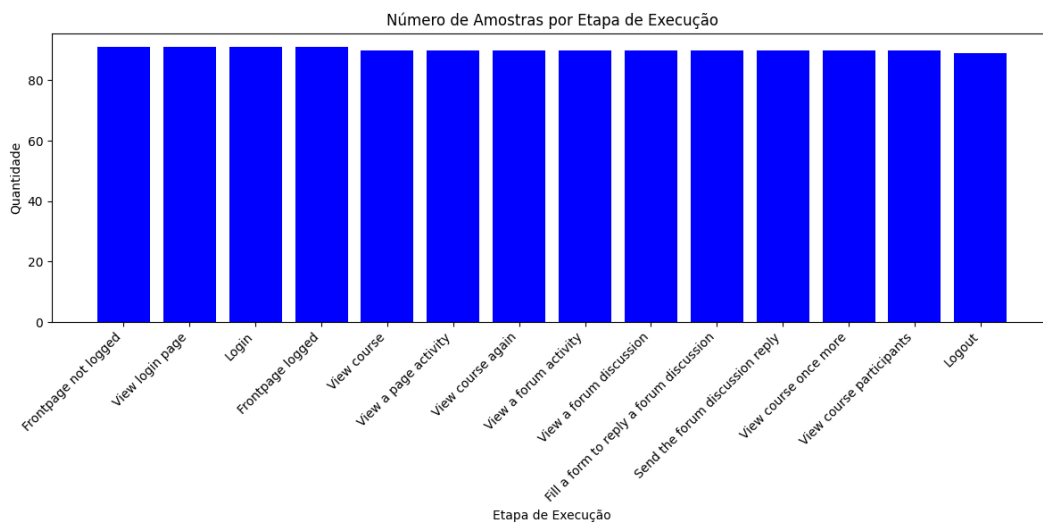
## 6.1 Execução do primeiro servidor do Moodle

Nesse parte será feita a análise das métricas referente ao primeiro servidor do Moodle que estar instalado na instância do tipo M7i-flex do serviço EC2 da AWS.

### 6.1.1 Primeira execução

O primeiro teste foi iniciado às 20:00 do dia 12 de fevereiro, com duração até às 20:15, executando uma carga de 150 threads no servidor Moodle. Embora o tempo total disponível tenha sido completamente utilizado, apenas 91 *threads* foram processadas com sucesso, conforme demonstrado na figura 29. A análise das requisições realizadas permitiu identificar três instâncias de falha durante a execução do plano de testes, conforme ilustrado na figura 30. O terceiro evento de falha apresentou maior relevância, decorrente do término do período alocado para execução do teste, o que resultou no encerramento forçado das *threads* pelo JMeter.

Figura 29 – Grafico de requisições por etapas - primeira execução do primeiro servidor.

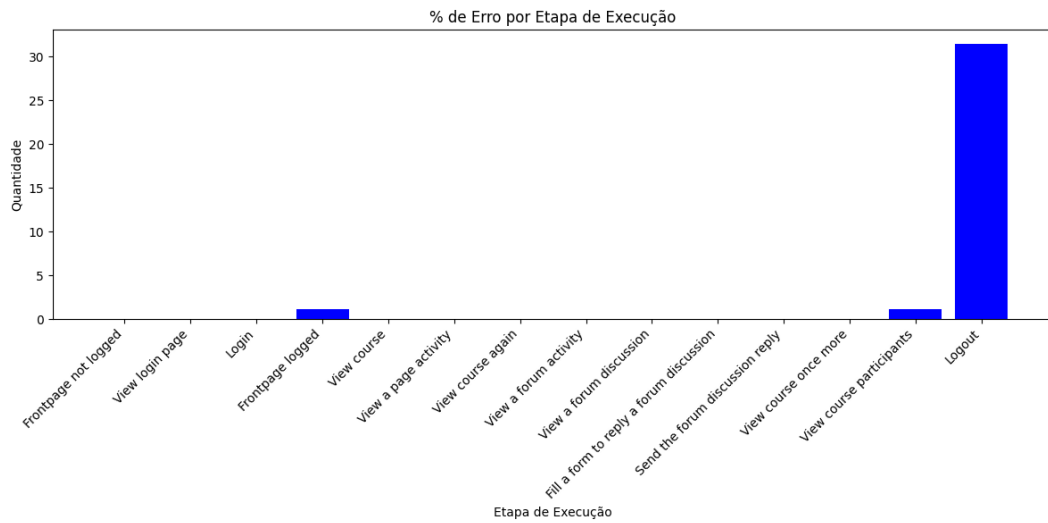


Fonte: Elaborado pelo autor.

Conforme análise das métricas do *Amazon CloudWatch*, com ênfase no parâmetro *CPUUtilization* que monitora o uso do processador, constatou-se que a instância EC2 hospedando o servidor Moodle atingiu utilização máxima da CPU durante os testes. Os dados apresentados na figura 31 demonstram que a carga aplicada, embora de curta duração (15 minutos), foi suficiente para saturar completamente os recursos computacionais disponíveis.

Também é possível observar nas métricas *NetworkOut* e *NetworkPacketsOut* no

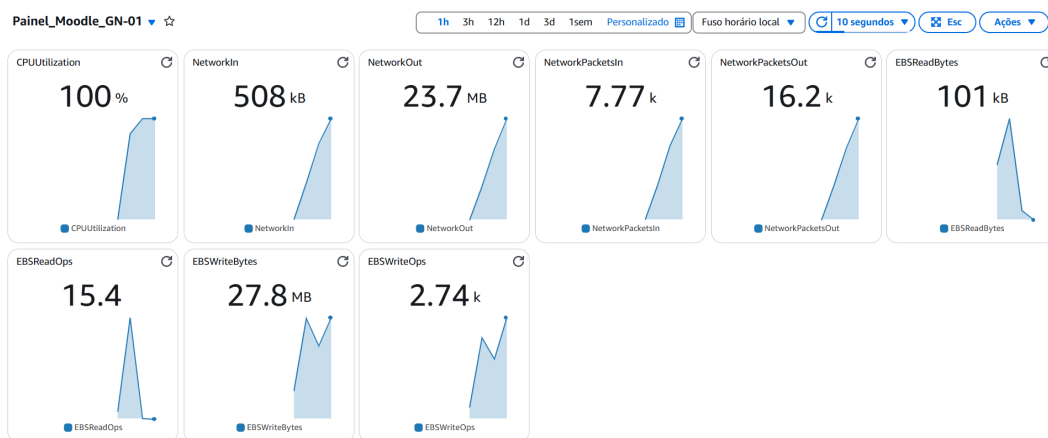
Figura 30 – Grafico de erros por etapas - primeira execução do primeiro servidor.



Fonte: Elaborado pelo autor.

*Amazon CloudWatch* revela que o volume de tráfego de saída excedeu consistentemente o tráfego de entrada nas interfaces de rede da instância. Esta disparidade quantitativa comprova a ausência de perdas significativas na transmissão de dados para o servidor durante o teste. Ademais, o servidor conseguiu lidar com o grande volume de pedidos feitos durante o teste, preservando a estabilidade do sistema e possibilitando o acesso mesmo após o término da avaliação.

Figura 31 – Grafico referente as metricas analisadas do teste do Moodle - primeira execução do primeiro servidor.



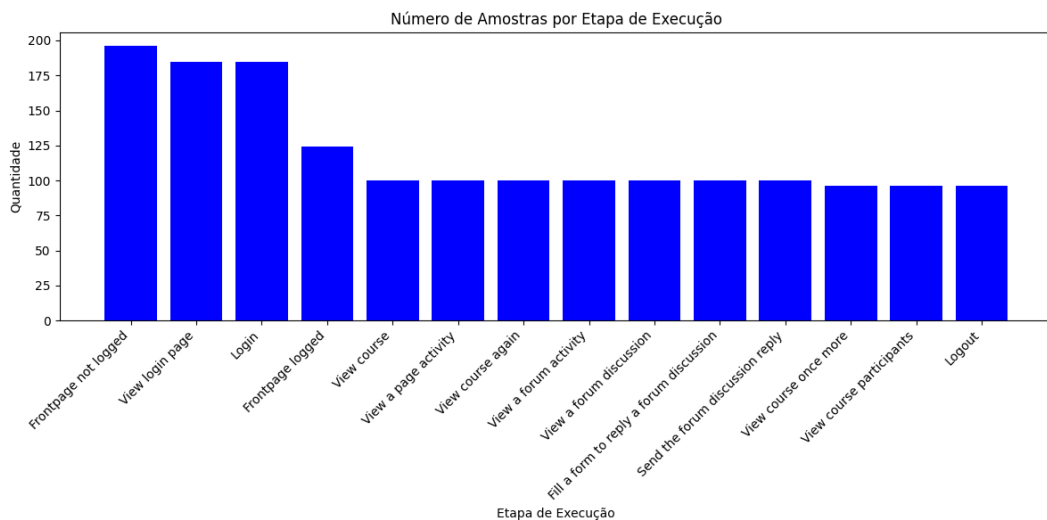
Fonte: Elaborado pelo autor.



### 6.1.2 Segunda execução

O segundo teste foi conduzido logo após intervalo de 10 minutos do primeiro, iniciando às 20:25 e concluindo às 20:45. Nesta fase, foi aplicada uma carga de 100 *threads* distribuídas em cinco *loops*, totalizando 500 *threads* planejadas. Contudo, conforme demonstrado na Figura 32, apenas 196 *threads* (39,2% do total) completaram alguma etapa do processo de teste dentro do período estabelecido. A análise dos resultados, ilustrada na Figura 33, revela a ocorrência de sete instâncias de falha durante a execução das *threads*. Nota-se que todas as falhas ocorreram predominantemente nas fases iniciais do teste, indicando possíveis problemas de inicialização ou estabilização do sistema sob carga.

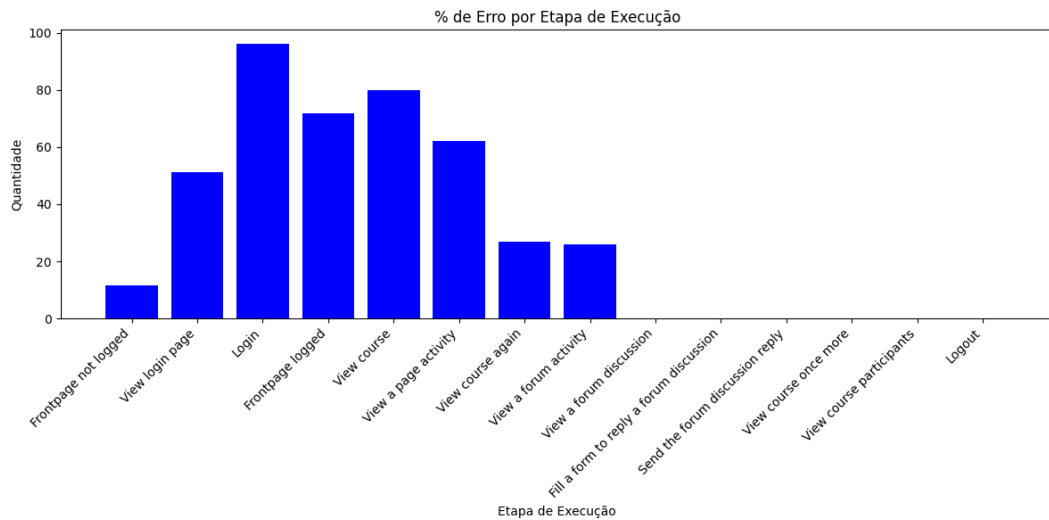
Figura 32 – Grafico de requisições por etapas - segunda execução do primeiro servidor.



Fonte: Elaborado pelo autor.

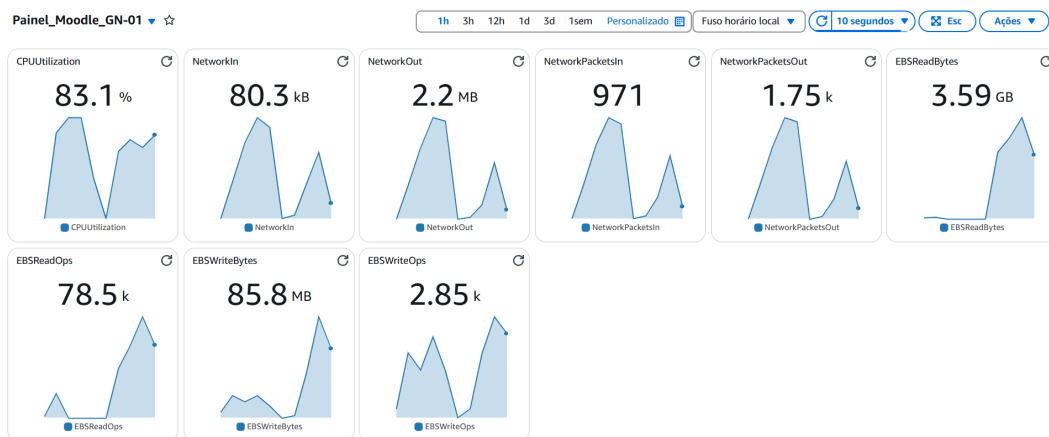
Análise das métricas do *Amazon CloudWatch* na Figura 34 revela que o consumo de CPU, embora elevado, apresentou níveis inferiores aos observados no teste anterior. Registraram-se dois picos de utilização transitórios, nenhum deles sustentado, sugerindo que a carga de trabalho impactou moderadamente o desempenho do servidor. Observa-se ainda que os fluxos de dados de entrada e saída mantiveram inicialmente paridade, indicando operação estável. Contudo, após determinado período, constatou-se declínio constante no *throughput*, fenômeno que pode correlacionar-se com as falhas registradas na Figura 33. Apesar dessas ocorrências, o servidor manteve estabilidade operacional, permanecendo acessível imediatamente após a conclusão do experimento.

Figura 33 – Grafico de erros por etapas - segundo execução do primeiro servidor.



Fonte: Elaborado pelo autor.

Figura 34 – Grafico referente as metricas analisadas do teste do Moodle - segunda execução do primeiro servidor.



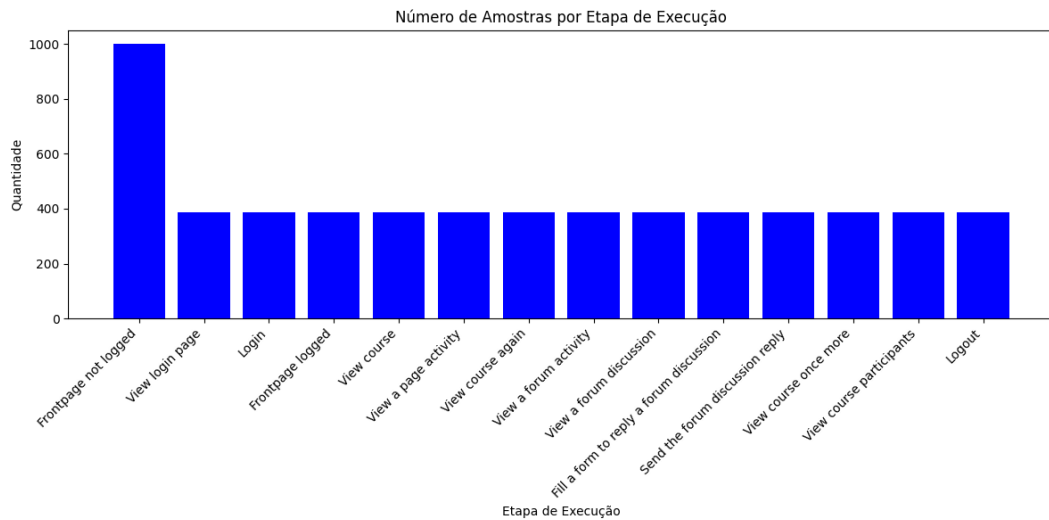
Fonte: Elaborado pelo autor.

### 6.1.3 Terceira execução

O terceiro teste foi iniciado às 20:56, após um intervalo de 11 minutos do teste anterior, com duração até 21:21. Nesta etapa, aplicou-se uma carga de 1.000 *threads* em um único *loop*. Durante o período total do teste, apenas parte da carga foi processada completando-se todas as requisições apenas na primeira etapa, mantendo-se constante o número de 388 *threads* nas etapas subsequentes conforme observado na Figura 35. A análise revelou uma taxa de falha de 100%, indicando que nenhuma *thread* concluiu com êxito todas as etapas do plano de teste observado na Figura 36.

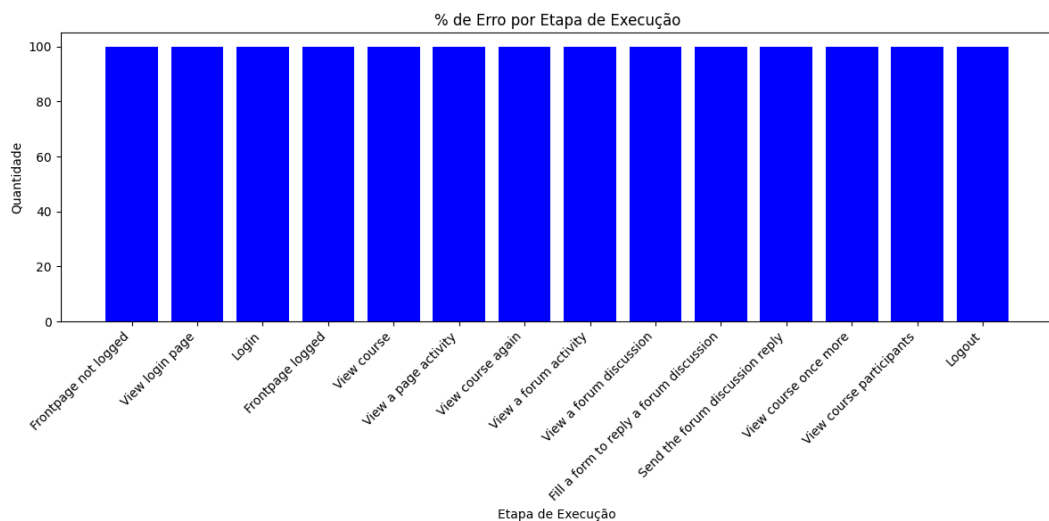
No gráfico da Figura 37 das métricas do Amazon CloudWatch demonstra que a CPU foi significativamente impactada pela carga de trabalho, registrando variações entre 60-

Figura 35 – Grafico de requisições por etapas - terceira execução do primeiro servidor.



Fonte: Elaborado pelo autor.

Figura 36 – Grafico de erros por etapas - terceira execução do primeiro servidor.



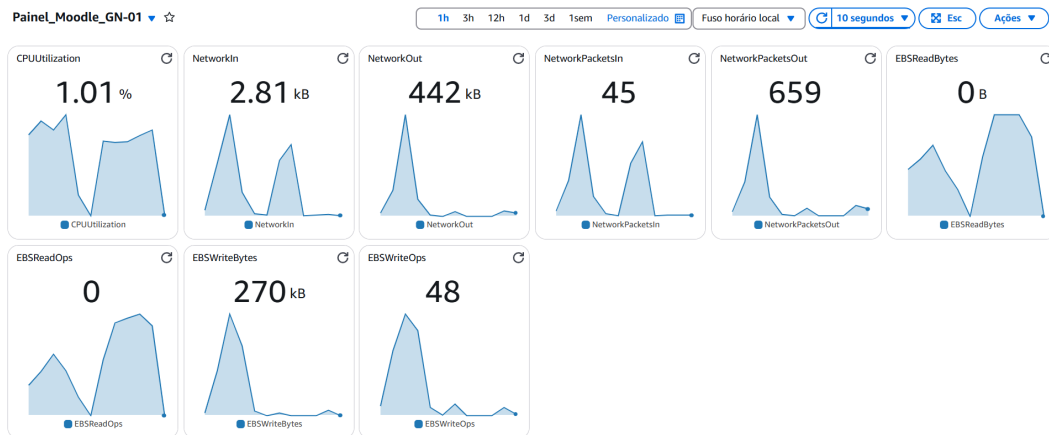
Fonte: Elaborado pelo autor.

75% de utilização antes de declinar abruptamente para níveis próximos a zero. Este padrão sugere a ocorrência de um gargalo de desempenho, onde a capacidade do servidor foi excedida, resultando em perda de dados durante a transmissão. Os dados evidenciam que, após um pico inicial no volume de tráfego de entrada, houve uma queda acentuada no *throughput*, correlacionando-se com os erros apresentados na Figura 36. Notavelmente, apesar da sobrecarga que causou o gargalo, a instância manteve-se disponível e acessível após o término do teste.

#### 6.1.4 Quarta execução

O quarto teste foi iniciado após um intervalo de 10 minutos do teste anterior, com início às 21:31 e término às 22:01. Nesta etapa, aplicou-se uma carga de 1.000 *threads* distribuídas

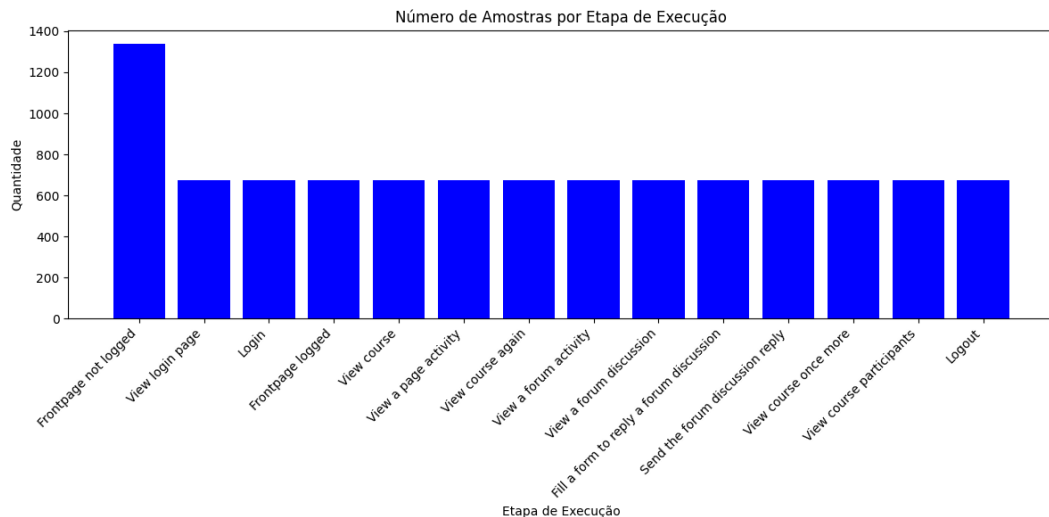
Figura 37 – Grafico referente as metricas analisadas do teste do Moodle - terceira execução do primeiro servidor.



Fonte: Elaborado pelo autor.

em dois *loops*, totalizando 2.000 *threads*. Conforme demonstrado na Figura 38, o sistema utilizou todo o tempo disponível para processar parcialmente a carga, concentrando-se principalmente na primeira etapa do plano de testes, com 1.338 requisições executadas. A análise dos resultados da Figura 39 revelou que todas as *threads* executadas apresentaram falhas completas, sem obtenção de resposta do servidor em qualquer etapa do processo de teste.

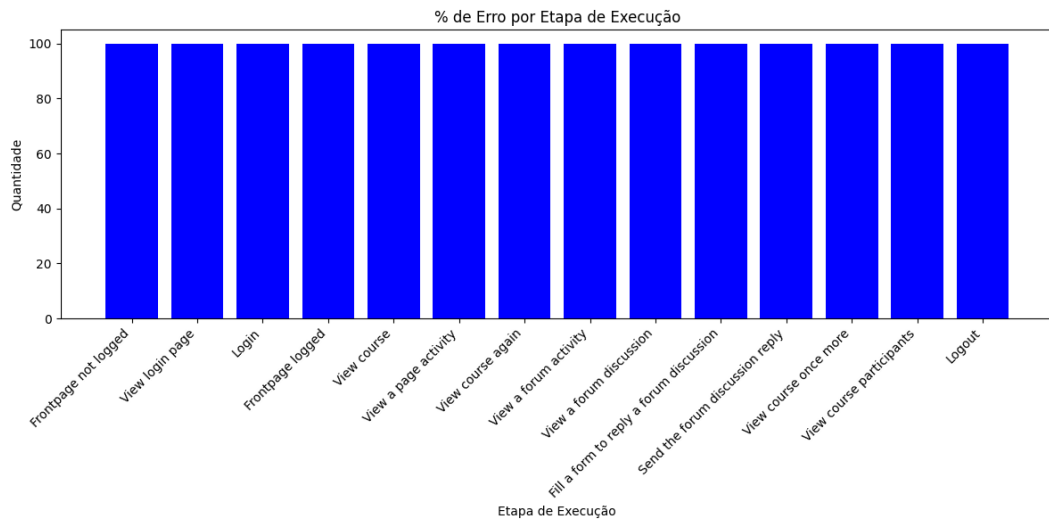
Figura 38 – Grafico de requisições por etapas - quarta execução do primeiro servidor.



Fonte: Elaborado pelo autor.

Análise das métricas do *Amazon CloudWatch* Figura 40 revela que o consumo da CPU, assim como no teste anterior, foi significativamente impactado pela carga de trabalho. O servidor apresentou rápido consumo de recursos no início do teste, atingindo aproximadamente 68% de utilização, seguido por um declínio constante até alcançar apenas 5,2% da capacidade

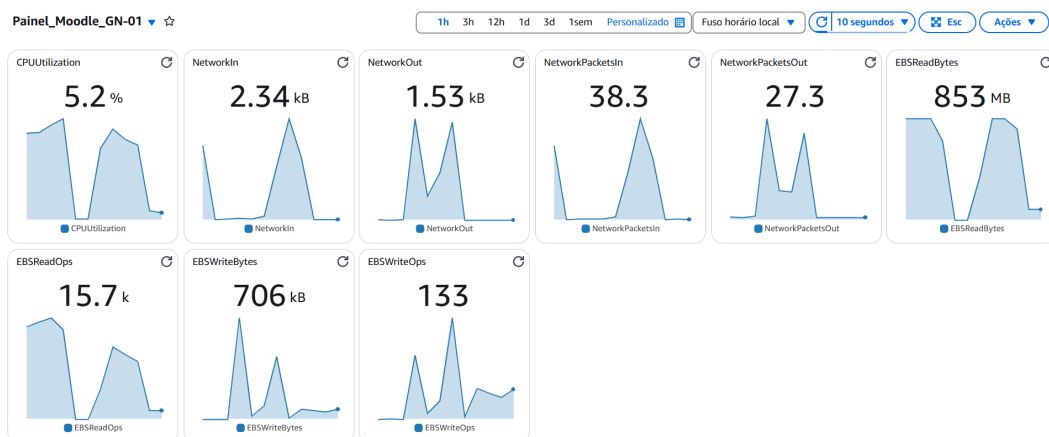
Figura 39 – Grafico de erros por etapas - quarta execução do primeiro servidor.



Fonte: Elaborado pelo autor.

ao final do teste. Este comportamento indica a ocorrência de um gargalo de desempenho no servidor Moodle. Observando as métricas de *NetworkIn* e *NetworkPacketsIn*, constata-se que o volume de dados recebidos pelo servidor foi inicialmente reduzido e diminuiu progressivamente, acompanhando o padrão de queda observado na utilização da CPU. Diferentemente dos testes anteriores, neste caso a carga aplicada foi suficiente para causar a indisponibilidade completa do servidor, enquanto nos demais havia apenas degradação de desempenho sem perda total de acessibilidade.

Figura 40 – Grafico referente as metricas analisadas do teste do Moodle - quarta execução do primeiro servidor.

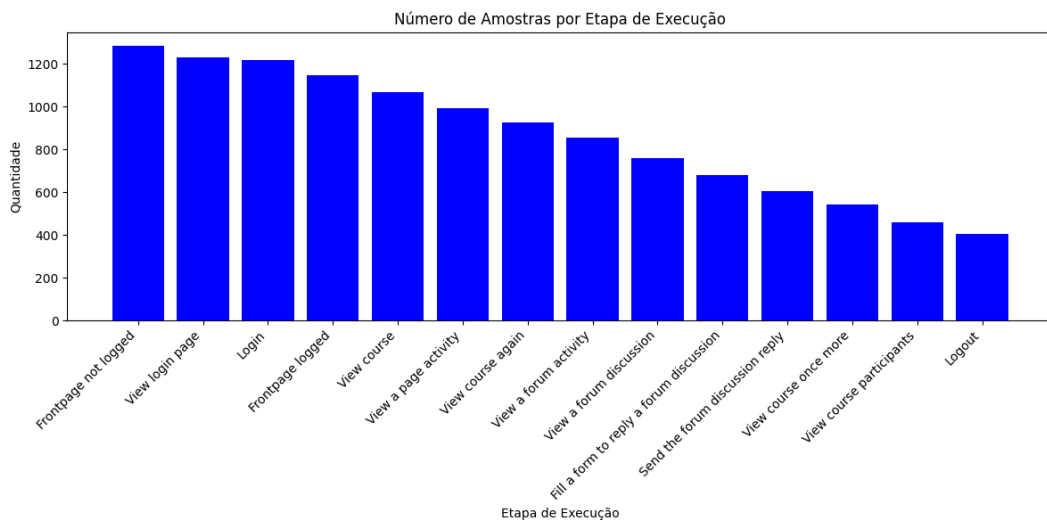


Fonte: Elaborado pelo autor.

### 6.1.5 Quinta execução

O quinto teste foi conduzido das 22:11 às 22:46, após um intervalo de 10 minutos do teste anterior. A carga aplicada consistiu em 1.000 *threads* distribuídas em três *loops*, totalizando 3.000 *threads*. Durante os 45 minutos de duração do teste, o sistema processou apenas parte da carga, concentrando-se principalmente nas etapas iniciais do plano de testes, conforme demonstrado na Figura 41. Embora o volume total de requisições processadas tenha sido inferior ao teste anterior, observou-se uma redução significativa na taxa de erros. A análise da Figura 42 revela que algumas *threads* foram executadas com sucesso, indicando uma melhoria relativa no desempenho do sistema em comparação com os testes precedentes.

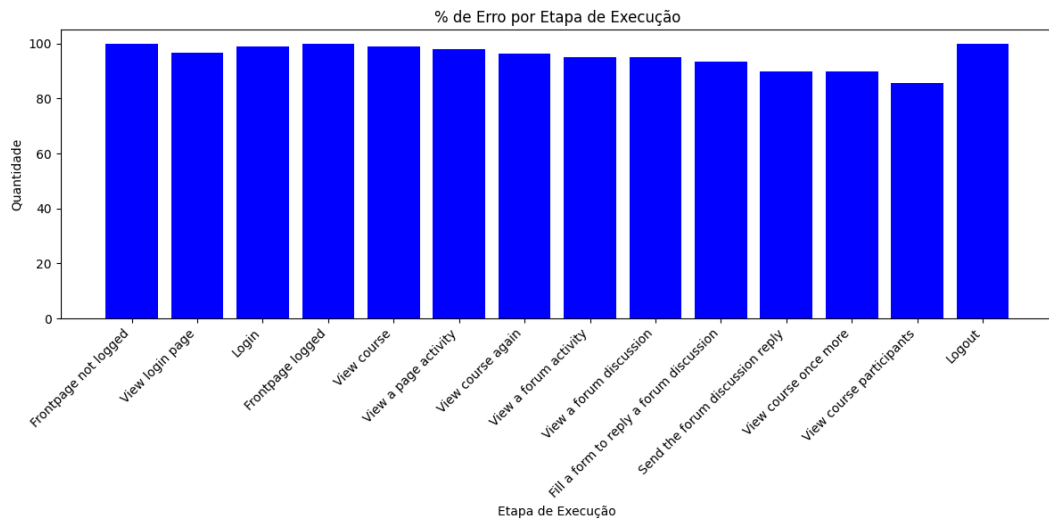
Figura 41 – Grafico de requisições por etapas - quinta execução do primeiro servidor.



Fonte: Elaborado pelo autor.

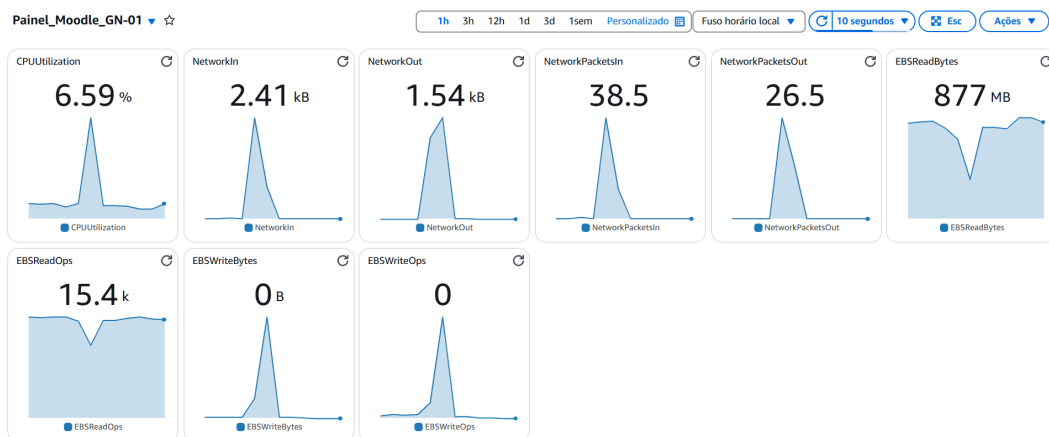
Análise das métricas da *Amazon CloudWatch* na Figura 43 demonstra que o consumo de CPU neste teste apresentou valores inferiores em comparação ao teste anterior, atingindo aproximadamente 42% de utilização no pico máximo, seguido por um declínio gradual que se manteve relativamente estável até o término do teste. Esta redução no consumo sugere a ocorrência de um gargalo de desempenho no servidor. Entretanto, ao examinar as métricas *NetworkIn* e *NetworkPacketsIn*, observa-se que o volume de dados recebidos pelo servidor foi significativamente maior que no teste precedente. Adicionalmente, a carga de trabalho aplicada não foi suficiente para causar a indisponibilidade completa do servidor, mantendo-o acessível durante todo o período de teste.

Figura 42 – Grafico de erros por etapas - quinta execução do primeiro servidor.



Fonte: Elaborado pelo autor.

Figura 43 – Grafico referente as metricas analisadas do teste do Moodle - quinta execução do primeiro servidor.



Fonte: Elaborado pelo autor.

## 6.2 Execução do segundo servidor do Moodle

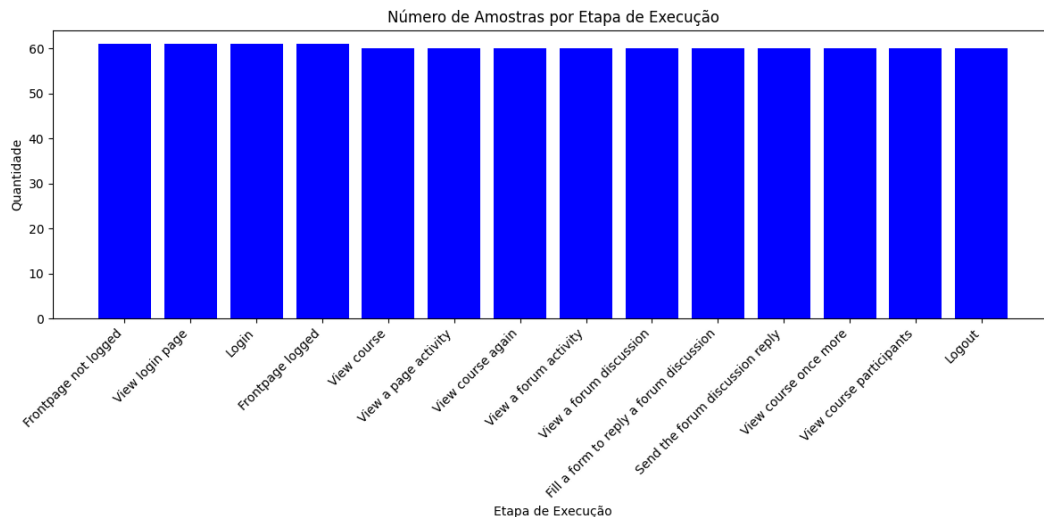
Nesse parte será feita a análise das métricas referente ao primeiro servidor do Moodle que estar instalado na instância do M6a do serviço EC2 da AWS.

### 6.2.1 Primeira execução

O primeiro teste do segundo servidor foi realizado no dia 13 de fevereiro, das 20:00 às 20:15, com aplicação de uma carga de 30 *threads* distribuídas em 5 loops totalizando 150 *threads*. Conforme demonstrado na Figura 44, o sistema utilizou todo o tempo disponível para processar 61 *threads* (40,7% da carga total estabelecida). A análise dos resultados da Figura 45

revelou a ausência de falhas durante a execução, com todas as *threads* sendo processadas com sucesso em suas respectivas etapas, indicando estabilidade do servidor sob as condições de teste aplicadas.

Figura 44 – Grafico de requisições por etapas - primeira execução do segundo servidor.



Fonte: Elaborado pelo autor.

Figura 45 – Grafico de erros por etapas - primeira execução do segundo servidor.



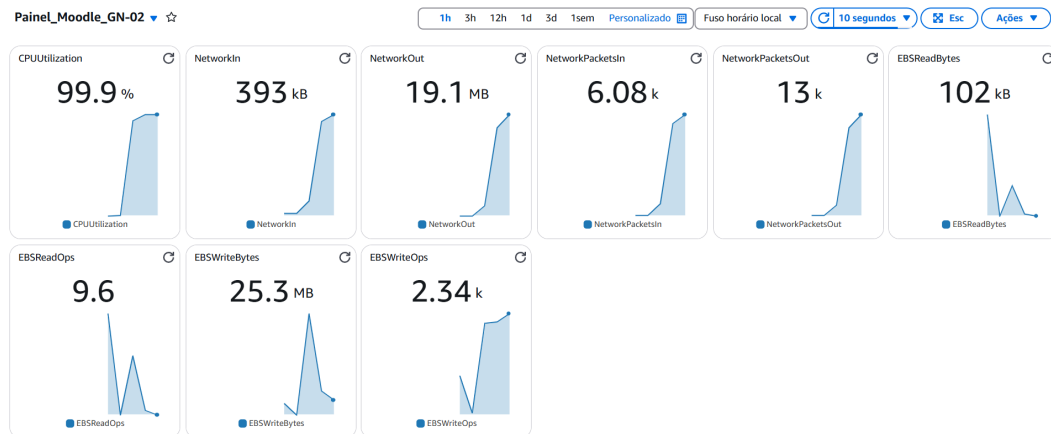
Fonte: Elaborado pelo autor.

Também é possível observar através do gráfico da Figura 46 das métricas da *Amazon CloudWatch*, em particular o parâmetro *CPUUtilization*, revela que a utilização da CPU atingiu aproximadamente 100%, indicando que a carga de trabalho aplicada foi suficiente para saturar os recursos do servidor durante um curto período. Observa-se inicialmente uma variação na utilização, seguida por um patamar constante de alto consumo. Exame das métricas *NetworkOut* e *NetworkPacketsOut* demonstra que o volume de dados transmitidos excedeu o tráfego de entrada,



confirmando que as requisições do plano de testes foram recebidas pelo servidor. Apesar da carga extrema, o servidor manteve estabilidade operacional durante todo o teste, permanecendo acessível imediatamente após sua conclusão.

Figura 46 – Grafico referente as metricas analisadas do teste do Moodle - primeira execução do segundo servidor.



Fonte: Elaborado pelo autor.

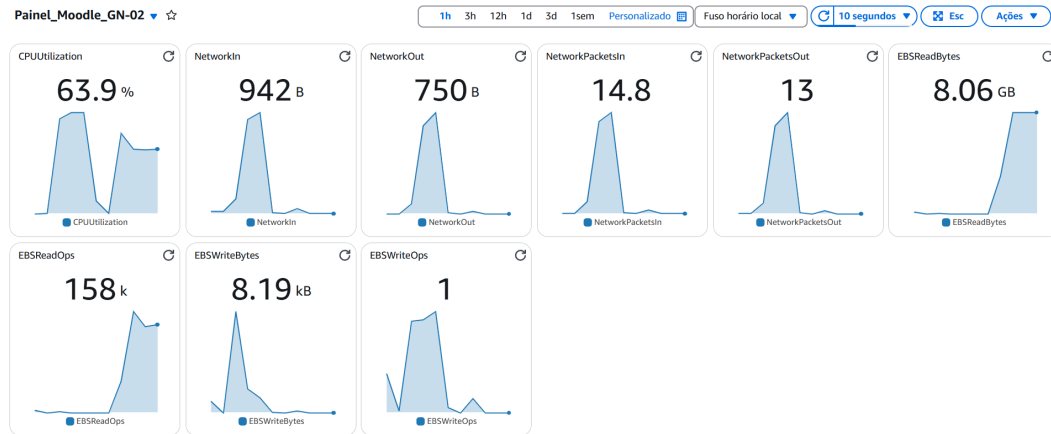
## 6.2.2 Segunda execução

O segundo teste foi conduzido entre 20:25 e 20:45, após um intervalo de 10 minutos do teste anterior. A carga aplicada consistiu em 100 *threads* distribuídas em cinco loops, totalizando 500 *threads*. O sistema utilizou todo o tempo disponível para execução, porém ocorreu uma anomalia não identificada que impossibilitou a mensuração precisa do número de *threads* efetivamente processadas. A replicação do teste em momentos posteriores não resultou em atualizações nos *listeners* configurados no JMeter, impossibilitando também a quantificação de eventuais falhas. Esses resultados sugerem que a causa do problema pode estar relacionada à carga aplicada ou à configuração da instância do servidor.

O gráfico da Figura 47 das métricas do *Amazon CloudWatch* revela que o consumo de CPU apresentou um comportamento moderado e estável durante a maior parte do teste, com exceção de um pico momentâneo entre 75% e 80% de utilização. Contudo, a carga aplicada impactou significativamente o desempenho do servidor. Observou-se também uma anomalia no fluxo de dados: inicialmente, o volume de tráfego de saída foi inferior ao de entrada, indicando funcionamento adequado nas primeiras etapas. Entretanto, após alguns minutos, ocorreu uma interrupção completa da comunicação, resultando na indisponibilidade total do servidor, que

permaneceu inacessível mesmo após o término do teste.

Figura 47 – Grafico referente as metricas analisadas do teste do Moodle - segunda execução do segundo servidor.



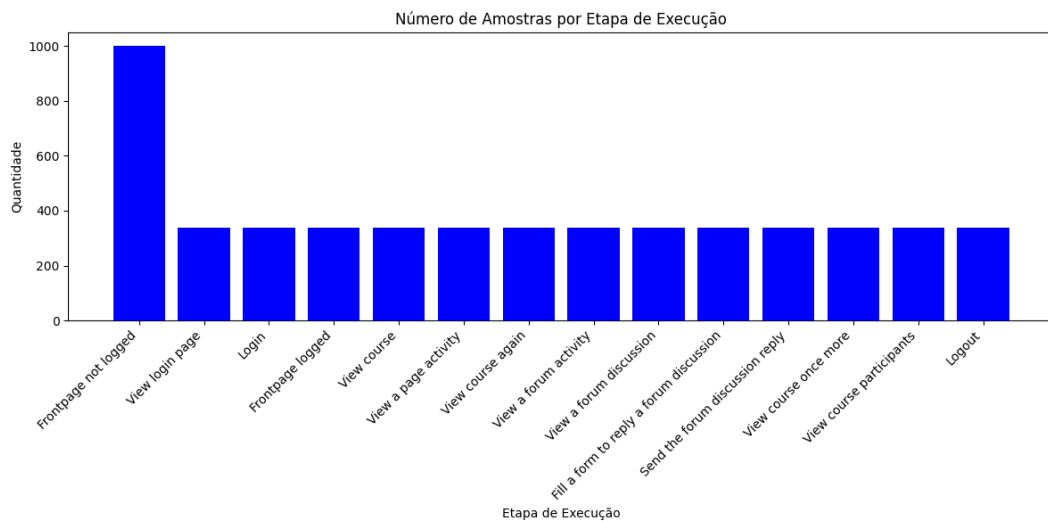
Fonte: Elaborado pelo autor.

### 6.2.3 Terceira execução

O terceiro teste foi conduzido entre 21:05 e 21:30, após o intervalo programado de 10 minutos entre os testes. Durante a execução, ocorreu uma interrupção no Apache JMeter, exigindo reinicialização do processo e aguardando o término da carga falha para evitar anomalias nos dados. A carga aplicada consistiu em 1.000 *threads* em um único *loop*, utilizando todo o tempo disponível para execução. Conforme demonstrado na Figura 48, das 1.000 *threads* executadas, apenas 338 completaram a maioria das etapas, sendo que somente uma concluiu todo o processo. A análise da Figura 49 revela uma taxa de erro aproximada de 100% em quase todas as etapas, com exceção da etapa '*view login page*', que apresentou taxa de 99,71%, indicando que algumas *threads* foram executadas com sucesso nesta fase específica.

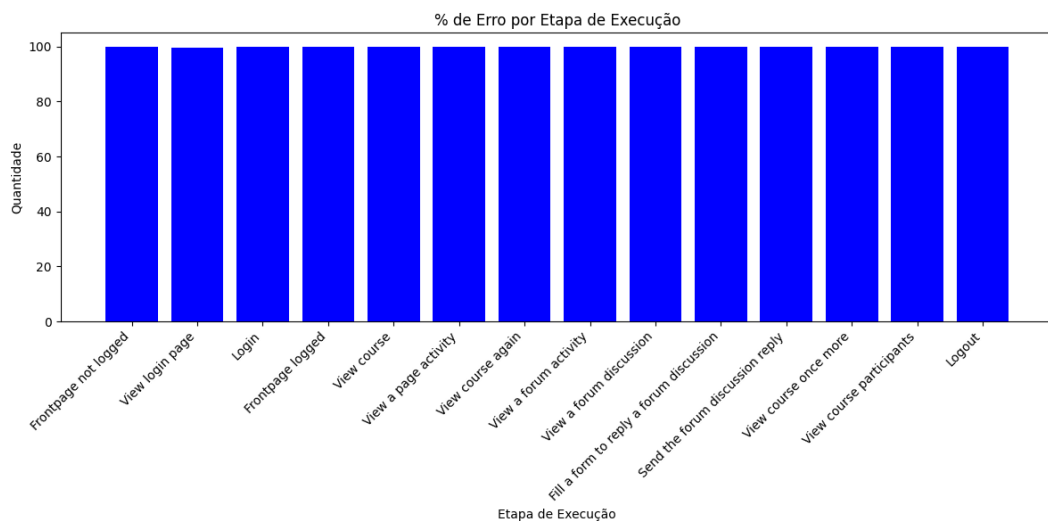
Também pode-se notar observando o gráfico da Figura 50 das métricas do *Amazon CloudWatch* revela que o consumo de CPU do servidor durante o teste atingiu 89,3%, após um período inicial de declínio. O padrão observado mostra que, após alcançar o pico de utilização, ocorreu um declínio constante até o final do teste, sugerindo a ocorrência de um gargalo que comprometeu o envio de dados quando a capacidade máxima foi atingida. Exame das quatro métricas subsequentes demonstra que o volume de dados transmitidos entre as interfaces da instância superou o tráfego de entrada, embora com significativas variações. Este comportamento indica que apenas uma parcela mínima dos dados foi transmitida com sucesso. Embora a carga

Figura 48 – Grafico de requisições por etapas - terceira execução do segundo servidor.



Fonte: Elaborado pelo autor.

Figura 49 – Grafico de erros por etapas - terceira execução do segundo servidor.



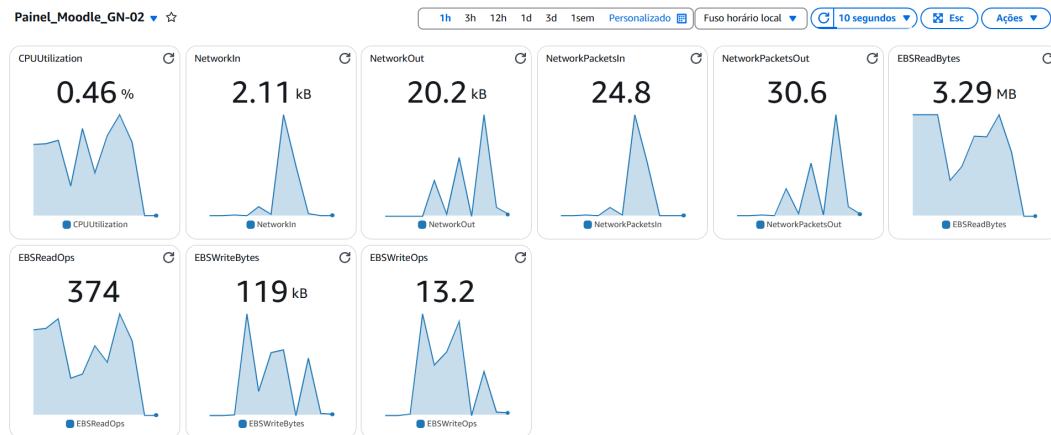
Fonte: Elaborado pelo autor.

aplicada tenha causado um gargalo considerável na transmissão de dados, ela não foi suficiente para provocar a indisponibilidade completa do servidor.

#### 6.2.4 Quarta execução

O quarto teste foi conduzido entre 21:40 e 22:10, após um intervalo de 10 minutos do teste anterior. A carga aplicada consistiu em 1.000 *threads* distribuídas em dois *loops*, totalizando 2.000 *threads* programadas para execução em 30 minutos. Conforme demonstrado na Figura 51, o sistema utilizou todo o tempo disponível para processar parcialmente a carga, com a maior concentração de execução na primeira etapa, padrão similar aos testes anteriores. A análise dos resultados da Figura 52 revelou que todas as *threads* executadas, aproximadamente 676 nas

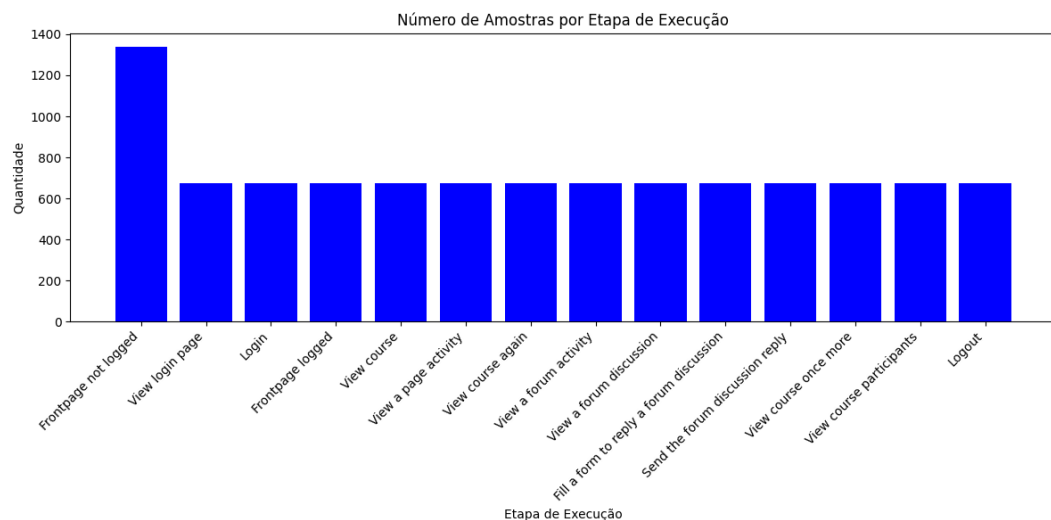
Figura 50 – Grafico referente as metricas analisadas do teste do Moodle - terceira execução do segundo servidor.



Fonte: Elaborado pelo autor.

etapas subsequentes apresentaram taxa de erro de 100% em todas as fases do processo, indicando falha completa na execução das requisições.

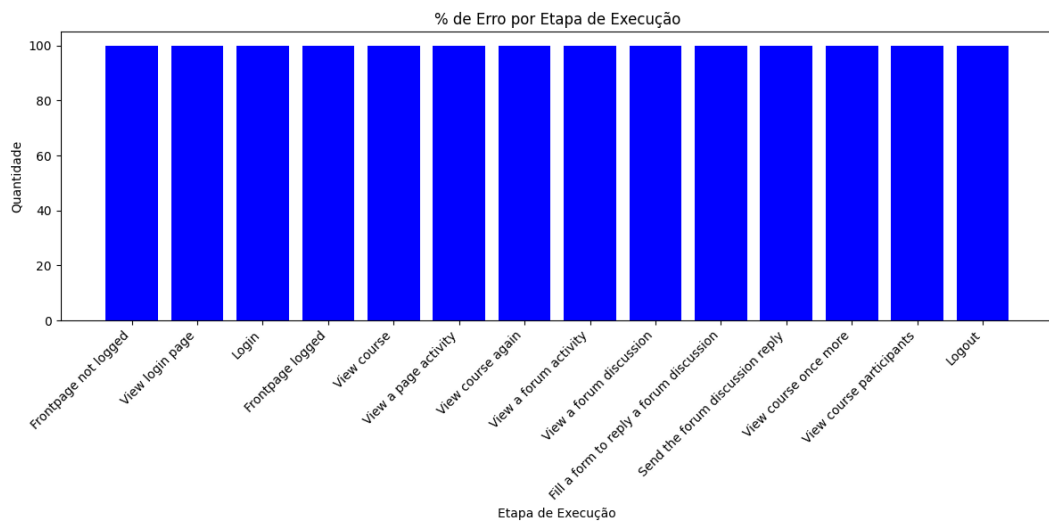
Figura 51 – Grafico de requisições por etapas - quarta execução do segundo servidor.



Fonte: Elaborado pelo autor.

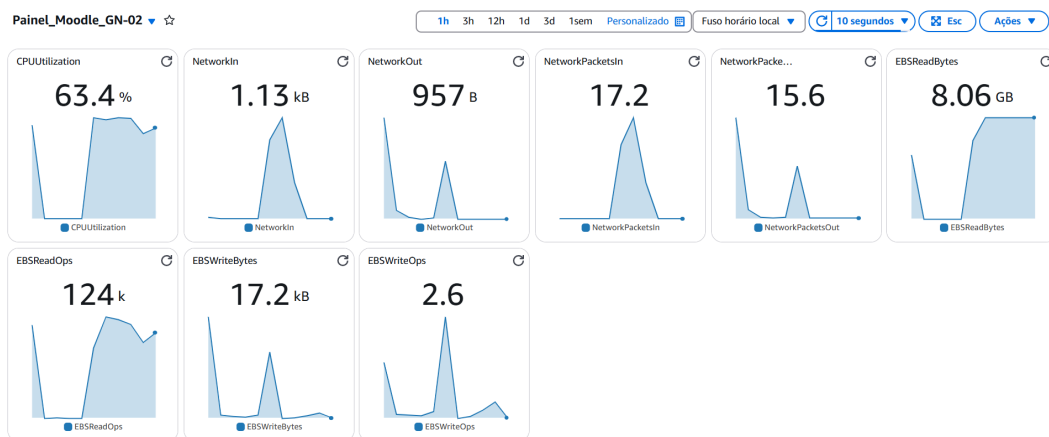
Análise da métrica da *Amazon CloudWatch* na Figura 53 mostra um comportamento estável da utilização da CPU durante a execução do teste, com variação máxima de 6,96% entre o pico de consumo e o final do período de testes. Esta estabilidade sugere que o procedimento foi executado conforme o planejado. Contudo, a avaliação das métricas subsequentes demonstra que o volume de dados transmitidos ao servidor excedeu significativamente os dados efetivamente processados, indicando perda durante a transmissão. Esta anomalia comprometeu a execução das etapas programadas. Adicionalmente, a carga aplicada resultou na indisponibilidade completa do servidor, tornando-o inacessível após a conclusão do teste.

Figura 52 – Grafico de erros por etapas - quarta execução do segundo servidor.



Fonte: Elaborado pelo autor.

Figura 53 – Grafico referente as metricas analisadas do teste do Moodle - quarta execução do segundo servidor.



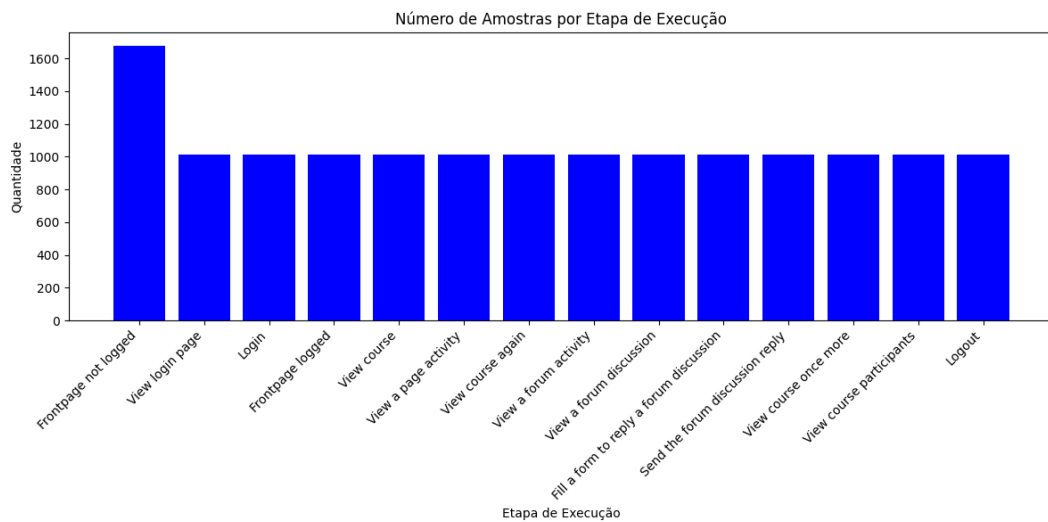
Fonte: Elaborado pelo autor.

### 6.2.5 Quinta execução

O quinto teste foi conduzido entre 22:20 e 22:55, após o intervalo do teste anterior. A carga aplicada consistiu em 1.000 *threads* distribuídas em três *loops*, totalizando 3.000 *threads* programadas para execução em 35 minutos. Conforme demonstrado na Figura 54, o sistema utilizou todo o tempo disponível para processamento parcial da carga, com padrão similar aos testes anteriores: a primeira etapa concentrou mais da metade das execuções, enquanto as demais etapas processaram aproximadamente um terço da carga. A análise dos resultados da Figura 55 revelou taxa de erro de 100% em todas as etapas, indicando falha completa na execução das requisições desde a fase inicial, o que comprometeu todo o processo subsequente.

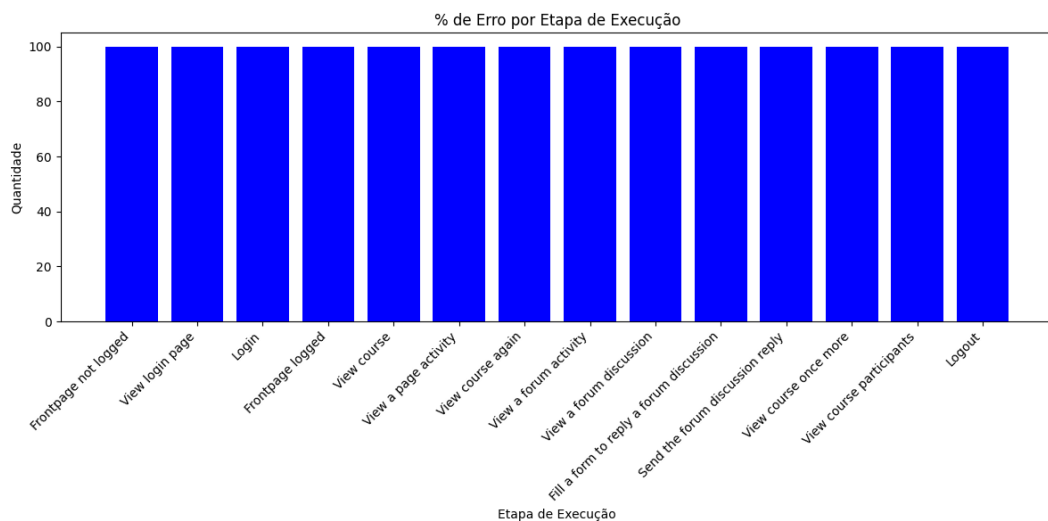
No gráfico da Figura 56 das métricas do Amazon CloudWatch revela que a utilização

Figura 54 – Grafico de requisições por etapas - quinta execução do segundo servidor.



Fonte: Elaborado pelo autor.

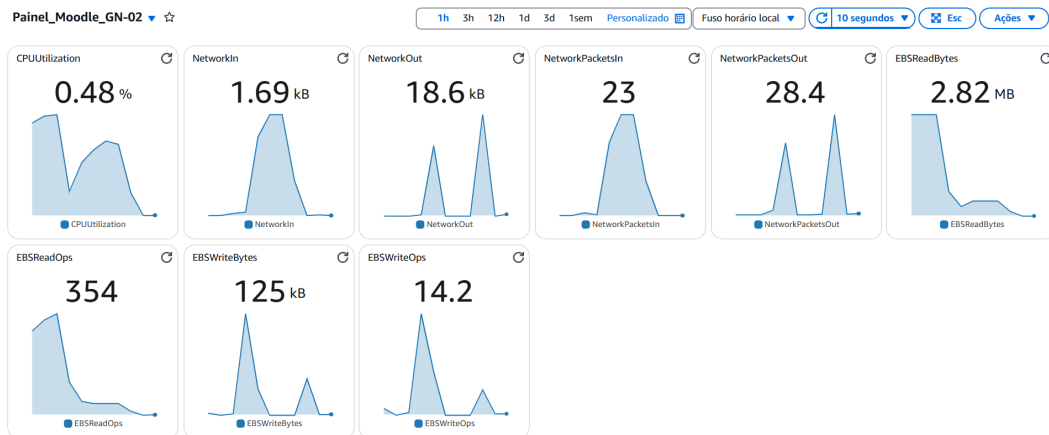
Figura 55 – Grafico de erros por etapas - quinta execução do segundo servidor.



Fonte: Elaborado pelo autor.

da CPU apresentou pico máximo entre 45% e 50% por breve período, seguido por declínio constante até o término do teste. Considerando o volume de *threads* processadas, esse comportamento sugere a ocorrência de um gargalo de desempenho, hipótese corroborada pela elevada taxa de falhas registradas. Observou-se ainda instabilidade significativa no fluxo de dados, com frequentes quedas de desempenho ao longo do período de teste. Embora a carga aplicada tenha causado limitações operacionais, conforme demonstrado pelo gargalo identificado, ele não foi suficiente para comprometer completamente a disponibilidade do servidor.

Figura 56 – Grafico referente as metricas analisadas do teste do Moodle - quinta execução do segundo servidor.



Fonte: Elaborado pelo autor.

### 6.3 Execução do terceiro servidor do Moodle

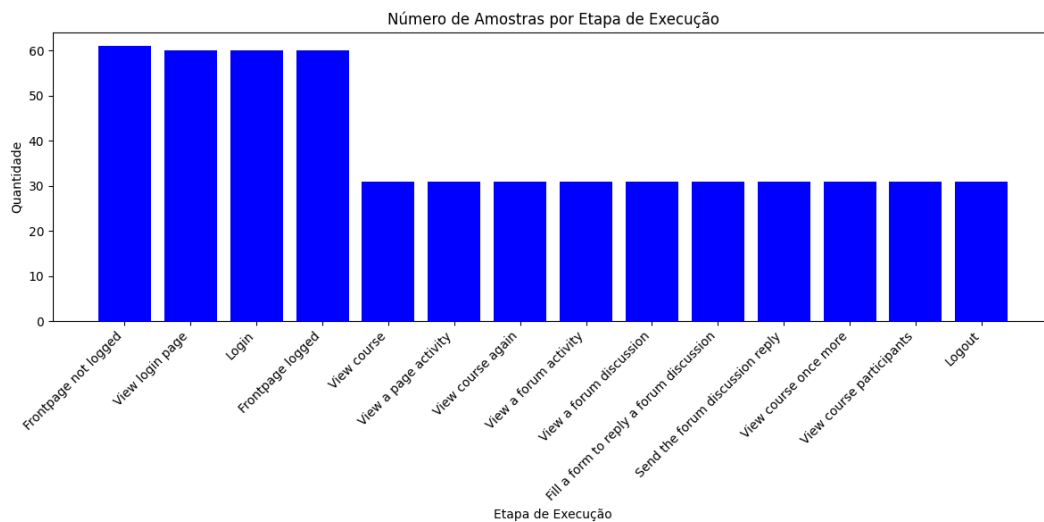
Nesse parte será feita a análise das métricas referente ao primeiro servidor do Moodle que estar instalado na instância do tipo M5n do serviço EC2 da AWS.

#### 6.3.1 Primeira execução

Os testes do terceiro servidor foram realizados no dia 14 de fevereiro, no período das 20:00 às 20:15. A carga aplicada consistiu em 150 *threads* distribuídas em 5 *loops* de 30 *threads* cada. Embora tenha utilizado todo o tempo alocado, o sistema processou menos da metade da carga total, com a maioria das etapas executando apenas um *loop* completo e apenas quatro etapas concluindo dois *loops*, conforme demonstrado na Figura 57. Durante a execução, identificaram-se falhas específicas nas requisições das *threads*, particularmente nas etapas '*Frontpage not logged*' e '*Frontpage logged*', resultando em uma taxa total de erro de 5,44% conforme observado na figura 58. Este resultado indica que mais de 90% da carga foi processada com sucesso, demonstrando relativa estabilidade do servidor sob as condições de teste aplicadas.

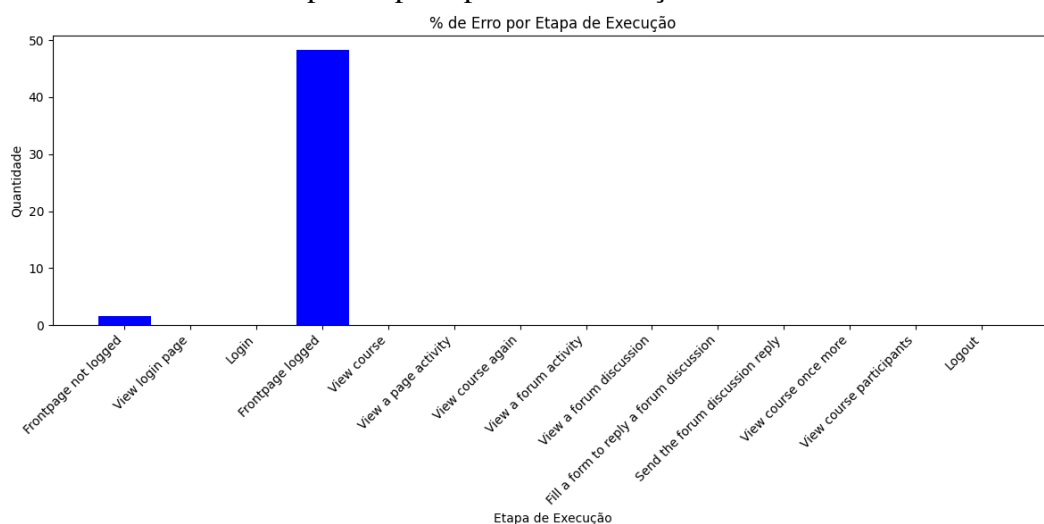
Também pode-se observar através do gráfico da Figura 59 das métricas do Amazon CloudWatch revela que a utilização da CPU atingiu 100% após crescimento constante durante o teste, indicando ocorrência de um gargalo de desempenho. Embora a carga aplicada tenha causado falhas pontuais, estas não comprometeram significativamente a execução do teste. Exame das quatro métricas subsequentes demonstra que o volume de dados transmitidos pelas interfaces

Figura 57 – Grafico de requisições por etapas - primeira execução do terceiro servidor.



Fonte: Elaborado pelo autor.

Figura 58 – Grafico de erros por etapas - primeira execução do terceiro servidor.



Fonte: Elaborado pelo autor.

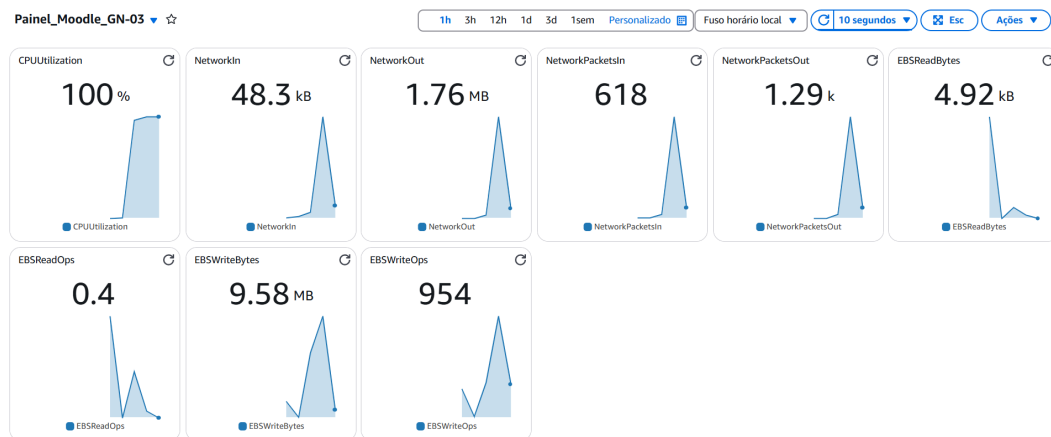
de rede superou o tráfego de entrada, sugerindo eficiência na transferência de dados. Conclui-se que a carga de trabalho provocou limitações operacionais moderadas, porém insuficientes para causar indisponibilidade completa do servidor.

### 6.3.2 Segunda execução

O segundo teste foi conduzido entre 20:25 e 20:45, com duração programada de 20 minutos. A carga aplicada consistiu em 500 *threads* distribuídas em 5 *loops* de 100 *threads* cada. O sistema utilizou todo o tempo disponível, porém, de forma análoga ao segundo teste do servidor anterior, ocorreu uma anomalia no Apache JMeter que resultou em travamento da aplicação, impossibilitando a mensuração precisa da carga efetivamente processada e dos



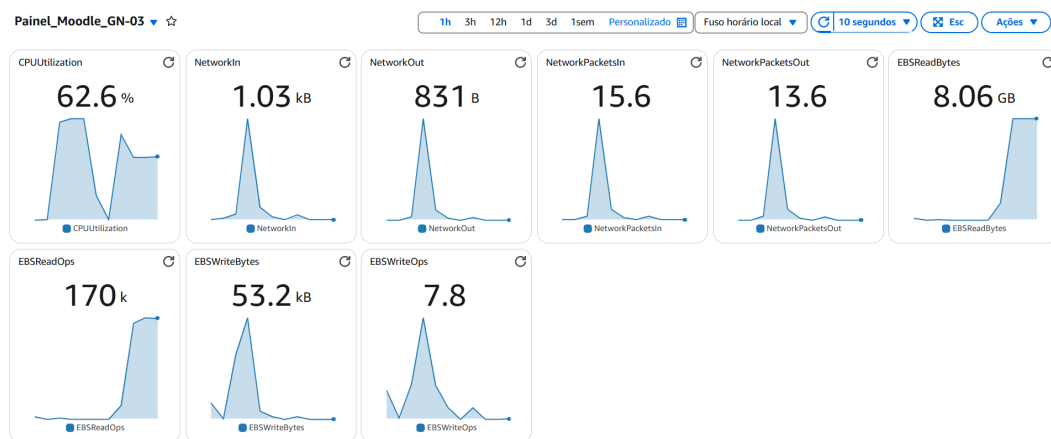
Figura 59 – Grafico referente as metricas analisadas do teste do Moodle - primeira execução do terceiro servidor.



Fonte: Elaborado pelo autor.

eventuais erros nas *threads*. Testes adicionais realizados no dia seguinte após a execução inicial, sob as mesmas condições, reproduziram os mesmos resultados, sugerindo que a origem do problema possa estar relacionada à carga de trabalho aplicada ou ao cenário de teste configurado, embora sua identificação precisa tenha se mostrado inviável.

Figura 60 – Grafico referente as metricas analisadas do teste do Moodle - segunda execução do terceiro servidor.



Fonte: Elaborado pelo autor.

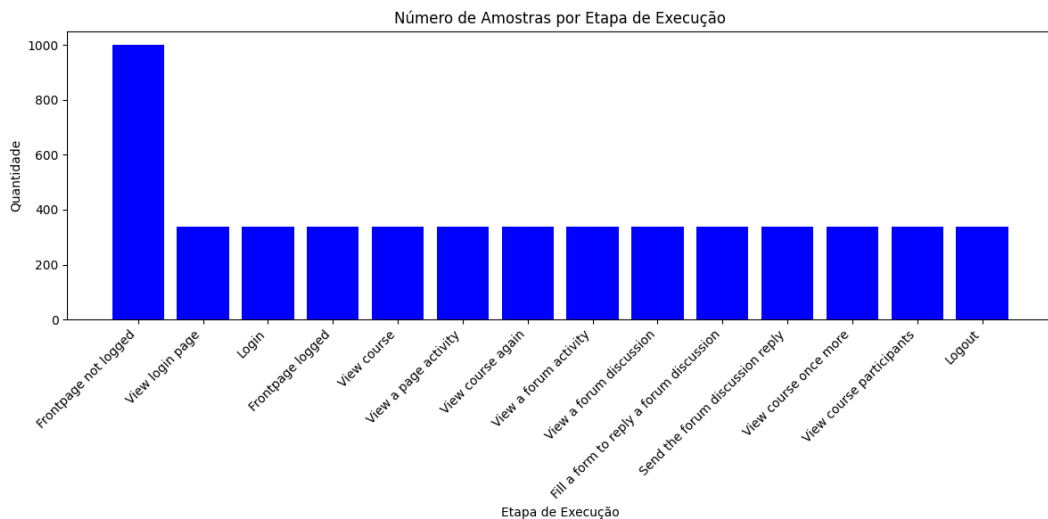
Análise das métricas do Amazon CloudWatch da Figura 60 revela que o servidor apresentou elevada utilização da CPU, com pico entre 75% e 85% de sua capacidade total. Esse nível de consumo indica que o servidor foi submetido a carga significativa, o que comprometeu severamente a transmissão de dados durante as etapas do teste. Observa-se que o volume de dados transmitidos entre as interfaces de rede foi inferior ao recebido pela instância, conforme demonstrado pelas métricas subsequentes. Após breve período inicial de transferência de dados,

ocorreu interrupção completa da comunicação, sugerindo que a carga aplicada provocou a indisponibilidade total do servidor, tornando-o inacessível.

### 6.3.3 Terceira execução

O terceiro teste foi conduzido entre 20:55 e 21:20, com duração de 25 minutos para execução de uma carga de 1.000 *threads* em um único *loop*. Ao término do teste, apenas 338 *threads* foram processadas na maioria das etapas, com exceção da primeira etapa que completou todas as requisições conforme demonstrado na Figura 61. A análise dos resultados revelou taxa de erro de 100% em todas as etapas como mostrado na Figura 62, indicando falha completa no acesso ao servidor Moodle por parte das threads, sem nenhum caso de execução bem-sucedida.

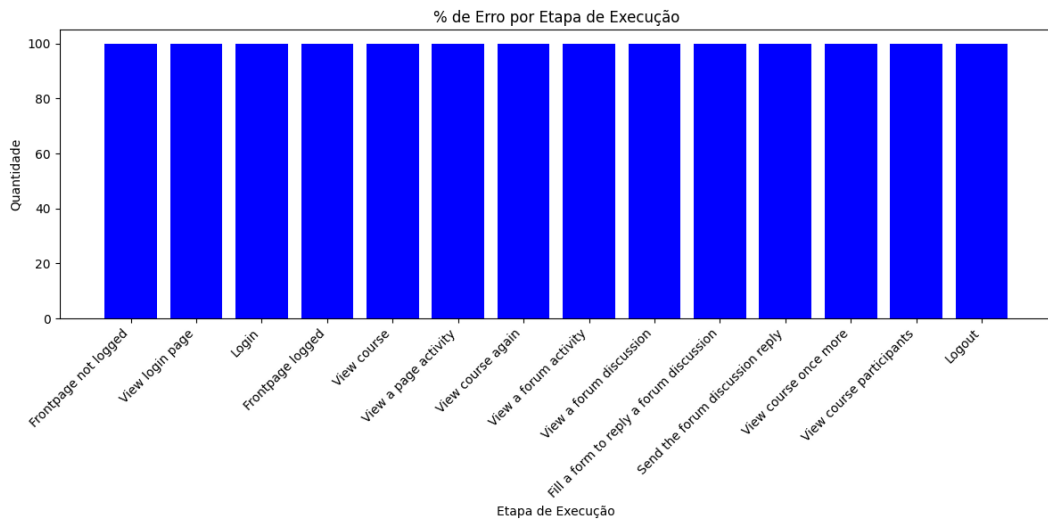
Figura 61 – Grafico de requisições por etapas - terceira execução do terceiro servidor.



Fonte: Elaborado pelo autor.

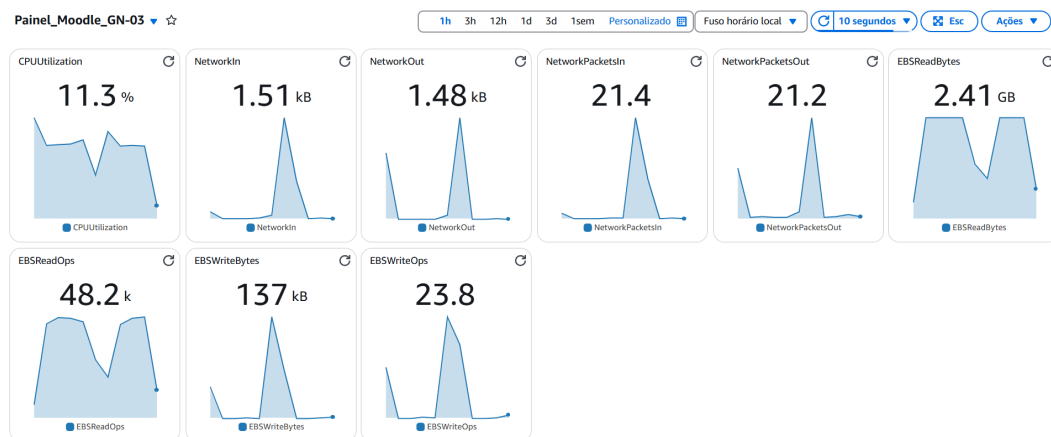
No gráfico da Figura 63 das métricas do *Amazon CloudWatch* revela que o consumo de CPU atingiu aproximadamente 75% em seu pico máximo, valor considerado elevado considerando a carga de trabalho aplicada durante o período de teste. Observou-se um padrão de comportamento distinto: após o pico inicial, a utilização manteve-se estável por cerca de metade do tempo de execução, seguido por declínio acentuado até atingir 11,3% de consumo. Este comportamento sugere a ocorrência de um gargalo de desempenho durante a transmissão de dados, hipótese corroborada pela análise das métricas relacionadas, que apresentaram um único fluxo de dados seguido por queda abrupta. O padrão observado indica que o envio concentrado de dados para a interface de rede da instância excedeu provavelmente a capacidade do servidor, resultando em sua indisponibilidade completa.

Figura 62 – Grafico de erros por etapas - terceira execução do terceiro servidor.



Fonte: Elaborado pelo autor.

Figura 63 – Grafico referente as metricas analisadas do teste do Moodle - terceira execução do terceiro servidor.



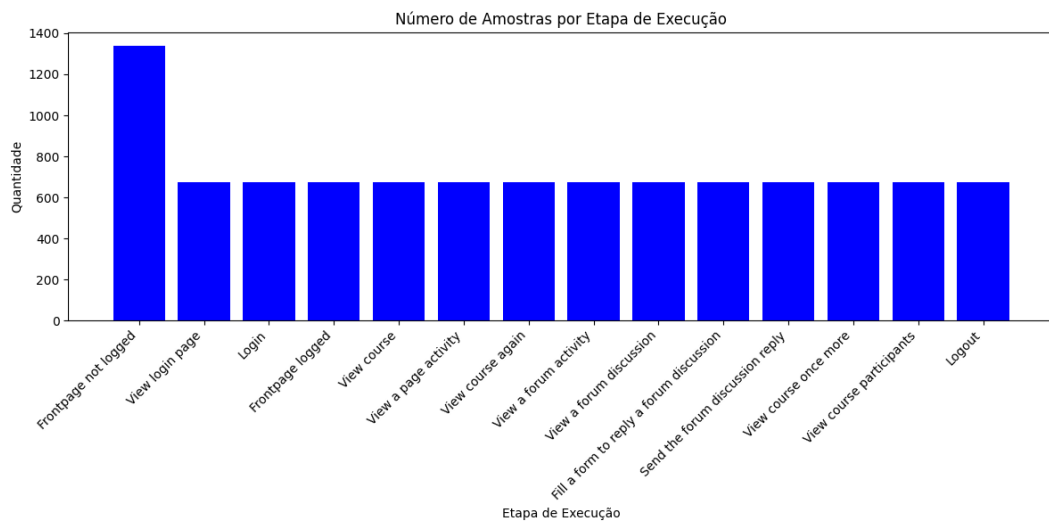
Fonte: Elaborado pelo autor.

#### 6.3.4 Quarta execução

O quarto teste foi conduzido entre 21:35 e 22:05, após intervalo de 15 minutos do teste anterior. A carga aplicada consistiu em 2.000 *threads* distribuídas em dois *loops* de 1.000 *threads* cada, com duração programada de 30 minutos. Conforme demonstrado na Figura 64, o sistema utilizou todo o tempo disponível, processando aproximadamente 25% da carga na maioria das etapas, exceto na primeira etapa que executou 1.338 *threads* (cerca de 67% da carga total). A análise dos resultados da Figura 65 revelou taxa de erro de 100% em todas as etapas, indicando falha completa na transmissão dos dados, sem que nenhuma thread tenha alcançado o servidor Moodle com sucesso.

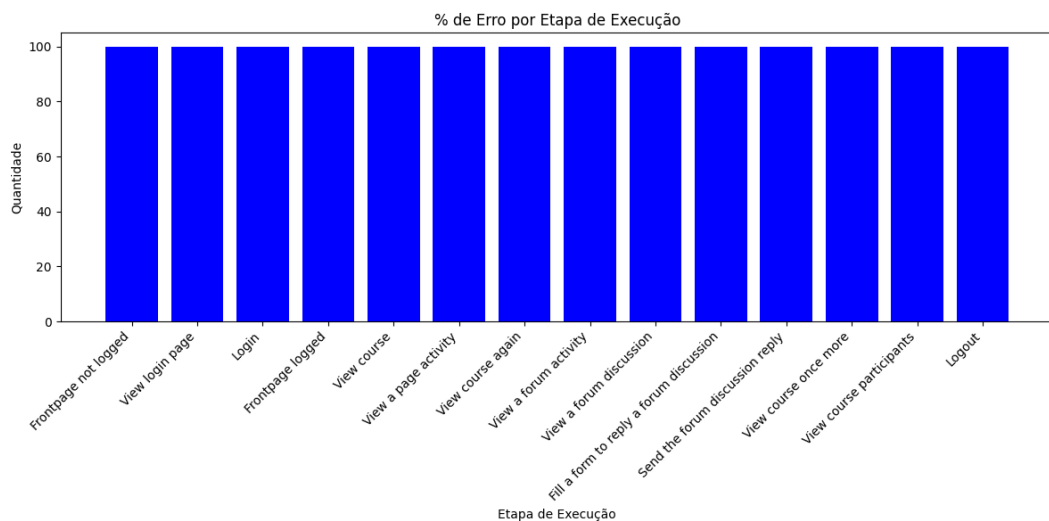
Análise das métricas do Amazon CloudWatch na Figura 66 revela que o consumo de

Figura 64 – Grafico de requisições por etapas - quarta execução do terceiro servidor.



Fonte: Elaborado pelo autor.

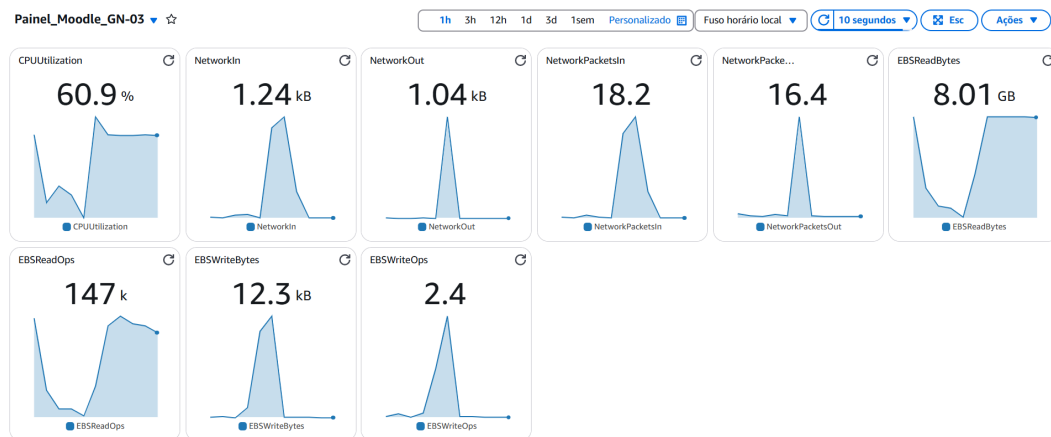
Figura 65 – Grafico de erros por etapas - quarta execução do terceiro servidor.



Fonte: Elaborado pelo autor.

CPU no servidor atingiu 74,5% no início do teste, estabilizando-se posteriormente em 60,9% até o término da execução. Este padrão sugere que a carga de trabalho manteve o servidor operando de forma relativamente estável. Contudo, a avaliação das demais métricas demonstra que a transferência de dados ocorreu em único momento do teste, padrão similar aos observados em testes anteriores. Conforme evidenciado na Figura 65, a ocorrência de 100% de erros indica a presença de um gargalo de desempenho que resultou na perda total dos dados e, finalmente, na indisponibilidade completa do servidor ao final do teste.

Figura 66 – Grafico referente as metricas analisadas do teste do Moodle - quarta execução do terceiro servidor.

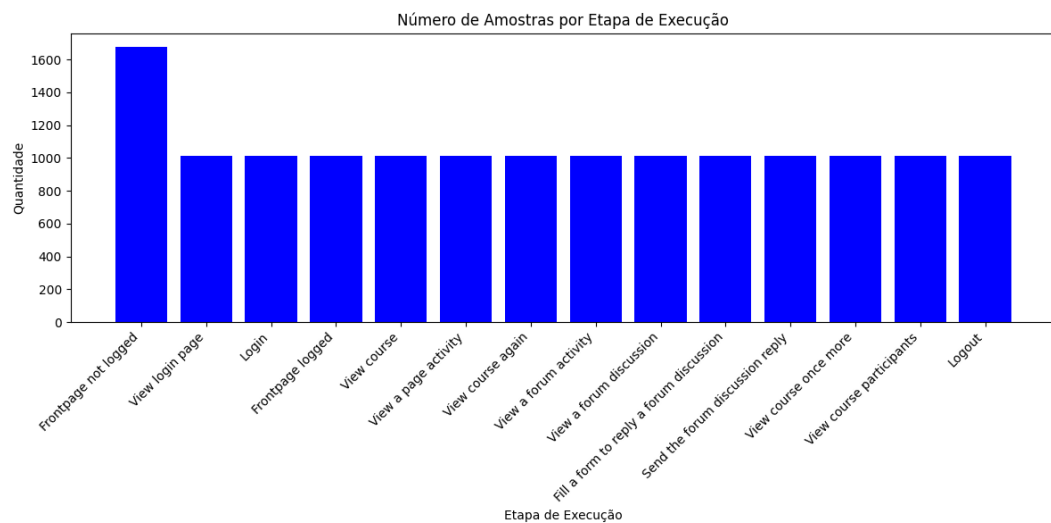


Fonte: Elaborado pelo autor.

### 6.3.5 Quinta execução

O quinto teste foi conduzido entre 22:15 e 22:55, após um intervalo de 10 minutos do teste anterior. A carga aplicada consistiu em 3.000 *threads* distribuídas em três *loops* de 1.000 *threads* cada, com tempo de execução programado de 35 minutos. Conforme demonstrado na Figura 67, o sistema utilizou todo o tempo disponível, processando aproximadamente 33% da carga na maioria das etapas, enquanto a primeira etapa executou cerca de 67% do total. A análise dos resultados da Figura 68 revelou taxa de erro de 100% em todas as etapas para as *threads* que foram processadas, indicando falha completa na execução das requisições.

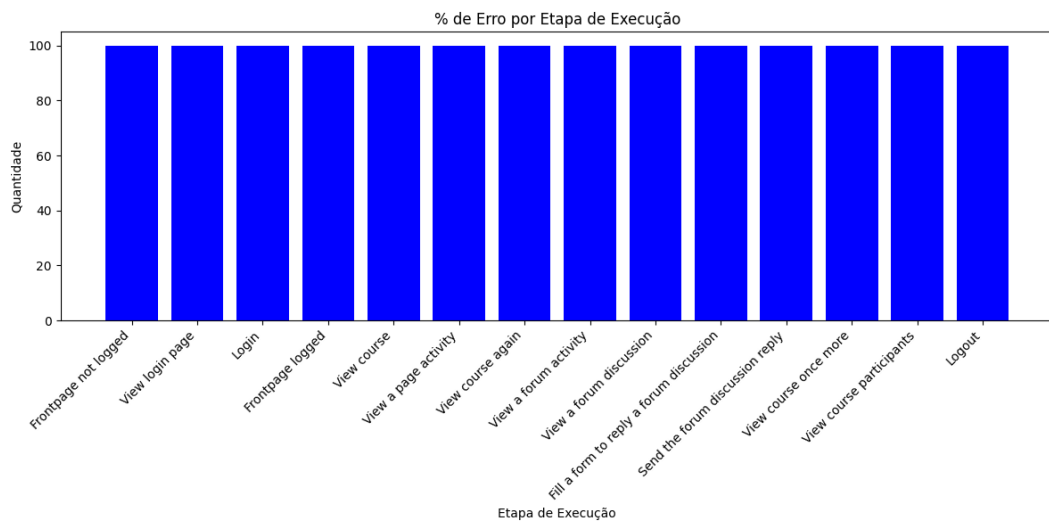
Figura 67 – Grafico de requisições por etapas - quinta execução do terceiro servidor.



Fonte: Elaborado pelo autor.

O gráfico da Figura 69 das métricas da *Amazon CloudWatch* revela que o consumo

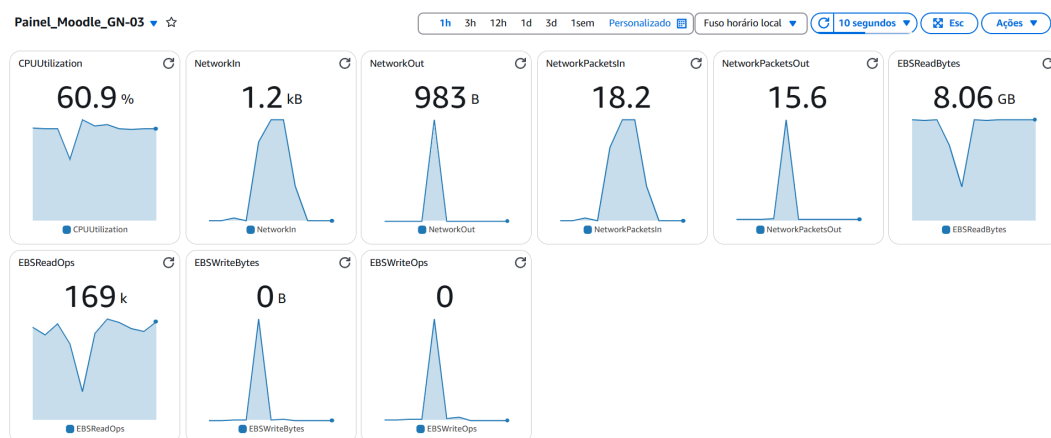
Figura 68 – Grafico de erros por etapas - quinta execução do terceiro servidor.



Fonte: Elaborado pelo autor.

de CPU atingiu 66,9% no pico inicial do teste, mantendo-se nesse patamar por breve período. Posteriormente, registraram-se dois momentos de instabilidade antes de estabilizar em 60,9% até o término do teste. Considerando a carga de trabalho aplicada, o consumo de recursos manteve-se em níveis moderados. Entretanto, as demais métricas apresentaram padrão similar aos testes anteriores, com transmissão de dados concentrada em único momento, seguida por interrupção completa. Este comportamento sugere a ocorrência de um gargalo de desempenho que resultou na perda dos dados. Adicionalmente, a carga aplicada provocou a indisponibilidade total do servidor imediatamente após o término do teste.

Figura 69 – Grafico referente as metricas analisadas do teste do Moodle - quinta execução do terceiro servidor.



Fonte: Elaborado pelo autor.

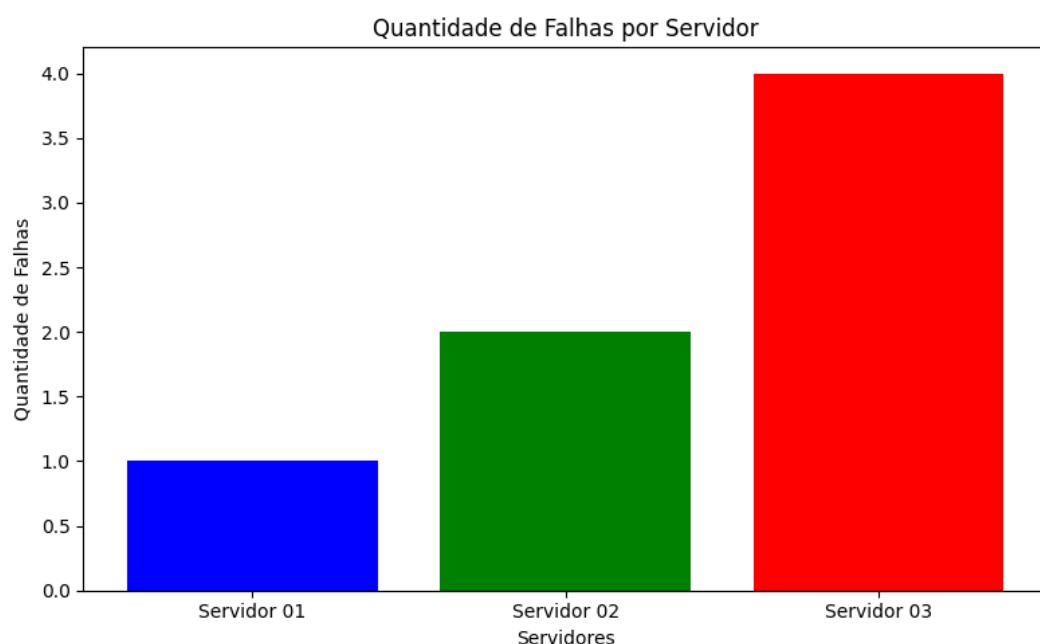
## 6.4 Discussão

Após análise dos resultados obtidos mediante a execução dos testes, foi possível extrair informações relevantes tanto dos *listeners* do Apache JMeter quanto dos painéis de métricas do *Amazon CloudWatch*. Os dados demonstram que o servidor Moodle utilizado neste estudo, quando submetido a determinados níveis de carga, apresenta falhas operacionais em seus módulos, culminando na indisponibilidade do serviço.

A análise dos *logs* dos *listeners* do JMeter evidenciou que o primeiro servidor do Moodle apresentou os melhores resultados sob as mesmas condições de teste em comparação aos demais servidores, conforme o plano de testes elaborado. Esse desempenho superior foi observado tanto no número de *threads* executadas com sucesso quanto na menor taxa de indisponibilidade, como ilustrado na Figura 70. Ressalta-se que os servidores 2 e 3 apresentaram falhas durante o teste devido a um problema técnico não identificado, reforçando a conclusão de que o primeiro servidor obteve o melhor desempenho, uma vez que executou o mesmo teste sem intercorrências.

Além disso, conforme demonstrado na figura anterior, o terceiro servidor registrou o pior desempenho em ambos os critérios analisados: *threads* executadas com sucesso e taxa de erro. Esse resultado sugere que, mesmo dentro de um grupo de instâncias configuradas de forma semelhante, nem todas são adequadas para suportar um servidor sob alta demanda de uso.

Figura 70 – Grafico de indisponibilidade dos servidores do Moodle.



Fonte: Elaborado pelo autor.

Além disso, foi possível determinar a capacidade média de usuários que um servidor Moodle, configurado conforme o modelo utilizado no campus da UFC de Quixadá, consegue suportar. Quando submetido a uma carga superior a 500 *threads* (equivalente a 500 usuários simultâneos), o servidor passa a apresentar perda de dados devido a gargalos de desempenho. Esses problemas ocorreram em todos os testes realizados, manifestando-se imediatamente após o envio das primeiras levas de *threads*, assim que a carga se aproximou do limite identificado.

Outro aspecto relevante identificado nos *logs* do JMeter refere-se à latência de cada servidor durante os testes, conforme demonstrado na Figura 71. Como evidenciado nos resultados do primeiro servidor, os testes com 150 e 500 *threads* apresentaram maior variação na latência, com tendência de crescimento progressivo. Essa escalada seria potencialmente mais acentuada caso tivesse sido possível completar a execução total da carga, particularmente no segundo teste.

Os testes três e quatro apresentaram resultados idênticos, caracterizados por elevada latência inicial que diminuiu rapidamente, mantendo-se estável até o final da execução. Esse comportamento ocorreu em função do grande volume de *threads* processadas simultaneamente, ocasionando gargalos de desempenho que interromperam o fluxo de dados e congestionaram o consumo de recursos computacionais. Em contrapartida, o quinto teste demonstrou padrão distinto, apresentando variações contínuas mesmo durante os intervalos programados. Essa anomalia decorreu da sobreposição de consumo de recursos entre o teste anterior e o atual, criando a falsa impressão de escalabilidade. Na realidade, o servidor sofreu sobrecarga progressiva que resultou em indisponibilidade total do sistema.

Em contrapartida, as latências observadas nos outros dois servidores apresentaram características distintas, com resultados quase idênticos que sugerem maior estabilidade operacional. Esta constância contrasta significativamente com a variabilidade apresentada pelo primeiro servidor. Embora ambos os servidores (segundo e terceiro) demonstrassem padrões similares, o segundo servidor obteve desempenho ligeiramente superior. Esta diferença tornou-se particularmente evidente no terceiro teste, onde algumas *threads* foram processadas com sucesso, um resultado que nem mesmo o primeiro servidor, considerado o mais eficiente, conseguiu alcançar em condições equivalentes.

É importante ressaltar que, apesar das variações de latência observadas nos três servidores, a taxa de erro tornou-se extremamente elevada a partir de 1000 *threads*, resultando na execução bem-sucedida de poucas ou nenhuma *thread*. Este fenômeno sugere uma possível correlação entre a estabilidade dos resultados e a incapacidade do sistema de processar cargas



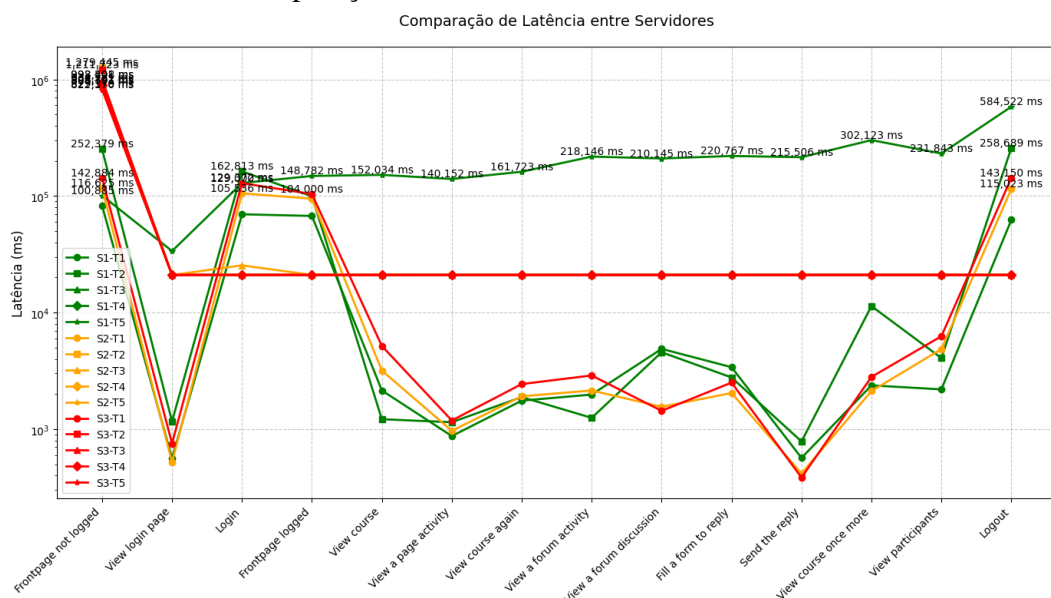
elevadas.

Pode-se concluir que a instância utilizada para hospedar o primeiro servidor obteve os melhores resultados no quadro geral de avaliação. Seu desempenho refletiu consistentemente os parâmetros estabelecidos durante a criação dos testes, nos quais o Moodle simulou todo o cenário operacional, desde a criação de usuários até a execução de atividades, utilizando todos os recursos disponíveis.

Entre os três servidores avaliados, o primeiro destacou-se por concluir os processos com maior rapidez, embora todos tenham sido configurados simultaneamente nas mesmas condições. No entanto, analisando com maior profundidade, verifica-se uma diferença na largura de banda das instâncias: o primeiro e o segundo são idênticos, enquanto no terceiro há uma variação que pode ser a causa da disparidade nos resultados. Isso ocorre devido à velocidade de leitura e gravação de dados, pois, ao executar cargas mais elevadas, é possível que tenham sido ultrapassados os limites suportados pela instância, ocasionando gargalos de desempenho e indisponibilidade do sistema.

Portanto, os resultados indicam que a instância utilizada no primeiro servidor apresenta a configuração mais adequada para hospedar a aplicação Moodle quando considerada a avaliação de desempenho em ambiente de nuvem AWS.

Figura 71 – Grafico de comparação de latências dos servidores do Moodle.



Fonte: Elaborado pelo autor.

Vale lembrar que a avaliação do servidor Moodle não pôde ser conduzida conforme planejado inicialmente devido a limitações impostas por diversos fatores. Primeiramente, as

configurações das instâncias foram restringidas por se tratar de uma conta de nível estudantil. Adicionalmente, a complexidade da aplicação dificultou a implementação de todos os cenários de teste previstos, assim como a documentação disponível se mostrou insuficiente para orientar adequadamente todas as customizações necessárias. Essas limitações serão detalhadamente discutidas no tópico específico destinado a esta finalidade.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões do trabalho, as dificuldades encontradas e sugestões para trabalhos futuros. O estudo teve como objetivo principal avaliar o desempenho de um ambiente virtual de aprendizagem Moodle que é amplamente utilizado no mundo que foi bastante utilizado no período da pandemia de COVID-19, onde ele será hospedado em uma nuvem pública da *Amazon Web Services (AWS)*. A avaliação foi realizada mediante a implementação de um ambiente de testes projetado especificamente para esta finalidade, utilizando cargas de trabalho simuladas através da ferramenta *Apache JMeter*. A análise buscou identificar os impactos causados por diferentes níveis de demanda no ambiente educacional virtual.

### 7.1 Considerações finais

Este estudo teve como objetivo avaliar o desempenho do Moodle, um dos principais ambientes virtuais de aprendizagem utilizados no âmbito educacional, quando hospedado na infraestrutura da *Amazon Web Services (AWS)*, sendo atualmente o maior serviço de nuvem pública em escala global. A avaliação foi conduzida mediante a aplicação de diferentes cargas de trabalho simuladas com a ferramenta *Apache JMeter*, com ênfase especial no parâmetro de disponibilidade do sistema. Sendo esse aspecto particularmente relevante para situações críticas do calendário acadêmico, tais como períodos de realização de avaliações online ou processos de matrícula, quando ocorrem os maiores picos de acesso simultâneo à plataforma.

Os resultados obtidos revelaram que o Moodle sofre impactos significativos quando apresenta um aumento na carga de usuários simultâneos quando é superior a 500 *threads*. O primeiro servidor utilizando uma instância do tipo M7i-flex apresentou o melhor desempenho, com menor taxa de indisponibilidade e maior eficiência no processamento das requisições. Em contraste, os servidores 2 e 3, configurados em instâncias M6a e M5n, respectivamente, mostraram-se menos resilientes, com falhas mais frequentes e maior sensibilidade a gargalos de desempenho. Em cenários com alta demanda, o servidor Moodle tende a saturar seus recursos, resultando em perda de dados e indisponibilidade temporária. Esses resultados destacam a importância de dimensionar adequadamente os recursos computacionais ao hospedar ambientes virtuais de aprendizagem em nuvem, garantindo que a infraestrutura suporte picos de acesso sem comprometer a experiência do usuário.

Além disso, o trabalho evidenciou a viabilidade de utilizar serviços de nuvem pública,

como a AWS, para hospedar aplicações educacionais, oferecendo escalabilidade e flexibilidade. No entanto, a escolha do tipo de instância e a configuração adequada são fatores críticos para otimizar o desempenho e evitar gargalos. A metodologia proposta, que combina testes de carga com monitoramento em tempo real, mostrou-se eficaz para identificar limitações e oportunidades de melhoria em ambientes virtuais de aprendizagem.

Por fim, este estudo contribui para a literatura ao preencher uma lacuna existente na avaliação de desempenho de AVAs em ambientes de nuvem pública, oferecendo *insights*, valiosos para instituições educacionais que desejam migrar ou otimizar seus sistemas de aprendizagem online. Os resultados obtidos podem servir como base para futuras pesquisas e implementações práticas, visando ambientes mais robustos e eficientes.

## **7.2 Limitações do trabalho**

Uma das principais limitações identificadas neste trabalho refere-se à criação das instâncias para hospedagem do servidor Moodle. Em virtude das restrições impostas pela conta de nível estudantil utilizada, houve limitação na alocação de recursos computacionais. Essa condição impossibilitou a configuração inicialmente planejada, que pretendia replicar as especificações do servidor em operação na UFC — Campus Quixadá. Consequentemente, as instâncias implementadas apresentaram configurações inferiores às originalmente projetadas, o que pode ter influenciado significativamente os resultados dos testes de carga mais robustos, particularmente aqueles que demandam maior capacidade de processamento.

Outra limitação significativa deste estudo refere-se à documentação disponível para a aplicação Moodle, que se encontra desatualizada e incompleta. Observou-se que durante o processo de migração para a nova documentação, várias páginas relacionadas à configuração de planos de teste para avaliação de desempenho e aos módulos internos do sistema estavam indisponíveis ou em branco. Adicionalmente, muitos módulos que executam funções similares não estão devidamente configurados após a instalação padrão.

A complexidade do código-fonte do Moodle também representou um obstáculo, impossibilitando a modificação de linhas de código críticas e a implementação de todos os cenários de teste originalmente planejados, como envio de arquivos e resolução de avaliações. Como alternativa, foi necessário utilizar os testes padrão fornecidos pelo próprio Moodle, que, embora mais simples de modificar, não contemplavam toda a abrangência da avaliação inicialmente proposta.

### 7.3 Trabalhos futuros

Como trabalhos futuros, uma sugestão seria a utilização de outros AVAs como Eureka, TelEduc e Blackboard para realizar uma comparação de desempenho entre eles e o AVA utilizado no trabalho sobre as mesmas condições de teste para determinar os limites em que essas aplicações suportam para que possa ser usado como referência para melhorias futuras bem como determinar qual AVA apresenta o melhor resultado para uso prático pensado nesse trabalho.

Outro trabalho futuro seria a utilização de outros ambientes de nuvem publica como, por exemplo, o *Google Cloud Storage*, *Microsoft Azure*, *DigitalOcean* e comparar os resultados com os obtidos na AWS, identificando vantagens e desvantagens de cada provedor dentro das mesmas condições já que cada provedor utilizar os seus próprios tipos de instâncias deve ser estudado os tipos oferecidos para encontrar o meio-termo para ser utilizado nos testes para não haver disparidade nos resultados.

Por último, seria a integração de ferramentas de monitoramento mais avançadas como Prometheus ou Grafana para complementar a coleta de métricas, permitindo uma análise mais detalhada e em tempo real do desempenho do sistema. Dessa forma possibilitando uma análise mais ampla dos resultados obtidos além dos dados obtidos da *CloudWatch* e dos *listeners* do *Apache JMeter*.

## REFERÊNCIAS

- ARRUDA, D. F. de. **Benefícios e Desafios encontrados na adoção de Cloud Computing**. 2016. Disponível em: <https://doceru.com/doc/xnxnnv1>. Acesso em: 22 Julho. 2025.
- BELUCE, A. C. **Ambientes virtuais de aprendizagem**: das estratégias de ensino às estratégias de aprendizagem 2012. v. 10, n. 2, p. 45–60, 2012. Tese (Doutorado) – Universidade Estadual de Londrina, Londrina, 2012.
- BLOG, I. S. **História da computação em nuvem**: como surgiu a cloud computing? IPM Sistemas blog, 2020. Disponível em: <https://www.ipm.com.br/blog/administracao-geral/historia-da-computacao-em-nuvem-como-surgiu-a-cloud-computing/>. Acesso em: 04 Maio. 2023.
- BORGES, H. P.; SOUZA, J. N. d.; SCHULZE, B.; MURY, A. R. **Computação em nuvem** instituto federal de educação, ciência e tecnologia do maranhão. 2011.
- CALLOU, G.; MACIEL, P.; TAVARES, E.; ANDRADE, E.; NOGUEIRA, B.; ARAUJO, C.; CUNHA, P. Energy consumption and execution time estimation of embedded system applications. **Microprocessors and Microsystems**, Elsevier, v. 35, n. 4, p. 426–440, 2011.
- CAMINERO, A.; HERNANDEZ, R.; ROS, S.; TOBARRA, L.; ROBLES-GOMEZ, A.; PASTOR, R. Comparison of lms: Which is the most suitable lms for my needs? **International Journal of Emerging Technologies in Learning (iJET)**, v. 8, n. 2013, 2013.
- CHANDEL, V.; PATIAL, S.; GULERIA, S. Comparative study of testing tools: Apache jmeter and load runner. **International Journal of Computing and Corporate Research**, v. 3, n. 3, p. 1–7, 2013.
- COUTINHO, E. F. **FOLE**: um framework conceitual para avaliação de desempenho da elasticidade em ambientes de computação em nuvem 2014. 151 f. Tese (Doutorado em ciência da computação) – Universidade Federal do Ceará, Fortaleza-CE, 2014.
- EULÁLIO, A. D. *et al.* **Modelo de integração entre ambientes virtuais de aprendizagem e sistemas de gestão acadêmica baseado em arquitetura orientada a serviços e computação em nuvem** 2016. 95 f. Dissertação (Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância) – Universidade Federal Rural de Pernambuco, Recife, 2016.
- FENNER, G. **Um modelo orientado ao negócio para suporte à tomada de decisão multicritério no gerenciamento de capacidade em provedores IaaS** 2019. 258 f. Tese (Doutorado em Ciência da Computação) – Universidade Federal do Ceará, Fortaleza, 2019.
- FERNANDES, C. M. *et al.* **Uma metodologia para avaliação de desempenho de ambientes virtuais de aprendizagem em nuvens computacionais privadas** 2017. 95 f. Dissertação (Programa de Pós-Graduação em Informática Aplicada) – Universidade Federal Rural de Pernambuco, Recife, 2017.
- FIORIO, M.; SILVA, J. da; RIBEIRO, A. Um framework de comunidades de prática em ambientes virtuais de aprendizagem. **RENOTE**, v. 9, n. 1, 2011.
- FRANCISCATO, F. T.; RIBEIRO, P. da S.; MOZZAQUATRO, P. M.; MEDINA, R. D. Avaliação dos ambientes virtuais de aprendizagem moodle, teleduc e tidia-ae: um estudo comparativo. **RENOTE**, v. 6, n. 1, 2008.

- GONZALEZ, M. **Fundamentos da tutoria em educação a distância**. [S. l.]: Avercamp, 2005.
- GUO, X.; SHI, Q.; ZHANG, D. A study on moodle virtual cluster in cloud computing. In: IEEE. **2013 Seventh International Conference on Internet Computing for Engineering and Science**. [S. l.], 2013. p. 15–20.
- JAIN, R. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling**. [S. l.]: john wiley & sons, 1990.
- JMETER. **Apache JMeter**. APACHE SOFTWARE FOUNDATION, 2019. Disponível em: <https://jmeter.apache.org/>. Acesso em: 15 Maio. 2023.
- LILJA, D. J. **Measuring computer performance: a practitioner's guide**. [S. l.]: Cambridge university press, 2005.
- LIMA, C. J. de; SILVA, A. V. G. da; ANDRADE, E. C. de; CALLOU, G. R. de A. Avaliação de desempenho e consumo de energia de um ambiente virtual de aprendizagem em nuvens privadas. **Revista Brasileira de Computação Aplicada**, v. 13, n. 1, p. 88–97, 2021.
- MACHIDA, F.; ANDRADE, E.; KIM, D. S.; TRIVEDI, K. S. Candy: Component-based availability modeling framework for cloud service management using sysml. In: **2011 IEEE 30th International Symposium on Reliable Distributed Systems**. [S. l.: s. n.], 2011. p. 209–218.
- MATAM, S.; JAIN, J. **Pro Apache JMeter: web application performance testing**. [S. l.]: Apress, 2017.
- MATHEW, S.; VARIA, J. Overview of amazon web services. **Amazon Whitepapers**, v. 105, p. 1–22, 2014.
- MELL, P.; GRANCE, T. **The NIST Definition of Cloud Computing**. gaithersburg: National institute of standards and technology. 2011. Disponível em: <https://csrc.nist.gov/publications/detail/sp/800-145/final>. Acesso em: 22 Julho. 2025.
- MEYER, A. I. da S. Ambientes virtuais de aprendizagem: Conceitos e características. **Kiri-Kerê-Pesquisa em Ensino**, v. 1, n. 12, 2022.
- MOODLE, m. **Compartilhe 15 anos de memórias do Moodle 1.0**. 2017. Ur-  
<https://twitter.com/moodle/status/898152671076659200>. Acesso em: 02 Maio. 2023.
- MOODLE.ORG. **The Moodle Story**. Moodle, 2023. Disponível em: <https://moodle.com/pt/sobre/a-moodle-story/>. Acesso em: 02 Maio. 2023.
- MOREIRA, J. P.; CARVALHO, L. F. B. S. de; SILVEIRA, T. de S. Serviços para computação em nuvem vantagem x desvantagens de sua implantação. **Revista Interdisciplinar em Gestão, Educação, Tecnologia e Saúde-GETS**, v. 4, n. 1, 2021.
- O'LEARY, R.; RAMSDEN, A. Virtual learning environments. **Learning and Teaching Support Network Generic Centre/ALT Guides, LTSN. Retrieved July**, Citeaser, v. 12, p. 2005, 2002.
- PEREIRA, A. T. C.; SCHMITT, V.; DIAS, M. **AVA - Ambientes virtuais de aprendizagem em diferentes contextos**. 2007.

POLASH, F.; ABUHUSSEIN, A.; SHIVA, S. A survey of cloud computing taxonomies: Rationale and overview. In: IEEE. **The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014)**. [S. l.], 2014. p. 459–465.

RIBEIRO, E. N.; MENDONÇA, G. A. d. A.; MENDONÇA, A. F. A importância dos ambientes virtuais de aprendizagem na busca de novos domínios da ead. In: CONGRESSO INTERNACIONAL DE EDUCAÇÃO A DISTÂNCIA, 2007. **Anais [...]**. Goiás: CEFET, 2007.

RODRIGUES, A. G.; DEMION, B.; MOUAWAD, P. **Master Apache JMeter-From Load Testing to DevOps: Master performance testing with JMeter**. [S. l.]: Packt Publishing Ltd, 2019.

ROSEMANN, D.; RAABE, A. L.; TEIVE, R. C. G. Personalização de conteúdo e avaliação multicritério em ambiente virtual de aprendizagem de código aberto. In: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2014. **Anais [...]**. São Paulo: USP, 2014. v. 3, n. 1, p. 203.

SABAREESHWARAN, M. Optimization of lms for improving user response time. **INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT)**, v. 3, n. 15, 2015.

SANTANA, N. A. d. *et al.* **Uma metodologia para avaliação de desempenho de aplicações em ambiente de computação móvel em nuvem** 2016. 82 f. Dissertação (Programa de Pós-Graduação em Informática Aplicada) – Universidade Federal Rural de Pernambuco, Recife, 2016.

SERVICES, A. W. **what is cloud computing?** Amazon, 2023. Disponível em: <https://aws.amazon.com/pt/what-is-cloud-computing/>. Acesso em: 03 Maio. 2023.

SILVA, A. V. G. da; LIMA, C. J. de; CALLOU, G. R. de A. Análise de desempenho do ambiente virtual de aprendizagem na nuvem privada apache cloudfoundry. **GESTÃO. Org**, Universidade Federal de Pernambuco, v. 17, n. 8, p. 120–133, 2019.

SOROR, A. A.; MINHAS, U. F.; ABOULNAGA, A.; SALEM, K.; KOKOSIELIS, P.; KAMATH, S. Automatic virtual machine configuration for database workloads. **ACM Transactions on Database Systems (TODS)**, ACM New York, NY, USA, v. 35, n. 1, p. 1–47, 2008.

WIECHORK, K.; CHARAO, A. S. Investigating the performance of moodle database queries in cloud environments. In: **ICEIS (1)**. [S. l.: s. n.], 2020. p. 269–275.

DORĐEVIĆ, B. S.; JOVANOVIĆ, S. P.; TIMČENKO, V. V. Cloud computing in amazon and microsoft azure platforms: Performance and service comparison. In: **2014 22nd Telecommunications Forum Telfor (TELFOR)**. [S. l.: s. n.], 2014. p. 931–934.