



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA
MESTRADO EM INFORMÁTICA APLICADA

PAULO ROBERTO DE LIMA

**Tomada de Decisão Multicritério em Nuvens Privadas
para Aplicações, Considerando os Requisitos de
Desempenho e Custo**

DISSERTAÇÃO DE MESTRADO

Recife

2025

PAULO ROBERTO DE LIMA

Tomada de Decisão Multicritério em Nuvens Privadas
para Aplicações, Considerando os Requisitos de
Desempenho e Custo

Dissertação submetida à Coordenação do
Programa de Pós-Graduação em Informática
Aplicada da Universidade Federal Rural de
Pernambuco, como parte dos requisitos neces-
sários para obtenção do grau de Mestre.

Orientadora: Profa. Dra. Erica Teixeira Go-
mes de Sousa

Coorientador: Prof. Dr. Robson Medeiros

Recife

2025

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Ana Catarina Macêdo – CRB-4 1781

L732t Lima, Paulo Roberto de.

Tomada de decisão multicritério em nuvens privadas para aplicações, considerando os requisitos de desempenho e custo / Paulo Roberto de Lima. - Recife, 2025.

80 f.; il.

Orientador(a): Erica Teixeira Gomes de Sousa.

Co-orientador(a): Robson Medeiros.

Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Informática Aplicada, Recife, BR-PE, 2025.

Inclui referências.

1. Computação em nuvem. 2. Tecnologia da informação - Administração. 3. Gerenciamento de recursos de informação. 4. Sistemas de recuperação da informação 5. Desempenho - Avaliação. I. Sousa, Erica Teixeira Gomes de, orient. II. Medeiros, Robson, coorient. III. Título

CDD 004

Dedicatória

A Deus, por me dar força e me guiar em todos os momentos.

À minha Mãe, Maria Luiza da Silva (*in memoriam*), que partiu durante a finalização do meu curso de Licenciatura em Computação na UFRPE.

Ao meu Pai, Samuel Pereira de Lima (*in memoriam*), que partiu durante a realização da pesquisa para esta dissertação.

À minha irmã, Priscila Lima, pelo apoio fundamental durante os momentos vividos com meu Pai.

À minha orientadora, Professora Dra. Erica Teixeira Gomes de Sousa, pela condução segura, pela orientação precisa e pelos valiosos ensinamentos, especialmente durante o estágio de docência. Sua maestria e didática foram uma inspiração.

Ao meu coorientador, Professor Dr. Robson Medeiros, a quem agradeço desde o curso de Licenciatura em Computação, pela clareza com que desmistificou conceitos complexos de programação, e pela fundamental colaboração nesta dissertação.

À minha esposa, Fabiana Borba, por toda a paciência, apoio e compreensão durante os longos períodos de ausência dedicados a esta pesquisa.

Aos meus filhos, razão de todo o meu esforço.

Agradecimentos

Agradecimentos

Agradeço à minha orientadora, Professora Dra. Erica Teixeira Gomes de Sousa, pela orientação precisa e pela condução desta pesquisa. Sou especialmente grato por seu apoio e compreensão durante o período de enfermidade do meu Pai, um momento em que sua flexibilidade e atenção foram fundamentais para que eu pudesse conciliar as responsabilidades pessoais e acadêmicas.

Ao meu coorientador, Professor Dr. Robson Medeiros, estendo minha gratidão pela valiosa colaboração e pelo apoio que transcendeu o âmbito acadêmico. Seu gesto de comparecer pessoalmente em um momento de perda familiar foi de imenso significado e demonstrou uma humanidade que guardarei com grande respeito. A atitude de ambos os orientadores reflete o ambiente de acolhimento do Programa de Pós-Graduação em Informática Aplicada da UFRPE, onde o corpo docente demonstra um genuíno compromisso com o desenvolvimento humano e acadêmico de seus discentes.

À minha esposa, Fabiana Borba, agradeço por todo o amor, paciência e incentivo incondicional. Seu apoio foi o alicerce que me permitiu superar os desafios e dedicar o tempo necessário para a realização desta pesquisa. Aos meus filhos, dedico meu amor e gratidão pelos sorrisos que renovaram minhas energias.

À minha irmã, Priscila Lima, por todo o suporte e pela força compartilhada nos momentos difíceis, especialmente durante o período em que acompanhamos meu Pai. Sua presença foi um esteio fundamental.

Aos meus pais, Maria Luiza da Silva e Samuel Pereira de Lima, que, mesmo não estando aqui para ver a conclusão desta jornada, foram a inspiração e a base para tudo. Este trabalho é também uma homenagem à memória de ambos.

Finalmente, agradeço a Deus, pela força, fé e por me guiar ao longo de todo este percurso.

A maior glória em viver não está em nunca cair,
mas em nos levantarmos cada vez que caímos.

Nelson Mandela

Resumo

A seleção de plataformas de nuvem privada é uma decisão estratégica para organizações que buscam controle e flexibilidade sobre sua infraestrutura de TI. Contudo, a literatura carece de processos estruturados que auxiliem na escolha dessas tecnologias frente a demandas específicas. Neste contexto, este trabalho apresenta uma metodologia de tomada de decisão multicritério para a seleção de nuvens privadas voltadas a aplicações, integrando requisitos de desempenho e custo. A abordagem proposta combina o Planejamento de Experimentos (DOE) para a avaliação técnica e um modelo de Custo Total de Propriedade (TCO) para a análise econômica, utilizando a Análise Hierárquica de Processos (AHP) para a síntese dos resultados. Essa estrutura metodológica é replicável, permitindo a realização de estudos com outras aplicações e plataformas de nuvem privada. O estudo de caso ilustra a aplicabilidade da metodologia através de uma análise entre as plataformas OpenStack e CloudStack, submetendo-as a uma carga de trabalho baseada em microserviços em ambiente controlado. Foram coletadas e analisadas métricas de desempenho, como tempo de resposta, latência, vazão e utilização de CPU e memória. Os resultados evidenciaram que a plataforma OpenStack apresentou um overhead computacional superior, refletido em maiores taxas de utilização do processador e da memória, bem como em uma latência mais sensível à intensidade da carga. Em contrapartida, o CloudStack demonstrou maior eficiência no uso de recursos e maior estabilidade nas métricas de serviço. A análise de TCO apresentou um custo operacional significativamente menor para o CloudStack, impulsionado tanto pela economia no consumo de energia elétrica quanto pela redução de despesas com equipe técnica, decorrente de sua menor complexidade de gerenciamento. Por fim, a aplicação do método AHP atribuiu uma pontuação de preferência de 77,8% ao CloudStack contra 22,2% ao OpenStack, indicando-o como a alternativa mais adequada para o cenário estudado.

Palavras-chave: Computação em Nuvem; Nuvem Privada; OpenStack; CloudStack; Avaliação de Desempenho; Análise Multicritério.

Abstract

The selection of private cloud platforms is a strategic decision for organizations seeking control and flexibility over their IT infrastructure. However, the literature lacks structured processes to aid in choosing these technologies in the face of specific demands. In this context, this work presents a multi-criteria decision-making methodology for selecting application-oriented private clouds, integrating performance and cost requirements. The proposed approach combines Design of Experiments (DOE) for technical evaluation and a Total Cost of Ownership (TCO) model for economic analysis, using Analytic Hierarchy Process (AHP) for synthesizing the results. This methodological framework is replicable, allowing studies to be conducted with other applications and private cloud platforms. The case study illustrates the applicability of the methodology through an analysis between the OpenStack and CloudStack platforms, subjecting them to a microservices-based workload in a controlled environment. Performance metrics such as response time, latency, throughput, and CPU and memory utilization were collected and analyzed. The results showed that the OpenStack platform presented a higher computational overhead, reflected in higher processor and memory utilization rates, as well as latency more sensitive to load intensity. In contrast, CloudStack demonstrated greater efficiency in resource use and greater stability in service metrics. The TCO analysis showed a significantly lower operational cost for CloudStack, driven by both savings in electricity consumption and reduced expenses with technical staff, resulting from its lower management complexity. Finally, the application of the AHP method assigned a preference score of 77.8% to CloudStack versus 22.2% to OpenStack, indicating it as the most suitable alternative for the scenario studied.

Keywords: Cloud Computing; Private Cloud; OpenStack; CloudStack; Performance Evaluation; Multicriteria Analysis.

Lista de ilustrações

Figura 1 – Exemplo de uma Arquitetura de Microserviços	21
Figura 2 – Arquitetura OpenStack	24
Figura 3 – Arquitetura CloudStack	25
Figura 4 – Metodologia	46
Figura 5 – Topologia	56
Figura 6 – Utilização Memória	63
Figura 7 – Utilização Processador	64
Figura 8 – Tempo de Resposta	65
Figura 9 – Latência	65
Figura 10 – Throughput	66
Figura 11 – Memória	67
Figura 12 – Processador	68
Figura 13 – Tempo de Resposta	69
Figura 14 – Latência	70
Figura 15 – Throughput	70
Figura 16 – Hierarquia do Problema	75

Lista de tabelas

Tabela 1 – Comparativo entre Arquitetura Monolítica e Microserviços	20
Tabela 2 – Comparativo entre modelos de Implantação de Nuvem	23
Tabela 3 – Comparativo Arquitetural: OpenStack vs. CloudStack	23
Tabela 4 – Decomposição do Custo Total de Propriedade (TCO) em Nuvem Privada	32
Tabela 5 – Comparativa de Trabalhos Relacionados	40
Tabela 6 – Características do servidor OpenStack/Cloudstack	57
Tabela 7 – Oferta de Serviço da Nuvem Privada	57
Tabela 8 – Fatores e níveis do planejamento de experimentos	59
Tabela 9 – Cenários gerados pelo planejamento de experimentos	59
Tabela 10 – Resultado das Métricas de Desempenho - OpenStack	61
Tabela 11 – Resultado das Métricas de Desempenho - CloudStack	62
Tabela 12 – Parâmetros para o Modelo de Custo Total de Propriedade (TCO) . . .	72
Tabela 13 – Cálculo do Custo Total de Propriedade (TCO) Mensal	73
Tabela 14 – Escala Fundamental de Saaty para Comparações Paritárias	76
Tabela 15 – Matriz de Comparação Paritária para os Critérios	76
Tabela 16 – Vetor de Prioridades com Pesos dos Critérios	77
Tabela 17 – Índice de Aleatoriedade (RI)	78
Tabela 18 – Pontuação Normalizada das Plataformas por Critério	79
Tabela 19 – Síntese Final para a Escolha da Plataforma	80

Lista de abreviaturas e siglas

AHP	Análise Hierárquica de Processos (do inglês Analytic Hierarchy Process)
API	Interface de Programação de Aplicação (do inglês Application Programming Interface)
ASGI	Interface de Gateway de Servidor Assíncrono (do inglês Asynchronous Server Gateway Interface)
AWS	Amazon Web Services
BWM	Método do Melhor-Pior (do inglês Best-Worst Method)
CAPEX	Despesas de Capital (do inglês Capital Expenditure)
CI	Índice de Consistência (do inglês Consistency Index)
CMP	lataforma de Gerenciamento de Nuvem (do inglês Cloud Management Platform)
COPRAS	Avaliação Proporcional Complexa (do inglês Complex Proportional Assessment)
CPU	Unidade Central de Processamento (do inglês Central Processing Unit)
CR	Razão de Consistência (do inglês Consistency Ratio)
CTMC	Cadeias de Markov de Tempo Contínuo (do inglês Continuous Time Markov Chains)
DOE	Planejamento de Experimentos (do inglês Design of Experiments)
GCP	Google Cloud Platform
HTTP	Protocolo de Transferência de Hipertexto (do inglês Hypertext Transfer Protocol)

ISM	Modelagem Estrutural Interpretativa (do inglês Interpretive Structural Modeling)
IaaS	Infraestrutura como Serviço (do inglês Infrastructure as a Service)
LAT	Latência
MCC	Computação Móvel em Nuvem (do inglês Mobile Cloud Computing)
MCDM	Tomada de Decisão Multicritério (do inglês Multi-Criteria Decision Making)
MLaaS	Machine Learning como Serviço (do inglês Machine Learning as a Service)
MOORA	Otimização Multi-objetivo com base na Análise de Razão (do inglês Multi-Objective Optimization on the basis of Ratio Analysis)
NFS	Servidor de Arquivos de Rede (do inglês Network File System)
OFAT	Um Fator por Vez (do inglês One-Factor-At-A-Time)
OPEX	Despesas Operacionais (do inglês Operational Expenditure)
ORM	Mapeamento Objeto-Relacional (do inglês Object-Relational Mapping)
OVS	Open vSwitch
PROMETHEE	Método de Organização de Ranqueamento de Preferência para Avaliações de Enriquecimento (do inglês Preference Ranking Organization Method for Enrichment Evaluations)
PUE	Eficácia do Uso de Energia (do inglês Power Usage Effectiveness)
QoS	Qualidade de Serviço (do inglês Quality of Service)
RI	Índice de Aleatoriedade (do inglês Random Index)
SPN	Redes de Petri Estocásticas (do inglês Stochastic Petri Nets)

TCO	Custo Total de Propriedade (do inglês Total Cost of Ownership)
THR	Throughput (Vazão)
TMR	Tempo de Resposta
TOPSIS	Técnica para Ordem de Preferência por Similaridade com a Solução Ideal (do inglês Technique for Order of Preference by Similarity to Ideal Solution)
VM	Máquina Virtual (do inglês Virtual Machine)
kWh	Quilowatt-hora

Sumário

1	Introdução	15
1.1	Motivação e Justificativa	16
1.2	Objetivos	17
1.3	Organização do Trabalho	17
2	Fundamentação Teórica	19
2.1	Microserviços	19
2.2	Computação em Nuvem	21
2.3	Plataforma de Nuvem OpenStack	24
2.4	Plataforma de Nuvem CloudStack	25
2.5	Avaliação de Desempenho	26
2.5.1	Medição	26
2.5.2	Modelagem Analítica	27
2.5.3	Simulação	28
2.6	Planejamento de Experimentos (DOE)	29
2.7	Processo de Análise Hierárquica (AHP)	30
2.8	Custo em Nuvem Privada	31
2.8.1	Custo de Pessoal e a Influência da Complexidade	32
2.8.2	Custo de Energia e a Relação com o Desempenho	33
3	Trabalhos Relacionados	34
3.1	Introdução	34
3.2	Modelos para Tomada de Decisão considerando Desempenho	34
3.3	Modelos para Tomada de Decisão considerando Avaliação de Custo	36
3.4	Modelos para Tomada de Decisão para Ambientes de Nuvem	37
3.5	Comparação com os trabalhos relacionados	39
4	Metodologia para Planejamento e Tomada de Decisão	43

4.1	Analisar o Objetivo do Estudo e Selecionar as Métricas	47
4.2	Configurar os Ambientes de Nuvem com as Plataformas OpenS- tack e CloudStack	47
4.3	Implantar a Aplicação para a Execução dos Testes de Carga .	48
4.4	Planejar os Experimentos por Meio da Metodologia DOE . .	48
4.5	Medir o Desempenho das Plataformas por Meio da Execução dos Testes	49
4.6	Desenvolver o Modelo de Custo	49
4.6.1	Custo de Hardware (Capex)	50
4.6.2	Custo da Equipe Técnica (OPEX)	51
4.6.3	Custo de Energia Elétrica (OPEX)	51
4.6.4	Custo Total de Propriedade (TCO)	53
4.7	Analisar Estatisticamente os Dados de Desempenho Coletados	53
4.8	Analisar os Cenários de Desempenho e Custo para Cada Pla- taforma	53
4.9	Realizar a Tomada de Decisão com o Método AHP	54
4.10	Considerações Finais	55
5	Estudo de Caso	56
5.1	Introdução	56
5.2	Configuração dos Ambientes de Nuvem com as Plataformas OpenStack e CloudStack	56
5.3	Implementação do Microserviço para a Execução dos Testes de Carga	57
5.4	Planejamento dos Experimentos por Meio da Metodologia DOE	58
5.5	Análise de Desempenho	60
5.5.1	Diferença percentual	62
5.5.2	Análise dos Efeitos dos Fatores sobre as Métricas de Desempenho . . .	66
5.6	Análise de Custo	71

5.7	Análise dos Cenários de Desempenho e Custo para Cada Plataforma	74
5.8	Realização da Tomada de Decisão com o Método AHP	74
5.8.1	Estruturação do problema de decisão	75
5.8.1.1	Hierarquia do Problema e Ponderação dos Critérios	75
5.8.1.2	Avaliação das Alternativas e Síntese Final	78
5.9	Considerações Finais	80
6	Conclusões e Trabalhos Futuros	83
6.1	Limitações	84
6.2	Trabalhos Futuros	85
	Referências	86

1 Introdução

A computação em nuvem representa uma mudança fundamental no modelo de provisionamento e consumo de recursos computacionais. A definição desta abordagem, já consolidada na literatura, engloba características essenciais como o autoatendimento sob demanda, o amplo acesso via rede, o *pool* de recursos, a rápida elasticidade e a mensuração de serviço, permitindo que as organizações acessem infraestrutura e plataformas de forma flexível e escalável (YOUSEFNEZHAD; FARD; HEMMATI, 2020). Dentro deste paradigma, as organizações podem optar por diferentes modelos de implantação. A adoção do modelo de nuvem privada, em particular, é frequentemente justificada pela necessidade de maior governança sobre a infraestrutura e os dados, visando atender a requisitos de segurança e conformidade regulatória (HASAN, 2022).

No ecossistema de soluções de código aberto (*open-source*) para nuvens privadas, as plataformas OpenStack e CloudStack representam duas das principais alternativas. Contudo, elas se baseiam em filosofias arquitetônicas distintas: o OpenStack é um conjunto de serviços independentes que se comunicam via APIs, oferecendo alta flexibilidade, enquanto o CloudStack adota uma abordagem mais integrada, priorizando a simplicidade operacional (SINGH; KAUR, 2021). Essas diferenças de *design* resultam em perfis de desempenho e consumo de recursos distintos, tornando a seleção entre as duas plataformas um desafio técnico e estratégico que carece de análises comparativas diretas e controladas.

Adicionalmente, o desempenho de uma infraestrutura de nuvem está intrinsecamente ligado à carga de trabalho que ela suporta. Aplicações modernas são cada vez mais desenvolvidas com base no estilo arquitetural de microsserviços, que estrutura uma aplicação como uma coleção de pequenos serviços autônomos e especializados que se comunicam por meio de APIs (FRANCESCO; MALAVOLTA; LAGO, 2021). Este modelo de desenvolvimento torna o desempenho da rede e a eficiência do processamento de requisições fatores críticos para desempenho da aplicação. Portanto, a avaliação de desempenho de plataformas de nuvem privada utilizando cargas de trabalho baseadas em microsserviços constitui um cenário de teste relevante e contemporâneo.

De forma complementar à análise técnica, a dimensão econômica desempenha um

papel decisivo na sustentabilidade da infraestrutura de TI. Para endereçar essa questão, este trabalho desenvolve e aplica um modelo de Custo Total de Propriedade (TCO) específico para nuvens privadas. A metodologia proposta converte os indicadores técnicos como a eficiência no consumo de recursos computacionais e a complexidade operacional de gerenciamento em valores monetários tangíveis. Esse modelo integra o investimento de capital em hardware com as despesas operacionais recorrentes, nomeadamente o consumo de energia elétrica e os custos associados à equipe técnica, permitindo uma comparação financeira direta entre as plataformas.

1.1 Motivação e Justificativa

A crescente dependência da computação em nuvem requer que as organizações realizem uma análise detalhada na seleção de suas plataformas, uma decisão com implicações diretas em custos, segurança e desempenho (RAMOS; JUSTO, 2023b). A escolha envolve mais do que uma avaliação técnica, constituindo um problema de decisão multicritério que requer a ponderação de fatores financeiros e operacionais.

Nesse escopo, o custo é um fator fundamental na análise. Para fornecer uma avaliação aplicada, este trabalho desenvolve e aplica um modelo de Custo Total de Propriedade (TCO) para nuvem privada, que traduz a eficiência técnica em impacto financeiro. O modelo utiliza os dados de utilização de recursos (*CPU* e memória), aferidos nos experimentos, como base para calcular o custo de energia elétrica, e incorpora a complexidade de gerenciamento das plataformas para estimar os custos operacionais com a equipe técnica. Este modelo abrange não apenas os custos operacionais de energia, derivados diretamente da eficiência de recursos da plataforma, mas também os custos de pessoal, associados à complexidade de gerenciamento, e a amortização do hardware. A análise desses custos em conjunto com às métricas de desempenho é um requisito para a justificativa do investimento e para a seleção da solução de plataforma de nuvem privada.

No ecossistema de nuvens privadas *open-source*, OpenStack e CloudStack são duas das principais alternativas. Contudo, a seleção entre estas duas tecnologias apresenta um desafio, pois a literatura e os estudos de caso existentes frequentemente não apresentam

uma análise de desempenho e custo comparativa, sob condições idênticas e controladas, que possa quantificar objetivamente os *trade-offs* entre elas. Esta lacuna resulta na ausência de uma base de evidências para uma decisão otimizada por parte de gestores e arquitetos de TI.

Ademais, mesmo quando uma comparação empírica é realizada, como a proposta neste trabalho, ela frequentemente revela o “paradoxo dos *trade-offs*”: uma plataforma pode apresentar maior eficiência no consumo de *CPU* (menor custo de energia), enquanto a outra exibe menor complexidade (menor custo de pessoal). A esta complexidade técnica, soma-se a dimensão financeira, o que levanta a questão de como ponderar objetivamente estes resultados distintos para se chegar a uma conclusão final.

Diante desse cenário, a questão central que esta pesquisa busca responder é: **Como selecionar nuvens privadas para aplicações considerando os requisitos de desempenho e custo?**

1.2 Objetivos

Este trabalho tem como objetivo principal apresentar uma metodologia para tomada de decisão na seleção de Ambientes de Nuvem Privada para aplicações considerando o Processo de Análise Hierárquica e o Planejamento de Experimentos baseada em requisitos de desempenho e custo.

Os objetivos específicos são descritos a seguir:

- Apresentar um estudo de caso para seleção entre as plataformas de nuvem Openstack e Cloudstack baseado nos requisitos de desempenho e custo.
- Propor modelo de custo baseado em TCO para avaliação de nuvens privadas.
- Desenvolver microserviço para auxiliar na tomada de decisão de nuvens privadas considerando requisitos de desempenho e custo.

1.3 Organização do Trabalho

Este documento está estruturado em seis capítulos.

- **Capítulo 1** introduz a pesquisa, apresentando o contexto, a motivação e justificativa, o problema de pesquisa e os objetivos do trabalho.
- **Capítulo 2** estabelece a Fundamentação Teórica, abordando os conceitos de Microserviços e Computação em Nuvem, com ênfase nas arquiteturas das plataformas OpenStack e CloudStack . Também define as métricas de Desempenho e detalha a teoria do modelo de Custo Total de Propriedade (TCO) aplicado a nuvens privadas.
- **Capítulo 3** dedica-se à análise dos trabalhos relacionados, revisando estudos anteriores nas áreas de avaliação de desempenho, custo e modelos para tomada de decisão, e finalizando com uma comparação que situa a presente pesquisa na literatura.
- **Capítulo 4** apresenta a Metodologia para Planejamento e Tomada de Decisão. Apresenta a abordagem combinada que integra o Planejamento de Experimentos (DOE) para a coleta de dados e o método AHP para a seleção. Descreve a definição das métricas, o desenvolvimento das equações do modelo de custo TCO e os procedimentos de análise estatística.
- **Capítulo 5** descreve os estudos de caso realizados, desde a concepção do ambiente até a aplicação e avaliação dos modelos de desempenho, custo e decisão.
- **Capítulo 6** apresenta as conclusões, resumizando as contribuições da dissertação, discutindo suas limitações e apontando direções para Trabalhos Futuros.

2 Fundamentação Teórica

Este capítulo estabelece as bases conceituais que sustentam a presente dissertação. Inicialmente, são abordados os conceitos de Microserviços, seguidos pela definição de Computação em Nuvem e o detalhamento de seus modelos de implantação. Na sequência, o capítulo caracteriza as plataformas de nuvem privada OpenStack e CloudStack, objetos centrais deste estudo. Posteriormente, são apresentados os fundamentos de Desempenho e as Técnicas de Avaliação, introduzindo o Planejamento de Experimentos (DOE) como método estatístico para a estruturação dos testes. Por fim, o capítulo aborda a modelagem de Custo e detalha o Processo de Análise Hierárquica (AHP), ferramenta utilizada para a tomada de decisão multicritério.

2.1 Microserviços

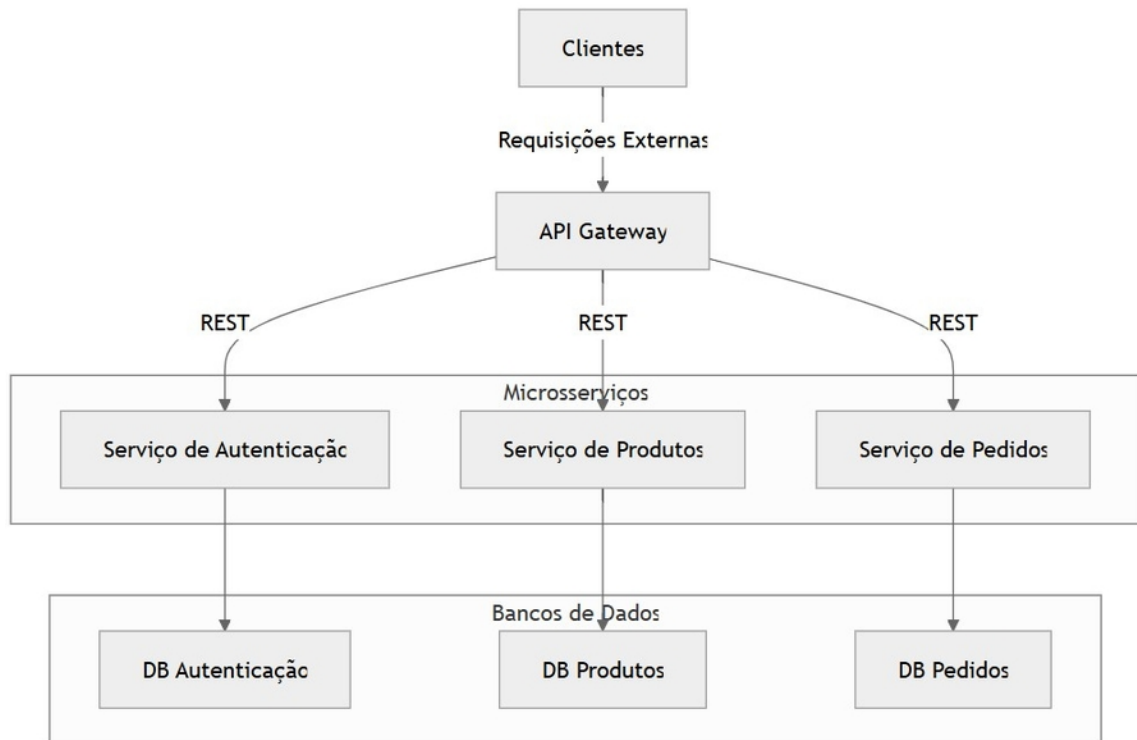
A arquitetura de microserviços representa uma abordagem para o desenvolvimento de sistemas de software na qual uma aplicação complexa é composta por um conjunto de serviços, autônomos e fracamente acoplados. Diferentemente da arquitetura monolítica, que agrupa toda a funcionalidade da aplicação em um único processo, os microserviços decompõem o sistema em serviços independentes, cada um responsável por uma capacidade de negócio específica (ZIMMERMANN, 2021). Essa especialização permite que cada serviço seja desenvolvido, implantado, escalado e mantido de forma independente, promovendo agilidade e flexibilidade no ciclo de vida do software. A Tabela 1 mostra as principais características das arquiteturas monolíticas e de microserviços, o comparativo destaca características estruturais e operacionais fundamentais, como formas de implantação, estratégias de escalabilidade e modelos de gestão de dados, contrapondo a rigidez da abordagem unificada à flexibilidade dos serviços distribuídos.

Tabela 1 – Comparativo entre Arquitetura Monolítica e Microserviços

Característica	Arquitetura Monolítica	Arquitetura de Microserviços
Estrutura	Aplicação de processo único com todos os componentes acoplados.	Conjunto de serviços pequenos e independentes.
Implantação	A aplicação inteira é implantada como uma única unidade.	Cada serviço pode ser implantado de forma independente.
Escalabilidade	Escalabilidade vertical ou horizontal de toda a aplicação.	Escalabilidade horizontal granular de serviços individuais.
Tecnologia	Pilha tecnológica única e homogênea.	Poliglotismo: tecnologias diferentes podem ser usadas para cada serviço.
Gestão de Dados	Banco de dados centralizado e único para toda a aplicação.	Cada serviço gerencia seu próprio banco de dados (descentralizado).
Comunicação	Chamadas de função internas ao processo.	Comunicação via rede, tipicamente por meio de APIs HTTP/REST ou mensageria.

Fonte: Adaptado de Dragulescu e Goth (2022)

A comunicação entre os serviços é um aspecto central e um dos principais desafios de desempenho desta arquitetura. Ela pode ocorrer de forma síncrona, onde um serviço faz uma requisição a outro e aguarda a resposta (geralmente via APIs REST), ou de forma assíncrona, utilizando sistemas de mensageria (como RabbitMQ ou Kafka) onde os serviços se comunicam por meio de eventos, sem acoplamento direto (DRAGOLESCU; GOTH, 2022). A Figura 1 ilustra uma arquitetura de microserviços onde um *API Gateway* gerencia a entrada de requisições e as distribui para serviços de *backend* independentes via REST. O diagrama enfatiza a descentralização de dados, mostrando que cada microserviço opera com seu próprio banco de dados exclusivo, garantindo o desacoplamento e a autonomia funcional entre os componentes do sistema .

**Figura 1 – Exemplo de uma Arquitetura de Microserviços**

Fonte: O Autor

A computação em nuvem oferece a infraestrutura e os serviços necessários para a implementação e operação de arquiteturas baseadas em microserviços. As plataformas de nuvem, especialmente as de nuvem privada como OpenStack e CloudStack, oferecem a infraestrutura elástica, a automação e o isolamento de recursos necessários para suportar a natureza distribuída e dinâmica dos microserviços. Tecnologias de containerização, como Docker, e orquestradores, como Kubernetes, são frequentemente utilizados em conjunto para empacotar e gerenciar o ciclo de vida dos microserviços (SINGH; KAUR, 2021). Contudo, essa natureza distribuída impõe uma sobrecarga (*overhead*) de comunicação na rede, tornando o desempenho da infraestrutura de nuvem subjacente um fator crítico para desempenho geral da aplicação.

2.2 Computação em Nuvem

A computação em nuvem permite acesso ubíquo e sob demanda via rede a um conjunto compartilhado de recursos computacionais configuráveis (YOUSEFNEZHAD;

FARD; HEMMATI, 2020). Este paradigma flexível pode ser implementado através de diferentes modelos de implantação, cada um atendendo a requisitos específicos de controle, custo e acessibilidade.

A Nuvem Pública é caracterizada pelo provisionamento da infraestrutura para uso aberto pelo público em geral. Ela é de propriedade, gerenciada e operada por uma organização provedora de serviços (como AWS, Google ou Microsoft), situando-se fisicamente nas instalações desse provedor. Este modelo oferece alta escalabilidade e eficiência de custos, pois os recursos são compartilhados entre múltiplos clientes sob um modelo de pagamento por uso (MELL; GRANCE, 2011).

Em contrapartida, a Nuvem Privada é provisionada para uso exclusivo de uma única organização, que pode compreender múltiplos consumidores (por exemplo, unidades de negócios). Diferente do modelo público, a nuvem privada oferece à organização um controle granular sobre os recursos, políticas de segurança e governança de dados. Ela pode ser gerenciada pela própria organização ou por terceiros, e pode estar localizada interno ou externo as instalações da empresa (MELL; GRANCE, 2011; HASAN et al., 2022).

A Nuvem Híbrida representa uma composição de duas ou mais infraestruturas de nuvem distintas (privada ou pública) que permanecem entidades únicas, mas são unidas por tecnologias padronizadas ou proprietárias. Essa arquitetura permite a portabilidade de dados e aplicações entre os ambientes, oferecendo às organizações a flexibilidade de manter dados sensíveis em sua infraestrutura privada enquanto utilizam a capacidade computacional da nuvem pública para cargas de trabalho menos críticas ou para lidar com picos de demanda (*cloud bursting*) (MELL; GRANCE, 2011).

A Tabela 2 apresenta um comparativo direto entre os três principais modelos de implantação, destacando suas diferenças em termos de propriedade, modelo de custo e controle.

Tabela 2 – Comparativo entre modelos de Implantação de Nuvem

Atributo	Nuvem Pública	Nuvem Privada	Nuvem Híbrida
Propriedade	Provedor de serviços (ex: AWS, GCP, Azure).	A própria organização.	Mista (organização e provedor).
Modelo de Custo	OPEX (pagamento por uso).	CAPEX (investimento inicial) + OPEX.	Misto.
Controle	Limitado (controlado pelo provedor).	Total (controlado pela organização).	Alto na parte privada, limitado na pública.
Segurança	Responsabilidade compartilhada.	Responsabilidade total da organização.	Complexa, depende da integração.
Escalabilidade	Alta e elástica.	Limitada ao hardware adquirido.	Alta (através do <i>cloud bursting</i>).

Fonte: Adaptado de Hasan et al. (2022)

Tabela 3 – Comparativo Arquitetural: OpenStack vs. CloudStack

Característica	OpenStack	CloudStack
Arquitetura	Modular e distribuída (conjunto de serviços).	Monolítica e integrada.
Complexidade	Alta; requer integração de múltiplos projetos.	Menor; instalação e gestão simplificadas.
Flexibilidade	Alta; permite customização profunda.	Menor; focada em casos de uso IaaS padrão.
Rede	Complexa e flexível (Neutron com <i>plugins</i>).	Integrada e mais simples.

Fonte: Adaptado de Singh e Kaur (2021)

2.3 Plataforma de Nuvem OpenStack

OpenStack é um conjunto de componentes de software de código aberto que fornece uma plataforma para gerenciamento de nuvem, primariamente como Infraestrutura como Serviço (IaaS). Sua característica arquitetural fundamental é a modularidade. Diferente de soluções integradas, OpenStack é composto por um conjunto de projetos ou serviços independentes que se comunicam através de APIs, como o Nova (para provisionamento de instâncias de computação), Keystone (para autenticação) e Neutron (para redes).

A Figura 2 apresenta a arquitetura de serviços do OpenStack, expõe a vasta gama de componentes independentes necessários para uma implantação funcional.

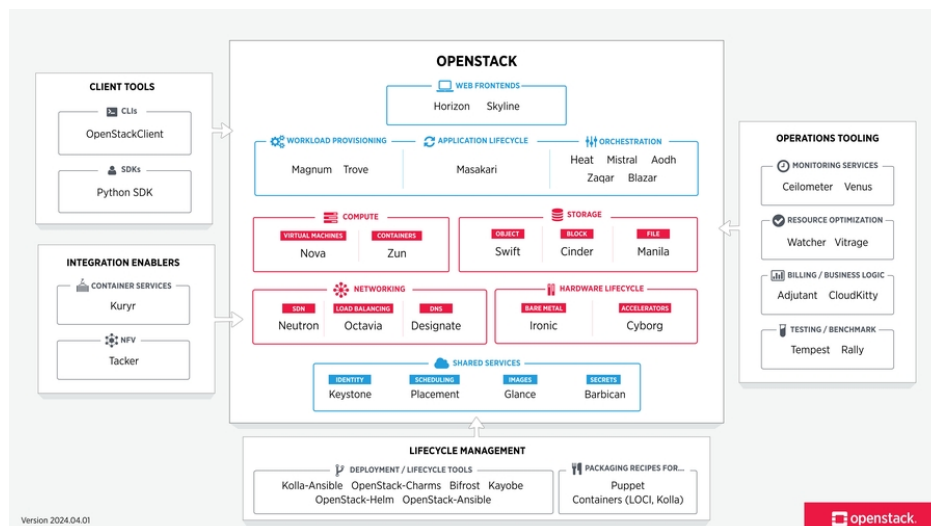


Figura 2 – Arquitetura OpenStack
Fonte: VEXXHOST

Embora essa arquitetura modular ofereça alta flexibilidade e escalabilidade, ela introduz uma complexidade operacional significativa. A necessidade de integrar e monitorar múltiplos serviços (Nova, Keystone, Neutron, Cinder) e o barramento de mensagens que os conecta, gera *overhead* (GROVER; VERMA; RAJAGOPALAN, 2023). Esta complexidade é um fator central na análise comparativa de TCO e desempenho desta dissertação, pois, como aponta a literatura, a abordagem distribuída do OpenStack consome ciclos de *CPU* e memória para a comunicação constante entre seus múltiplos serviços independentes (PATEL; PATEL, 2016). Especificamente o componente de rede, Neutron, é frequentemente citado por sua complexidade, o que pode impactar diretamente o desempenho da

comunicação, impondo uma carga de processamento de pacotes significativa sobre a *CPU* do *host* e, conseqüentemente, aumentando a latência das aplicações (CHAUDHRY, 2015).

2.4 Plataforma de Nuvem CloudStack

Apache CloudStack é uma plataforma de nuvem de software de código aberto projetada para implantar e gerenciar grandes redes de máquinas virtuais, também como uma solução IaaS. Em contraste direto com a abordagem modular do OpenStack, CloudStack adota uma arquitetura monolítica e integrada (SINGH; KAUR, 2021). Todos os seus serviços centrais, como gerenciamento de computação, rede e armazenamento, são pré-integrados em um único pacote de software, priorizando a simplicidade operacional e a rapidez na implantação.

A Figura 3 ilustra a arquitetura do Apache CloudStack. A imagem reforça o conceito da arquitetura monolítica e integrada descrita.

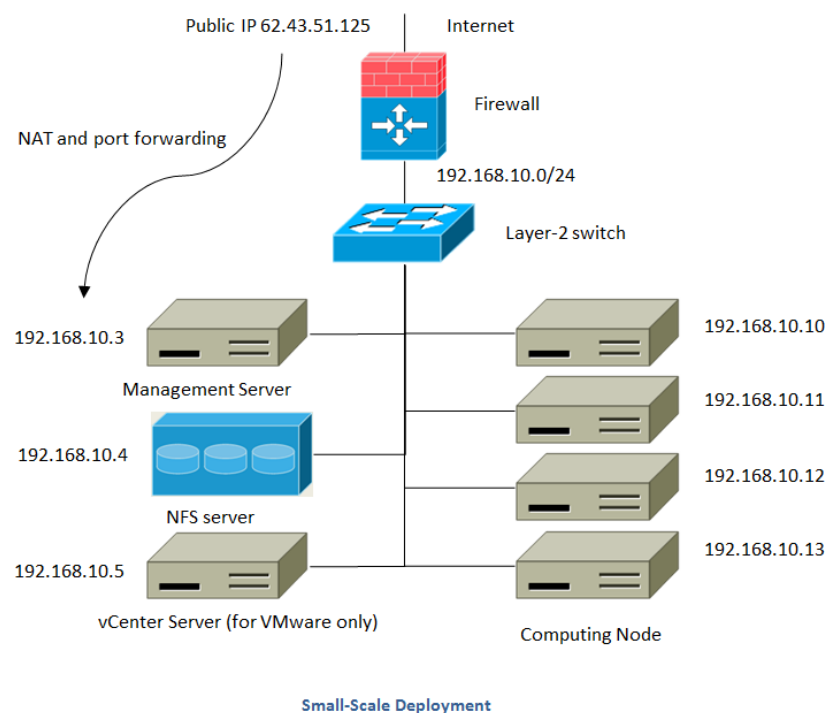


Figura 3 – Arquitetura CloudStack
Fonte: Apache Software Foundation

No centro do diagrama, encontra-se o Servidor de Gerenciamento (*Management Server*), que atua como o núcleo monolítico da plataforma. Este único componente é

responsável por orquestrar e gerenciar todos os recursos da infraestrutura — incluindo Computação (*Hosts*), Armazenamento (Primário e Secundário) e Rede — através de APIs internas.

Seu *design “turnkey”* (pronta para uso), como apontado por S. Filho e G. Filho (2016), resulta em uma arquitetura com menos componentes móveis e menor complexidade de gerenciamento. Uma das análises feita por esta pesquisa foi identificar se essa simplicidade arquitetural se traduz em um menor *overhead* de recursos (*CPU* e memória) e em um desempenho de rede mais estável (menor latência) quando comparada diretamente ao OpenStack sob uma carga de trabalho idêntica.

2.5 Avaliação de Desempenho

A avaliação de desempenho em sistemas de nuvem é o processo de medição e análise de um conjunto de indicadores quantitativos para caracterizar o comportamento do sistema sob uma carga de trabalho específica. Trata-se de um conceito multidimensional, frequentemente agrupado sob o termo Qualidade de Serviço (QoS), que não pode ser representado por uma única métrica (ALI, 2021).

O processo de avaliação de desempenho de sistemas computacionais pode ser realizada por meio de três técnicas principais: medição, modelagem e simulação. A escolha da técnica mais adequada depende fundamentalmente da fase do ciclo de vida em que o sistema se encontra (JAIN, 1991; MACIEL, 2022).

2.5.1 Medição

A medição é a técnica de avaliação de desempenho baseada na observação direta e coleta de dados em um sistema computacional real ou em um protótipo.

A medição captura o comportamento real do sistema sob uma carga de trabalho específica. No entanto, ela só é aplicável se um sistema ou protótipo já existir (JAIN, 1991). As medições requerem a definição cuidadosa de métricas, a escolha de ferramentas (como *benchmarks* e monitores de sistema) e um planejamento experimental rigoroso, como o Planejamento de Experimentos (DOE), para garantir que os resultados sejam

estatisticamente válidos e replicáveis. (LILJA, 2000) reforça a importância da medição, destacando o uso de programas de *benchmark* como estratégias centrais para essa técnica. Ele enfatiza que a medição é essencial para compreender o desempenho real e que a análise estatística dos dados medidos é crucial para interpretar os resultados corretamente. A medição prática envolve a utilização de ferramentas de software (monitores que coletam dados sobre a utilização de recursos) para capturar o comportamento do sistema sob cargas de trabalho específicas. Maciel (2020) corrobora que a medição é a técnica apropriada quando o sistema já existe. O autor enfatiza que a medição de desempenho não se resume à coleta de dados, mas exige um planejamento experimental como o DOE, utilizado nesta tese e uma caracterização de carga de trabalho (*workload*) precisa. A medição é fundamental para obter dados que sirvam de base para a validação de modelos ou para a comparação direta de alternativas.

2.5.2 Modelagem Analítica

A avaliação de desempenho por modelagem analítica consiste na utilização de representações matemáticas e lógicas para descrever o comportamento de um sistema e prever seus índices de performance. Essa abordagem analítica é considerada rápida e custo-efetiva, sendo ideal para as fases iniciais do projeto, quando o sistema real ainda não foi construído (JAIN, R., 1991). Ela permite analisar uma ampla gama de parâmetros e configurações. Contudo, sua precisão depende diretamente das suposições e simplificações feitas para tornar o modelo matematicamente tratável, o que pode limitar a representação fiel de sistemas muito complexos.

Lilja (2000) define a modelagem analítica, com ênfase na análise de filas, como uma das técnicas fundamentais para a avaliação de desempenho. O autor posiciona o método como uma alternativa eficaz para obter *insights* sobre o comportamento do sistema, como a identificação de gargalos e taxas de utilização, exigindo menor esforço computacional do que a simulação, embora demande um conhecimento aprofundado de ferramentas estatísticas e matemáticas (LILJA, D. J., 2000). Complementarmente, o uso de formalismos como Redes de Petri Estocásticas (SPN) permite a construção de modelos capazes de capturar a

concorrência e o sincronismo em sistemas complexos, possibilitando a derivação de métricas de desempenho (como vazão e tempo de resposta) (MACIEL, P. R. M., 2022).

No âmbito da modelagem analítica, a Teoria das Filas é uma técnica central para analisar sistemas com compartilhamento de recursos. Jain (1991) define a análise de filas como a ferramenta primária para prever o desempenho de sistemas de computação, permitindo estimar tempos de espera e utilização de servidores sob diferentes cargas de trabalho (JAIN, R., 1991). Lilja (2000) corrobora essa visão, utilizando modelos de filas e leis operacionais, como a Lei de Little, para estabelecer relações matemáticas precisas entre vazão e latência, facilitando a identificação de gargalos sem a necessidade de protótipos físicos (LILJA, D. J., 2000). Maciel (2022) integra essa abordagem ao destacar que as redes de filas, sejam abertas ou fechadas, são fundamentais para modelar a contenção de recursos e o fluxo de tarefas em arquiteturas distribuídas, fornecendo métricas médias de desempenho em estado estacionário (MACIEL, P. R. M., 2022).

Adicionalmente, as Cadeias de Markov são empregadas para representar sistemas que evoluem probabilisticamente através de estados discretos. Jain descreve os processos de Markov como essenciais para modelar comportamentos onde o estado futuro do sistema depende apenas do estado atual, sendo a base para análises de disponibilidade e confiabilidade (JAIN, R., 1991). Embora o foco de Lilja seja predominantemente na medição de desempenho, a lógica de estados discretos fundamenta a compreensão de sistemas que alternam entre funcionamento e falha (LILJA, D. J., 2000). Maciel (2022) aprofunda o uso desta técnica, defendendo as Cadeias de Markov de Tempo Contínuo (CTMC) como o motor matemático subjacente a formalismos de alto nível, essencial para gerar as equações de equilíbrio que permitem calcular a probabilidade de o sistema estar disponível ou indisponível ao longo do tempo (MACIEL, P. R. M., 2022).

2.5.3 Simulação

É uma técnica que imita o comportamento de um sistema ao longo do tempo por meio de um modelo computacional, permitindo a coleta de estatísticas sobre seu desempenho.

A simulação é uma técnica mais detalhada e flexível que a modelagem analítica, capaz de refletir comportamentos complexos do sistema que são difíceis de modelar matematicamente (JAIN, 1991). Simulações podem ser de eventos discretos, utilizando dados de entrada reais ou por aleatoriedade e repetição para obter resultados estatísticos. Embora geralmente mais precisa que a modelagem, a simulação é computacionalmente intensiva e seu desenvolvimento pode ser demorado. A simulação também pode ser definida como um método prático quando a medição é impossível (o sistema não existe) ou quando a modelagem analítica é muito complexa (LILJA, 2000). Ele destaca a importância da geração de números aleatórios e da validação estatística dos resultados (análise de transientes, número de replicações) para garantir que as conclusões da simulação sejam confiáveis e representativas do sistema real.

2.6 Planejamento de Experimentos (DOE)

O Planejamento de Experimentos (DOE, do inglês *Design of Experiments*) é definido como uma metodologia sistemática e rigorosa utilizada para investigar a relação de causa e efeito entre variáveis de entrada (fatores) e as variáveis de saída (respostas) de um processo ou sistema.

Segundo Mathews (2005), ao observar a resposta do sistema sob uma matriz planejada de configurações de fatores, é possível determinar um modelo matemático estatisticamente válido que descreve o comportamento do objeto de estudo. Essa abordagem não se limita apenas a coletar dados, mas foca em estruturar os testes de forma a maximizar a quantidade de informações obtidas com o menor consumo possível de recursos experimentais.

Uma das principais distinções do DOE em relação às abordagens experimentais tradicionais é a sua capacidade de avaliar múltiplos fatores simultaneamente. Enquanto o método de “um fator por vez” (*OFAT - One-Factor-At-A-Time*) altera uma única variável mantendo as outras constantes, o que frequentemente falha em detectar como os fatores influenciam uns aos outros, o DOE é projetado especificamente para identificar e quantificar as interações entre variáveis (SAS INSTITUTE INC., 2014).

Siebertz (2016) reforça que a aplicação de arranjos fatoriais permite testar combinações de níveis de fatores, garantindo que o espaço experimental seja explorado de maneira eficiente e que as conclusões sobre o desempenho do sistema sejam robustas e livres de vieses induzidos por variáveis de ruído não controladas.

Os objetivos da aplicação do DOE transcendem a observação, permitindo a otimização e a modelagem preditiva do sistema. O método é utilizado para selecionar os fatores, separando as variáveis críticas daquelas não críticas, e para ajustar os níveis desses fatores de modo a atingir um desempenho ótimo (MATHEWS, 2005). No contexto de sistemas complexos, essa ferramenta estatística fornece a base para a tomada de decisões orientada por dados, permitindo que melhorias de qualidade e eficiência sejam implementadas com base em evidências quantitativas concretas, em vez de tentativa e erro (SAS INSTITUTE INC., 2014).

2.7 Processo de Análise Hierárquica (AHP)

O Processo de Análise Hierárquica (AHP), desenvolvido por Thomas L. Saaty na década de 1970, é um dos métodos mais utilizados no campo da Tomada de Decisão Multicritério (MCDM). O método assume que uma tomada de decisão complexa pode ser facilitada pela decomposição do problema em uma estrutura hierárquica linear, composta por um objetivo global no topo, critérios e subcritérios nos níveis intermediários e as alternativas de decisão na base (SAATY, 1990). Essa estruturação permite fragmentar uma questão multifacetada em componentes menores e mais manejáveis, facilitando a análise lógica e sistemática das variáveis envolvidas.

O núcleo operacional do AHP é realização de comparações paritárias. Em vez de tentar ordenar todas as alternativas simultaneamente, o decisor avalia os elementos dois a dois em relação a um critério comum, utilizando uma escala fundamental de valores absolutos que varia de 1 (igual importância) a 9 (extrema importância). Segundo Saaty (2008), esse processo converte julgamentos subjetivos e intangíveis em valores numéricos objetivos, gerando uma matriz de comparação a partir da qual são derivados os vetores de prioridade (pesos) que representam a influência relativa de cada elemento na decisão final.

Uma diferença do AHP em relação a outros métodos de decisão é a sua capacidade de medir a consistência lógica dos julgamentos. O método calcula um Índice de Consistência (CI) e uma Razão de Consistência (CR) para verificar se as avaliações do decisor são coerentes e transitivas. Saaty (1990) estabelece que um valor de CR inferior a 0,10 é aceitável, garantindo que a decisão não seja fruto de aleatoriedade. Ao final do processo, o método sintetiza as prioridades locais de cada nível para calcular uma prioridade global, resultando em um *ranking* quantitativo das alternativas que fundamenta de acordo com os critérios.

2.8 Custo em Nuvem Privada

A análise de custo em computação em nuvem é um processo complexo que transcende a simples comparação de preços, envolvendo a avaliação de diferentes modelos financeiros e operacionais. A decisão pela adoção de uma infraestrutura de nuvem privada, foco desta pesquisa, implica um modelo financeiro distinto da nuvem pública, que é melhor analisado pelo **Custo Total de Propriedade (TCO - Total Cost of Ownership)**.

O TCO é um *framework* de análise financeira projetado para avaliar o custo completo de um ativo de tecnologia ao longo de sua vida útil (SOUZA; MOREIRA; MACIEL, 2022). A análise TCO é fundamental para justificar o investimento em uma nuvem privada, pois revela custos que vão além da aquisição inicial do hardware, como manutenção, consumo de energia e despesas com pessoal (KHAN; KASHIF, 2021). Esses custos são classicamente divididos em duas categorias principais: Despesas de Capital (CAPEX) e Despesas Operacionais (OPEX).

Tabela 4 – Decomposição do Custo Total de Propriedade (TCO) em Nuvem Privada

Tipo de Custo	Descrição	Exemplos de Componentes
CAPEX (Despesas de Capital)	Investimentos iniciais em ativos de longa duração.	Aquisição de hardware (servidores, rede, armazenamento), licenças de softwares, custo de implementação inicial e infraestrutura do datacenter.
OPEX (Despesa Operacional)	Custos recorrentes necessários para manter a infraestrutura em funcionamento.	Consumo de energia elétrica, refrigeração, manutenção, salários da Equipe de TI, conectividade, espaço físico.

Fonte: Adaptado de Souza, Moreira e Maciel (2022)

2.8.1 Custo de Pessoal e a Influência da Complexidade

As despesas com pessoal representam um dos maiores componentes do OPEX em um datacenter (GRIEGO; LIRIA; CUESTA, 2021). A complexidade arquitetônica e operacional de uma plataforma de nuvem impacta diretamente a necessidade de pessoal especializado e o tempo gasto em tarefas de gerenciamento, provisionamento e manutenção. Plataformas com maior complexidade intrínseca podem exigir equipes maiores ou com maior nível de especialização, resultando em um custo de pessoal mais elevado. A literatura que compara OpenStack e CloudStack frequentemente aponta para a abordagem “*turnkey*” (pronta para uso) e integrada do CloudStack como sendo operacionalmente mais simples em contraste com a arquitetura modular e distribuída do OpenStack, que, embora mais flexível, demanda maior esforço de integração e gerenciamento (S. FILHO; G. FILHO, 2016).

2.8.2 Custo de Energia e a Relação com o Desempenho

O custo de energia elétrica é outro vetor fundamental do OPEX. Este custo não é fixo, mas diretamente proporcional ao consumo de recursos de hardware como *CPU* e memória (BELOGLAZOV; BUYYA, 2012). Portanto, a eficiência de uma plataforma — sua capacidade de realizar mais trabalho com menos recursos — tem uma implicação financeira direta. Plataformas que impõem um *overhead* computacional menor resultam em menor consumo de energia e, conseqüentemente, em menor custo operacional.

Para quantificar essa relação, é necessário modelar o consumo energético do hardware. A potência de um servidor não é constante; ele consome uma quantidade significativa de energia mesmo em estado ocioso (*idle power*). Conforme demonstrado por Barroso e Hölzle (2018), o consumo de um servidor ocioso pode facilmente ultrapassar 50% de seu consumo em pico. A partir desse ponto, a literatura estabelece que a relação entre a utilização de recursos e o consumo de energia pode ser aproximada com alta fidelidade por um **modelo linear**, que interpola entre o estado de ociosidade e o de carga máxima.

A potência média de consumo pode, então, ser aferida pela seguinte fórmula:

$$P_{\text{consumo}} = P_{\text{ociosa}} + (P_{\text{mxima}} - P_{\text{ociosa}}) \times U_{\text{recursos}} \quad (2.1)$$

Onde:

- **Pconsumo:** É a Potência Média de Consumo (em Watts).
- **Pociosa:** É a potência do servidor em estado ocioso.
- **Pmáxima:** É a potência máxima do servidor sob carga total.
- **Urecursos:** É a Utilização Média de Recursos (%), um valor empírico extraído dos testes de desempenho.

Adicionalmente, o custo total de energia de um datacenter deve considerar a infraestrutura de suporte, como a refrigeração. A eficiência energética de um datacenter é medida pelo indicador *Power Usage Effectiveness (PUE)*, que é a razão entre a energia total consumida pela instalação e a energia consumida apenas pelos equipamentos de TI (KOOMEY, 2011).

3 Trabalhos Relacionados

3.1 Introdução

Esta seção apresenta a análise dos trabalhos relacionados que fundamentam a presente pesquisa. A seleção destes estudos foi orientada pelo seguinte problema: qual plataforma de nuvem privada, OpenStack ou CloudStack, apresenta a melhor relação de desempenho e custo para suportar uma aplicação baseada em microsserviços sob diferentes cargas de trabalho e configurações de serviço? A análise alinha-se, ainda, ao objetivo de realizar uma comparação de desempenho e eficiência entre as referidas plataformas, utilizando a metodologia de planejamento de experimentos (DOE) e análise multicritério (AHP). O propósito deste levantamento é compreender e contextualizar as pesquisas que já abordaram o tema. Para a seleção desses artigos, foi realizada uma busca em bases de dados, incluindo Scopus, IEEE Xplore, ACM Digital Library, SpringerLink e o Portal de Periódicos Capes. A busca utilizou a string: “cloud computing **AND** multicriteria decision **OR** performance evaluation **OR** cost“. A partir dos resultados obtidos, foi aplicado um critério de inclusão: foram selecionados apenas os artigos relacionados a Avaliação de Desempenho e Avaliação de Custo em Ambientes de Nuvens. Esta filtragem resultou na seleção de 11 artigos, cuja análise é o foco das subseções seguintes.

3.2 Modelos para Tomada de Decisão considerando Desempenho

Esta seção agrupa os estudos em que a avaliação de métricas de desempenho em ambientes de nuvem (como tempo de resposta, latência, utilização de recursos e escalabilidade) constitui o núcleo da análise para a tomada de decisão.

Com o objetivo de sistematizar e reduzir a subjetividade na seleção de provedores de Computação em Nuvem, (HUAMANÍ, 2014) propôs um modelo baseado em Processo de Análise Hierárquica - AHP. A principal contribuição do seu trabalho foi a inclusão explícita do critério “Rendimento” (Desempenho) para medir a velocidade e agilidade do serviço, ao lado de outros critérios como Escalabilidade e Acessibilidade. A metodologia envolveu a estruturação de uma hierarquia de decisão e a realização de comparações

pareadas. O resultado da aplicação do modelo por três grupos distintos de avaliadores variou (dois selecionaram o Google e um a Microsoft), evidenciando que a classificação final é sensível aos critérios de escolha dos avaliadores. O estudo justifica a abordagem AHP como uma ferramenta eficaz para formalizar a compreensão intuitiva em problemas multicritério, reforçando a importância da avaliação direta do desempenho.

Mishra (2020) focou na necessidade de otimizar a distribuição de recursos em ambientes híbridos de computação em névoa (*fog-cloud*), propondo políticas de alocação baseadas em AHP com o objetivo primordial de reduzir o atraso (*delay*) total das tarefas, uma métrica de desempenho crítica. As políticas foram concebidas para considerar simultaneamente as cargas de computação e de rede. A validação, realizada no simulador iFogSim, demonstrou que as políticas baseadas em AHP apresentaram desempenho superior aos esquemas preexistentes, com menor tempo de resposta e uma melhoria de eficiência de até 10% nos recursos. A pesquisa conclui que o fator determinante para o desempenho superior foi a inclusão da carga da rede no processo de decisão, uma contribuição fundamental para o desenvolvimento de estratégias de alocação de recursos mais eficientes.

Kurup e Guruprasad (2022) tiveram como objetivo validar técnicas para a seleção da infraestrutura de nuvem ideal em ambientes de computação móvel (MCC), onde o desempenho é um fator crítico. A metodologia adotou o AHP para a ponderação de seis parâmetros de Qualidade de Serviço (QoS), onde critérios de performance como “Tempo de Resposta” e “Velocidade” receberam maior relevância. Subsequentemente, três métodos de ranqueamento (TOPSIS, PROMETHEE II e MOORA) foram aplicados em paralelo. O resultado foi uma notável consistência: todos os métodos identificaram o “Cloud 1” como o servidor ideal. A robustez da conclusão foi o principal achado, confirmada por uma rigorosa análise de sensibilidade. A contribuição do trabalho é a validação de uma metodologia consistente e a demonstração da importância da análise de sensibilidade para garantir a confiabilidade de decisões baseadas em desempenho.

Com o objetivo de aprimorar a eficiência e a confiabilidade da tomada de decisão em grupo, Mostafa (2024) propôs o Group Best-Only Method (GBOM), uma nova abordagem para Tomada de Decisão Multicritério - MCDM. Embora o método seja uma contribuição

metodológica, sua validação empírica foi fortemente centrada no desempenho. A eficácia do Group Best-Only Method - GBOM foi ilustrada em um estudo de caso que avaliou dez provedores com base em cinco critérios de QoS, incluindo desempenho de *CPU*, memória e disco. O resultado classificou o Processo de Análise Hierárquica - AHP como a melhor alternativa, uma escolha atribuída diretamente à sua superioridade no critério de desempenho mais valorizado (Desempenho da *CPU*). A contribuição do trabalho é um método de decisão mais ágil, mas também o reforço de que o desempenho computacional é um pilar fundamental na seleção final de um provedor.

3.3 Modelos para Tomada de Decisão considerando Avaliação de Custo

Esta seção apresenta os trabalhos que adotam a avaliação de custo como critério para tomada de decisão.

Sharma e Sehrawat (2020) investigaram os determinantes da adoção da nuvem no setor de saúde indiano, utilizando uma abordagem híbrida (Modelagem Estrutural InterpretativaI - SM, Processo Analítico Hierárquico - AHP e Técnica para Ordem de Preferência por Similaridade com a Solução Ideal - TOPSIS). Embora tenham avaliado múltiplos critérios, a análise do custo foi central. Na metodologia, o custo foi tratado como um “fator de custo” no ranqueamento com TOPSIS (onde valores menores são melhores). O resultado indicou que o provedor mais adequado (CSP5) não era a opção de menor custo. A principal contribuição do trabalho, no que tange ao custo, foi demonstrar que os benefícios agregados, especialmente em segurança de dados, foram ponderados como mais significativos do que a desvantagem de custo. O estudo posicionou o custo como um critério fundamental, mas não como o único fator decisório.

Visando solucionar os desafios na combinação de serviços para nuvens híbridas, Park (2020) propôs um Processo de Seleção e Integração de Serviços (C-SIP). A principal contribuição do trabalho em relação ao custo foi tratá-lo não apenas como um critério de ponderação (“custo-eficácia”), mas também como uma restrição eliminatória. A metodologia permite que o usuário defina um “orçamento máximo” que funciona como um filtro para remover alternativas financeiramente inviáveis antes mesmo da avaliação de desempenho. O

estudo de caso demonstra essa funcionalidade ao descartar automaticamente combinações de serviços que excediam o orçamento estabelecido. Essa abordagem oferece uma solução para a gestão de custos no processo de seleção.

Ramos e Justo (2023) empregaram o método AHP para selecionar o modelo de implantação de nuvem mais apropriado para uma instituição de ensino pública. O custo foi um dos quatro critérios principais e foi detalhado em subcritérios (infraestrutura, mensalidade, etc.). Embora tenha recebido a menor prioridade na ponderação inicial (13,5%), o resultado mostrou que o custo funcionou como um fator estratégico na escolha final. A nuvem híbrida foi indicada como a alternativa mais adequada, justificada por uma estratégia de otimização de custos a longo prazo, permitindo à instituição utilizar sua infraestrutura de datacenter já existente. A contribuição é demonstrar como o custo, mesmo com baixo peso, pode ser o vetor de uma decisão estratégica de longo prazo.

Bhola, Mohanty^b e Pattnaik^c (2024) propuseram uma abordagem AHP-TOPSIS para a seleção de provedores de *Machine Learning as a Service* (MLaaS), onde o custo foi o critério decisivo. O objetivo era estruturar essa complexa decisão. A metodologia utilizou o AHP para determinar os pesos de dezessete critérios. O principal resultado da ponderação foi que o critério “custo-Eficácia” recebeu o maior peso (11%) entre todos os avaliados. Consequentemente, ao ser aplicado no TOPSIS, este peso elevado influenciou diretamente a classificação final das alternativas. A contribuição do estudo é indicar que, para serviços especializados como o *Machine Learning as a Service* - MLaaS, a relação entre o investimento financeiro e o retorno obtido é um fator de consideração primário no processo de seleção.

3.4 Modelos para Tomada de Decisão para Ambientes de Nuvem

Esta seção apresenta propostas de métodos de tomada de decisão em ambientes de nuvem, os quais oferecem contribuições e melhorias ao método AHP.

Tiwari et al. (2023) desenvolveram uma extensão do método TOPSIS, denominada Técnica para Ordem de Preferência por Similaridade com a Solução Ideal - Ordem Mista - Livre de Inversão de Posto - RMO-TOPSIS, com o objetivo de lidar com a heterogeneidade

das informações de QoS e, crucialmente, eliminar o problemático fenômeno da inversão de *rank*. O artigo realiza um estudo de caso para mostrar a aplicação do RMO-TOPSIS no campo da seleção de serviços em nuvem com base em desempenho e custo. A principal inovação metodológica é introdução de duas alternativas fictícias para estabilizar a matriz de decisão, a função primária dessas alternativas fictícias é estabilizar o valor normalizado de cada parâmetro de Qualidade de Serviço (QoS) das alternativas. O resultado, validado por uma análise de sensibilidade, confirmou que o RMO-TOPSIS é imune à inversão de *rank*. A principal contribuição do método é sua capacidade de processar diretamente múltiplos tipos de dados sem a necessidade de transformações, o que não apenas previne a perda de informação, mas também aumenta substancialmente a consistência e a confiabilidade na classificação das alternativas.

Petrov (2021) apresentou uma abordagem combinada para a seleção de serviços em nuvem com base em desempenho e custo com o objetivo de otimizar a sinergia entre a opinião de especialistas e a objetividade dos dados. A metodologia proposta estrutura a AHP “em blocos”, que agrupa os critérios em categorias lógicas para reduzir a complexidade das comparações pareadas. O resultado demonstrou uma otimização expressiva, com a sistematização em blocos reduzindo o número de comparações em mais de 4,6 vezes e alcançando um notável índice de consistência. A pesquisa evidencia que a integração dos juízos de especialistas com a análise de entropia resulta em uma combinação equilibrada de profissionalismo e objetividade, tornando a metodologia particularmente relevante para aplicações em larga escala.

Diante das limitações de eficiência e escalabilidade do AHP, (MISHRA; KUMAR, 2025) propuseram metodologias híbridas. O objetivo foi demonstrar a capacidade de abordagens baseadas no *Best-Worst Method* (BWM) para superar os desafios de cenários com múltiplos critérios de avaliação baseada em desempenho e custo. A metodologia emprega o BWM para calcular os pesos dos critérios, técnica escolhida por sua maior eficiência ao exigir menos comparações pareadas que o AHP. O resultado da análise de eficiência confirmou a superioridade do BWM. A contribuição do estudo é a validação de abordagens híbridas baseadas em BWM como uma solução mais escalável e eficiente para

o problema de seleção de serviços em nuvem.

3.5 Comparação com os trabalhos relacionados

Os trabalhos relacionados destacam que a seleção de um serviço ou de um modelo de implantação em um mercado competitivo representa um desafio estratégico, com implicações diretas em custos, segurança e desempenho (RAMOS; JUSTO, 2023).

Para endereçar essa complexidade, os estudos apontam para métodos de Tomada de Decisão Multicritério (MCDM) como o arcabouço metodológico mais adequado. A justificativa para o uso de MCDM é a necessidade de avaliar alternativas com base em múltiplos critérios conflitantes, permitindo formalizar o conhecimento de especialistas e transformar julgamentos qualitativos em dados quantificáveis (HUAMANÍ HUAMANÍ, 2013; MISHRA; KUMAR, 2025). A aplicação desses métodos nos artigos analisados frequentemente segue um *design* de duas fases principais.

A primeira foca na ponderação dos critérios, onde métodos como o Processo Analítico Hierárquico (AHP) ou o *Best-Worst Method* (BWM) são usados para determinar a importância relativa de cada fator. A segunda fase é a classificação das alternativas, na qual os pesos obtidos são utilizados como entrada para algoritmos de ranqueamento, como TOPSIS, PROMETHEE, entre outros, que processam os dados de desempenho para gerar a classificação final (BHOLA et al., 2024; KURUP; GURUPRASAD, 2022).

A Tabela 5 apresenta uma análise comparativa do escopo e das contribuições dos trabalhos relacionados a esta dissertação. A comparação entre os estudos é realizada com base na abordagem de nove eixos de análise: a utilização de Análise Multicritério com AHP (C1) ou de uma Análise Multicritério Geral (C2); a presença de uma Avaliação de Desempenho (C3), de um *Design* de Experimentos (C4) e de uma Avaliação de custo (C5); o foco do estudo em Nuvem Privada (C6) ou Nuvem Pública (C7); e a condução de um Estudo Comparativo Empírico entre plataformas (C8) ou de Testes de Carga com *Workload* de Microsserviços/APIs (C9). A partir desta análise, pode-se notar que a totalidade dos trabalhos selecionados emprega algum tipo de Análise Multicritério (C1, C2) como ferramenta para a Avaliação de Desempenho (C3), indicando a consolidação

desta abordagem metodológica para o problema de seleção e análise de serviços em nuvem.

Tabela 5 – Comparativa de Trabalhos Relacionados

Trabalhos	C1	C2	C3	C4	C5	C6	C7	C8	C9
(Sharma & Sehrawat, 2020)	X	X		X	X				
(Park et al., 2020)	X		X	X	X		X		
(Tiwari et al., 2023)		X	X	X	X				
(Mostafa, 2024)	X	X	X	X	X		X		
(Petrov, 2021)	X	X	X	X	X				
(Mishra et al., 2020)	X	X	X	X					
(Huamaní Huamaní, 2014)	X	X	X	X			X		
(Ramos & Justo, 2023)	X	X		X	X	X	X		
(Kurup, 2022)	X	X	X	X	X				
(Mishra & Kumar, 2025)	X	X		X			X		
(Bhola et al., 2024)	X	X		X	X		X		
Esta dissertação	X		X	X	X	X	X	X	X

Fonte: O autor

A análise comparativa dos trabalhos relacionados indica que, embora todos empreguem métodos de Tomada de Decisão Multicritério (MCDM) para problemas no domínio da computação em nuvem, seus focos e escopos apresentam variações. A maior parte dos trabalhos relacionados apresentados concentra-se na avaliação e seleção de provedores de nuvem pública (C7). Estudos como os de Sharma e Sehrawat (2020), Bhola, Mohanty e Pattnaik (2024), Mostafa (2024) e Kurup e Guruprasad (2022) aplicam diferentes abordagens híbridas de MCDM (C1, C2) para classificar serviços de nuvem pública com base em critérios de Qualidade de Serviço, custo e outros determinantes. Outros trabalhos, como os de Park et al. (2020) e Mishra et al. (2020), estendem essa análise para arquiteturas híbridas, mas ainda com o objetivo de selecionar ou alocar recursos em serviços de nuvem pública que se integram a outros ambientes. Em contraste, um número menor de estudos aprofunda a análise da nuvem privada (C6).

O trabalho de Ramos e Justo (2023), por exemplo, utiliza o AHP para avaliar modelos de implantação, incluindo a nuvem privada como uma alternativa explícita e concluindo com a indicação de um modelo híbrido.

A presente dissertação se diferencia ao conduzir o único Estudo Comparativo Empírico (C8) direto entre duas plataformas de nuvem privada, OpenStack e CloudStack. Para isso, utiliza um *Design* de Experimentos (C4) para planejar e executar Testes de Carga com *Workload* de Microsserviços/APIs (C9), coletando métricas para uma Avaliação de Desempenho (C3) detalhada. Sua Avaliação de Custo (C5) é particularmente aprofundada, pois desenvolve um modelo de Custo Total de Propriedade (TCO) detalhado para o ambiente de nuvem privada. O diferencial desta análise é que o modelo adotado considera critérios de desempenho e de custo operacional. O Custo de Energia Elétrica (OPEX) é calculado com base nos dados de utilização de recursos aferidos nos testes de desempenho, e o Custo da Equipe Técnica (OPEX) é modelado com base na complexidade de gerenciamento de cada plataforma. Os dados de custo (TCO) resultantes, juntamente com as métricas de desempenho coletadas, são então ponderados pelo método AHP (C1) para gerar uma classificação final. Desta forma, enquanto os 11 trabalhos relacionados aplicam o MCDM à seleção de serviços de nuvem pública, esta dissertação contribui ao focar na análise de desempenho empírico e no custo TCO detalhado de plataformas de nuvem privada.

A investigação da literatura revelou que, para endereçar o problema da seleção de serviços em nuvem, há uma consolidação no uso de métodos de Tomada de Decisão Multicritério (MCDM). A abordagem predominante na maioria dos estudos consiste na aplicação de modelos híbridos, combinando técnicas de ponderação, como AHP e/ou BWM, com métodos de ranqueamento, como TOPSIS, ARAS e COPRAS para avaliar e selecionar provedores de nuvem pública (SHARMA; SEHRAWAT, 2020; MISHRA; KUMAR, 2025).

A análise, contudo, permitiu identificar uma lacuna na literatura. Observa-se uma escassez de trabalhos que realizam uma análise comparativa de desempenho, de forma empírica e direta, entre plataformas específicas de nuvem privada. Nenhum dos artigos selecionados combinou um *Design* de Experimentos (DOE) com testes de carga

utilizando *workloads* de microsserviços e APIs para comparar diretamente o desempenho de plataformas como OpenStack e CloudStack. Adicionalmente, a abordagem de avaliação de custo (TCO) desenvolvida nesta dissertação representa uma contribuição não encontrada nos trabalhos correlatos. O modelo adotado considera critérios de desempenho e custo financeiro de energia, além do custo da equipe técnica associado à complexidade de gerenciamento de cada plataforma, em vez de focar apenas em custos de aquisição ou em métricas de custo de nuvem privada.

Diante do exposto, o capítulo seguinte apresentará a metodologia desenvolvida para esta dissertação, projetada para preencher a lacuna identificada. Será detalhado o método para a avaliação de desempenho e de custo de duas infraestruturas de nuvem privada, OpenStack e CloudStack, sob um *workload* de microsserviços. A metodologia proposta integra o Planejamento de Experimentos para a coleta de dados, uma análise de custo, e a aplicação do AHP para sintetizar os resultados de desempenhos e custos, fornecendo uma base para a seleção da plataforma de nuvem mais adequada ao cenário estudado.

4 Metodologia para Planejamento e Tomada de Decisão

Esta seção detalha a metodologia adotada para auxiliar na tomada de decisão sobre a escolha de nuvens privadas, considerando as métricas de desempenho e custo. A pesquisa empregou uma metodologia que combina duas técnicas: o Planejamento de Experimentos (DOE) e o Processo de Análise Hierárquica (AHP).

Para a coleta de dados, o DOE foi escolhido em detrimento de abordagens de teste não estruturadas (como *ad-hoc* ou um-fator-por-vez) por ser a única capaz de isolar estatisticamente a influência de variáveis concorrentes, como a oferta de serviço e a carga de trabalho, garantindo a confiabilidade dos dados de entrada. Já para a tomada de decisão, embora existam outros métodos multicritério na literatura, como TOPSIS ou PROMETHEE, o AHP foi a opção preferencial devido à sua capacidade de estruturar hierarquicamente o problema e resolver os *trade-offs* entre critérios conflitantes. Diferente de métodos baseados apenas em distância ou superação, o AHP utiliza comparações paritárias para ponderar matematicamente o equilíbrio entre a eficiência técnica e o custo financeiro, alinhando-se diretamente ao objetivo de justificar a escolha da plataforma.

O Planejamento de Experimentos (DOE) foi utilizado para estruturar os dados coletados. Diante de múltiplas variáveis que poderiam influenciar o desempenho, como o tamanho da máquina virtual e a intensidade da carga de trabalho. O DOE ofereceu um método estruturado para testar o impacto de cada um desses fatores de forma controlada, permitindo isolar os efeitos principais e as interações entre as variáveis, o que seria inviável com abordagens experimentais não sistematizadas (MONTGOMERY, 2019).

O AHP decompõe o problema de decisão em uma estrutura hierárquica e utiliza comparações em pares para quantificar a importância relativa de cada critério. Este processo traduz as prioridades definidas na pesquisa em pesos numéricos, permitindo a síntese de múltiplos indicadores em um escore final de preferência para cada alternativa (SAATY, 1990).

A metodologia proposta é composta de nove atividades, iniciando com a análise do objetivo do estudo e seleção de métricas. Em seguida, prossegue com a configuração dos ambientes, a implementação do microsserviço, o planejamento de experimentos (DOE) e

a medição de desempenho das plataformas. As etapas finais consistem na elaboração do modelo de custo, na análise estatística dos dados, na análise comparativa dos cenários de desempenho e custo, e concluem na tomada de decisão através do método AHP.

Para estruturar o fluxo de trabalho e facilitar a compreensão da abordagem adotada, essas nove atividades foram organizadas em quatro fases sequenciais: **Planejamento**, **Execução**, **Análise de Custo e Desempenho**, e **Tomada de Decisão Final**. Essa divisão reflete a evolução lógica da pesquisa, partindo da definição experimental até a síntese decisória. A seguir, detalha-se o escopo e os procedimentos realizados em cada uma dessas fases.

Fase 1: Planejamento

Esta fase inicial estabelece o alicerce experimental da pesquisa, definindo as bases para a comparação justa entre as plataformas. A etapa começa com a análise dos objetivos e a Seleção das Métricas, onde são definidos os indicadores de desempenho (como CPU, memória e latência) e custo (TCO) fundamentados na literatura. Na sequência, realiza-se a Configuração dos Ambientes, garantindo a paridade de hardware entre OpenStack e CloudStack, seguida pela Implantação da Aplicação, onde o microserviço de teste é instalado nas máquinas virtuais. A fase encerra-se com o Planejamento dos Experimentos (DOE), que estrutura estatisticamente os cenários de teste, definindo as combinações de fatores (oferta de serviço e carga) necessárias para isolar as variáveis de interesse.

Fase 2: Execução

A fase de execução concentra-se na operacionalização dos experimentos definidos no planejamento para a coleta de dados empíricos. O foco principal é a atividade de Medir o Desempenho, que consiste na execução controlada dos testes de carga utilizando ferramentas como o Apache JMeter. Durante esta etapa, o ambiente é submetido aos cenários de estresse estipulados, enquanto ferramentas de monitoramento (Zabbix/Grafana) capturam, simultaneamente, a perspectiva do cliente (tempos de resposta e vazão) e a perspectiva do servidor (consumo de recursos via SNMP/agentes), gerando o artefato de dados brutos essenciais para as análises subsequentes.

Fase 3: Análise de Custo e Desempenho

Esta fase dedica-se ao processamento e à interpretação dos dados coletados, transformando registros brutos em informações estratégicas. Inicia-se com o Desenvolvimento do Modelo de Custo, onde as equações de TCO são aplicadas para traduzir a eficiência técnica em valores monetários de CAPEX e OPEX. Paralelamente, realiza-se a Análise Estatística dos Dados, consolidando as múltiplas réplicas dos testes em médias e desvios-padrão robustos. O processo culmina na Análise dos Cenários, uma avaliação comparativa que confronta os resultados de cada plataforma, identificando os *trade-offs* entre a eficiência de recursos, a estabilidade do serviço e o custo mensal operacional.

Fase 4: Tomada de Decisão Final

A fase final integra os resultados quantitativos processados em um *framework* de decisão estruturado para solucionar o problema de escolha. A atividade central é a Tomada de Decisão com o Método AHP, que decompõe o problema hierarquicamente. Nesta etapa, os critérios de desempenho e custo são ponderados através de comparações paritárias, e os dados normalizados das plataformas são sintetizados em um vetor de prioridades. O resultado é a classificação matemática das alternativas, permitindo a seleção justificada da plataforma de nuvem privada que oferece o melhor equilíbrio entre os requisitos técnicos e econômicos estabelecidos pelo estudo.

A Figura 4 detalha a sequência das atividades (representadas por retângulos) e suas respectivas saídas que detalha os artefatos gerados e o fluxo lógico que conecta as fases de planejamento, execução, análise de custo e desempenho, e a tomada de decisão final.

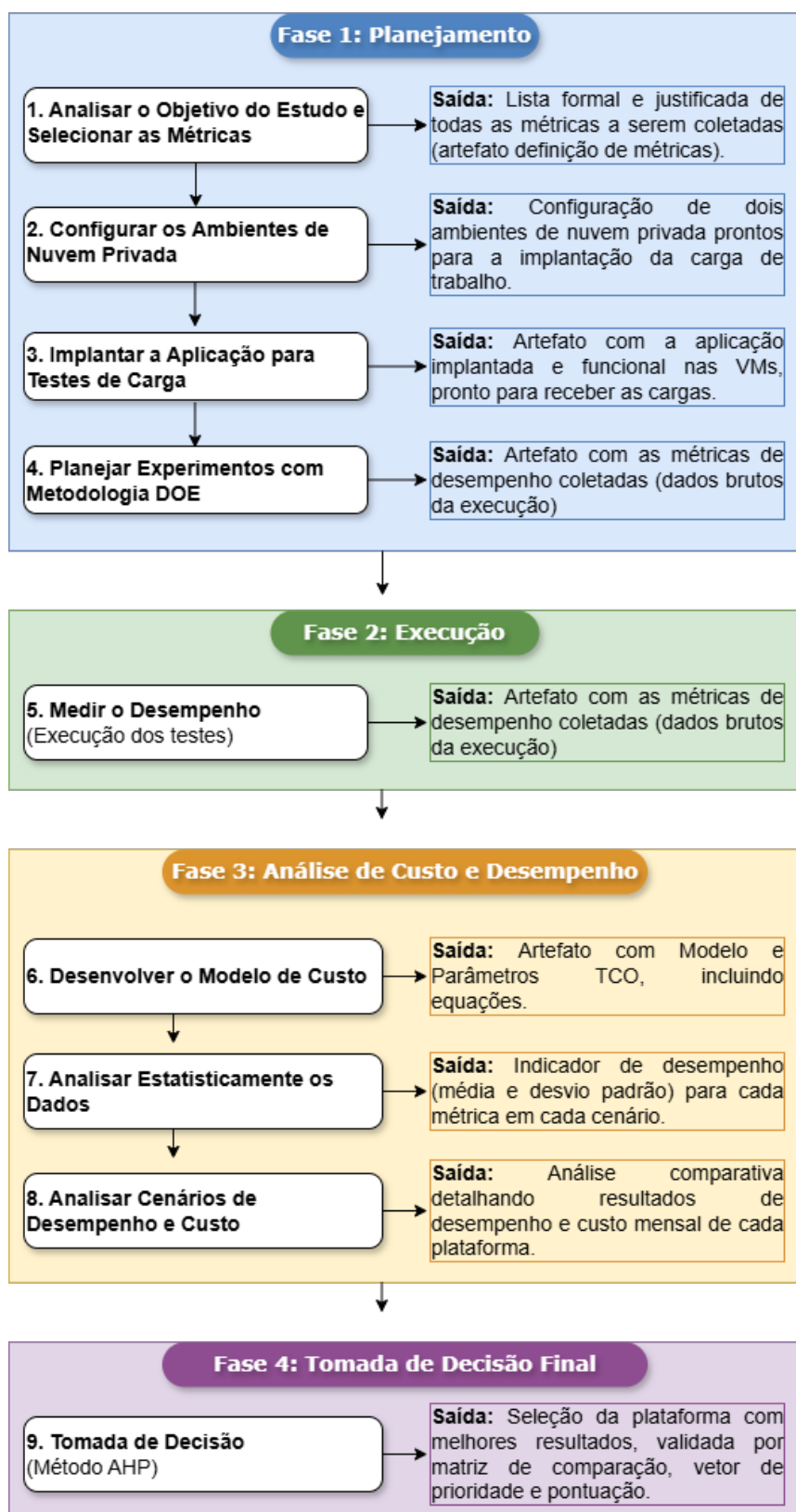


Figura 4 – Metodologia

Fonte: O autor

Para proporcionar uma compreensão do processo, cada uma das atividades apresentadas no diagrama da metodologia será detalhada a seguir.

4.1 Analisar o Objetivo do Estudo e Selecionar as Métricas

O objetivo desta atividade é estabelecer a base quantitativa desta pesquisa, definindo os indicadores de desempenho e custo que serão utilizados para a avaliação comparativa das plataformas. Para isso, a seleção das métricas é fundamentada na literatura sobre avaliação de desempenho em nuvem e arquiteturas de microsserviços (GÓMEZ; PÉREZ; ROJAS, 2022).

As métricas de desempenho são tipicamente divididas em duas categorias principais. A primeira abrange indicadores de desempenho, que medem o consumo e o *overhead* da plataforma, como a utilização de processador e de memória. A segunda categoria foca na qualidade de serviço (QoS) percebida pelo cliente, incluindo métricas como tempo de resposta, latência e vazão (throughput), que quantificam a agilidade e a capacidade de processamento do sistema.

De forma complementar aos indicadores de desempenho, a análise de custo adota como métrica o Custo Total de Propriedade (TCO), utilizado para avaliar o custo completo de um ativo de tecnologia ao longo de sua vida útil. Sua aplicação em nuvens privadas envolve a decomposição dos custos em Despesas de Capital (CAPEX), que incluem os investimentos iniciais em hardware amortizados ao longo do tempo, e Despesas Operacionais (OPEX), que representam os custos recorrentes para manter a infraestrutura, como o consumo de energia e os salários da equipe técnica.

O resultado esperado desta etapa é, portanto, uma lista formal de todas as métricas a serem coletadas, constituindo o artefato definição de métricas.

4.2 Configurar os Ambientes de Nuvem com as Plataformas OpenStack e CloudStack

Esta atividade cria os ambientes de nuvem privada experimentais com funcionalidades idênticas, eliminando variações de infraestrutura como causa de eventuais diferenças de

desempenho. Para isso, o procedimento envolve a instalação e configuração das plataformas de nuvens privadas em servidores físicos com hardware idêntico (processador, memória e disco). A topologia da rede deve ser projetada para isolar as funções de geração de carga, monitoramento e execução, garantindo que os recursos de teste não interfiram nos recursos medidos.

O resultado esperado é a configuração de dois ambientes de nuvem privada prontos para a implantação da carga de trabalho.

4.3 Implantar a Aplicação para a Execução dos Testes de Carga

O objetivo é implantar uma aplicação, que funcionará como a carga de trabalho padronizada, em máquinas virtuais idênticas em cada ambiente de nuvem. Essa ação garante que o sistema a ser testado seja o mesmo em ambas as plataformas, assegurando que a única variável na comparação seja a própria plataforma de gerenciamento de nuvem privada.

O resultado esperado é o artefato com a aplicação implantada e funcional nas VMs, pronto para receber as cargas de trabalho.

4.4 Planejar os Experimentos por Meio da Metodologia DOE

O Planejamento de Experimentos (DOE) consiste em uma abordagem estatística sistemática utilizada para planejar, conduzir, analisar e interpretar testes controlados. Segundo Montgomery (2019), essa metodologia permite avaliar simultaneamente o efeito de múltiplos fatores de entrada (variáveis independentes) sobre uma ou mais variáveis de resposta (desempenho), bem como identificar as interações entre esses fatores.

Esta atividade visa a estruturação do plano de experimentos utilizando a metodologia DOE, que permite isolar estatisticamente a influência de variáveis concorrentes e identificar as interações entre elas. Para este estudo, adotou-se um Planejamento Fatorial Completo 3x3, assegurando que todas as combinações possíveis entre os fatores fossem sistematicamente testadas. Foram definidos como fatores (variáveis independentes) a “Oferta de Serviço” (com níveis Small, Medium e Large) e a “Intensidade de Carga” (500, 700 e

900 requisições). Essa configuração resultou em 9 cenários de teste exclusivos para cada plataforma, garantindo a obtenção de dados estatisticamente válidos do comportamento das nuvens privadas sob análise.

O resultado do DOE é o plano de experimentos, um conjunto de cenários de teste que cobrem as combinações de fatores e seus níveis, com um número definido de réplicas para cada cenário, a fim de garantir os dados estatísticos.

4.5 Medir o Desempenho das Plataformas por Meio da Execução dos Testes

Esta atividade da metodologia executa testes de carga planejados para a coleta de dados brutos de todas as métricas de desempenho. O procedimento consiste em utilizar uma ferramenta de geração de carga (como o Apache JMeter) para aplicar a carga definida em cada cenário e coletar as métricas da perspectiva do cliente (tempo de resposta, latência, etc.). Simultaneamente, ferramentas de monitoramento (como Zabbix/Grafana) devem ser usadas para coletar as métricas de utilização de recursos (*CPU* e memória) da perspectiva do servidor. Para assegurar a integridade do monitoramento, o protocolo ICMP é empregado para verificações constantes de conectividade e disponibilidade das instâncias, enquanto a coleta detalhada dos indicadores de consumo de hardware é realizada via protocolo SNMP (ou através de agentes de monitoramento nativos), garantindo a precisão dos dados sistêmicos coletados. Essa atividade terá como resultado o artefato com as métricas de desempenho coletadas, contendo os dados brutos da execução.

4.6 Desenvolver o Modelo de Custo

Essa seção apresenta os modelos de custo desenvolvidos para calcular o custo de hardware, equipe técnica e consumo de energia, com base no modelo TCO. TCO é fundamental para justificar o investimento em uma nuvem privada, pois revela custos que vão além da aquisição inicial do hardware (KHAN; KASHIF, 2021). O modelo visa traduzir a eficiência e os requisitos operacionais de cada plataforma em um valor monetário mensal estimado, fornecendo um *framework* que pode ser adaptado para cenários de diferentes escalas.

O TCO foi decomposto em três elementos principais: Custo de Hardware (CAPEX), Custo da Equipe Técnica (OPEX) e Custo de Consumo da Energia Elétrica (OPEX). Essa atividade gera o artefato com o Modelo e Parâmetros TCO, incluindo suas equações.

4.6.1 Custo de Hardware (Capex)

O primeiro componente do Custo Total de Propriedade (TCO) quantifica o investimento de capital (CAPEX) em toda a infraestrutura física. Para uma análise TCO completa, é essencial considerar não apenas os servidores, mas todos os ativos de hardware que compõem a nuvem privada, como servidores, *switches*, roteadores.

Para traduzir este investimento inicial em uma despesa mensal recorrente, o custo total de aquisição de todos os ativos é somado e, em seguida, amortizado ao longo de um tempo de vida útil padrão. A equação 4.1 calcula o custo mensal total de hardware amortizado, considerando a soma dos valores de aquisição dos servidores e dos demais ativos de rede, dividida pelo tempo de vida útil estipulado.

Explicação dos Elementos da Equação:

$C_{hardware}$: Representa o Custo Mensal Total de Hardware Amortizado (R\$/mês). É a parcela mensal do investimento de capital que impacta o TCO.

$\sum_{i=1}^n V_{servidor_i}$: É o somatório que representa o custo total de aquisição de todos os servidores, desde o primeiro servidor (i=1) até o último (n).

$\sum_{k=1}^p V_{ativo_rede_k}$: Corresponde ao somatório do custo total de aquisição de todos os outros ativos de rede (como *switches*, roteadores), desde o primeiro ativo (k=1) até o último (p).

T_{vida} : É o Tempo de Vida Útil padrão definido para a amortização de todos os ativos de hardware da infraestrutura, medido em meses.

Equação:

$$C_{hardware} = \frac{(\sum_{i=1}^n V_{servidor_i}) + (\sum_{k=1}^p V_{ativo_rede_k})}{T_{vida}} \quad (4.1)$$

4.6.2 Custo da Equipe Técnica (OPEX)

O Custo da Equipe Técnica é um dos componentes mais significativos das despesas operacionais (OPEX) de uma nuvem privada. Este custo não depende apenas do nível de qualificação da equipe, mas também do tempo efetivamente gasto em tarefas de gerenciamento, manutenção e solução de problemas. Plataformas mais complexas tendem a demandar um maior número de horas de trabalho da equipe de TI, elevando os custos operacionais (GUPTA; SHARMA, 2023).

Para modelar este custo de forma precisa, a equação 4.2 calcula o custo total da equipe técnica com base na quantidade de horas de trabalho estimadas para a gestão da plataforma e no custo por hora de cada membro da equipe.

Explicação da equação:

C_{pessoal} : Custo Mensal Total da Equipe Técnica (R\$/mês).

$\sum_{j=1}^m$: É o somatório que percorre cada membro da equipe, desde o membro ($j=1$) até o último membro (m).

H_{gesto_j} : Total de Horas Mensais de Gestão estimadas para o j -ésimo membro da equipe. Este é o fator que quantifica o esforço. Plataformas mais complexas exigem mais horas de gestão.

V_{hora_j} : Valor da Hora de Trabalho do j -ésimo membro da equipe (em R\$/hora), derivado de seu salário e nível de qualificação.

Equação:

$$C_{\text{pessoal}} = \sum_{j=1}^m (H_{\text{gesto}_j} \times V_{\text{hora}_j}) \quad (4.2)$$

4.6.3 Custo de Energia Elétrica (OPEX)

O custo total de consumo de energia é o somatório do custo de consumo de energia de cada componente da infraestrutura de comunicação. A equação 4.3 primeiro soma o consumo de energia em kWh de todos os servidores e depois aplica os fatores de custo (tarifa e PUE).

Explicação da equação:

- C_{energia} : Custo Mensal Total de Energia Elétrica (R\$/mês).
- $\sum_{i=1}^n$: Somatório que calcula a energia total (em kWh) consumida por todos os n servidores.
- P_{consumo_i} : Potência Média de Consumo do i -ésimo servidor (em Watts).
- H_{ms} : Total de horas em um mês (padrão de 720h).
- 1000: para converter a unidade de Watt-hora para kilowatt-hora (kWh), que é a unidade padrão para a cobrança de energia.
- V_{tarifa} : Valor da Tarifa de Energia Elétrica (R\$/kWh).
- PUE: *Power Usage Effectiveness* do datacenter. O valor adotado foi uma estimativa baseada na literatura para representar a eficiência de um datacenter tradicional. (KOOMEY, 2011)

Equação:

$$C_{\text{energia}} = \left(\sum_{i=1}^n \frac{P_{\text{consumo}_i} \times H_{\text{ms}}}{1000} \right) \times V_{\text{tarifa}} \times \text{PUE} \quad (4.3)$$

Onde a Potência de Consumo para cada servidor (P_{consumo_i}) é:

$$P_{\text{consumo}_i} = P_{\text{ociosa}_i} + (P_{\text{mxima}_i} - P_{\text{ociosa}_i}) \times U_{\text{recursos}_i}$$

P_{consumo_i} : É a Potência Média de Consumo do i -ésimo servidor (em Watts). Este é o resultado que se deseja calcular e representa o consumo médio de energia do servidor durante a execução dos testes.

P_{ociosa_i} : É a Potência Ociosa do i -ésimo servidor (em Watts). Representa o consumo de energia base do servidor apenas por estar ligado, antes de processar qualquer carga de trabalho. Conforme demonstrado por Barroso e Hölzle (2018), este valor é significativo e pode corresponder a mais de 50% da potência de pico.

$P_{\text{mxima}_i} - P_{\text{ociosa}_i}$: Este termo representa a faixa de potência dinâmica do i -ésimo servidor. É a quantidade de energia que o servidor pode alocar para o processamento de cargas de trabalho, ou seja, a diferença entre seu consumo máximo e seu consumo base.

U_{recursos_i} : É a Utilização Média de Recursos do i -ésimo servidor (em percentual). Este é o dado empírico extraído diretamente dos seus testes de desempenho (como a média

da utilização de *CPU* e memória). Ele atua como um multiplicador que define qual porção da faixa de potência dinâmica está sendo efetivamente utilizada.

4.6.4 Custo Total de Propriedade (TCO)

O Custo Total de Propriedade (TCO) é calculado pela Equação 4.4 que soma os custos de hardware, pessoal e energia elétrica, consolidando todos os custos de capital amortizados e os custos operacionais do período. O resultado dessa equação é o custo mensal estimado (R\$/mês) para cada plataforma, que constitui um robusto indicador para a tomada de decisão.

Equação:

$$TCO = (C_{hardware} = \frac{(\sum_{i=1}^n V_{servidor_i}) + (\sum_{k=1}^p V_{ativo_rede_k})}{T_{vida}}) + (C_{pessoal} = \sum_{j=1}^m (H_{gesto_j} \times V_{hora_j})) + (C_{energia} = \left(\sum_{i=1}^n \frac{P_{consumo_i} \times H_{ms}}{1000} \right) \times V_{tarifa} \times PUE) \quad (4.4)$$

4.7 Analisar Estatisticamente os Dados de Desempenho Coletados

O objetivo desta atividade é processar os dados brutos coletados na fase de medição. O procedimento consiste em agregar os resultados das múltiplas réplicas de cada cenário de teste. As principais medidas são média aritmética e desvio-padrão, resultando em um indicador de desempenho robusto para cada métrica em cada cenário.

4.8 Analisar os Cenários de Desempenho e Custo para Cada Plataforma

O objetivo desta etapa é realizar a análise interpretativa dos dados já tratados. O processo é dividido em três atividades:

- 1) **Análise Comparativa de Desempenho:** Comparar diretamente as métricas de desempenho consolidadas para cada plataforma, utilizando gráficos e tabelas para identificar os pontos fortes (como a eficiência dos recursos e a estabilidade do serviço), os pontos fracos (como a degradação da plataforma e a baixa escalabilidade em relação à carga de trabalho) e os *trade-offs*.

- 2) **Análise Comparativa de Custo:** Comparar os valores de TCO mensal calculados, decompondo o custo para identificar os principais vetores de diferença financeira.
- 3) **Interpretação e Correlação com a Arquiteta:** Contextualizar as disparidades de desempenho e custo observadas, correlacionando-as com cada plataforma, com base na literatura.

O artefato gerado por esta atividade consiste em uma análise comparativa que detalha os resultados de desempenho e o custo mensal de cada plataforma de nuvem privada.

4.9 Realizar a Tomada de Decisão com o Método AHP

O Processo de Análise Hierárquica (AHP) é um método de apoio à tomada de decisão multicritério desenvolvido para estruturar e solucionar problemas complexos onde a escolha depende da ponderação de múltiplos fatores, muitas vezes conflitantes. Segundo Saaty (1990), essa abordagem fundamenta-se na decomposição do problema em uma estrutura hierárquica que compreende o objetivo global, os critérios de avaliação e as alternativas disponíveis. O método utiliza comparações paritárias para capturar tanto dados objetivos quanto o julgamento de valor do decisor, convertendo essas avaliações em pesos numéricos (vetores de prioridade) para sintetizar um *ranking* final que justifica matematicamente a escolha da melhor alternativa.

O objetivo da atividade final é aplicar o método AHP para sintetizar todos os critérios de desempenho e custo em um único score, ponderando-os de acordo com as prioridades da pesquisa para selecionar a plataforma que teve os melhores resultados. Esta atividade envolve quatro subatividades:

- 1) **Estruturação Hierárquica do Problema:** Decompor o problema em objetivo, critérios e alternativas.
- 2) **Comparações Paritárias e Definição de Prioridades:** Ponderar a importância relativa de cada critério usando a Escala de Saaty (SAATY, 1990a).

- 3) **Avaliação das Alternativas e Síntese Final:** Normalizar os dados de desempenho e custo e combiná-los com os pesos dos critérios para gerar um escore final.
- 4) **Análise de Consistência:** Validar a coerência lógica dos julgamentos por meio do cálculo da Razão de Consistência (CR).

Esta atividade terá como resultado a seleção da plataforma de nuvem privada que apresentou os melhores resultados de acordo com os critérios, validada pelos artefatos: matriz de comparação, vetor de prioridades e pontuação normalizada.

4.10 Considerações Finais

A metodologia proposta integra o Planejamento de Experimentos (DOE), o modelo de Custo Total de Propriedade (TCO) e o Processo de Análise Hierárquica (AHP) para auxiliar a tomada de decisão na escolha de plataformas de nuvem privada.

O modelo TCO permite a tradução dos dados empíricos em um impacto financeiro real, considerando não apenas o consumo de energia, mas também os custos de hardware e de qualificação da equipe técnica. O AHP oferece um mecanismo estruturado para sintetizar esses múltiplos critérios, permitindo uma tomada de decisão de plataformas de nuvens privadas.

5 Estudo de Caso

5.1 Introdução

Este capítulo apresenta um estudo de caso para avaliação da metodologia proposta. O objetivo do estudo é apresentar uma análise comparativa das plataformas de nuvem OpenStack e CloudStack, considerando as métricas de desempenho e custo. Os resultados das métricas de tempo de execução das VMs, utilização de processador, utilização de memória, tempo de resposta, latência, *throughput*, taxa de erro e custo total de propriedade são adotados para a tomada de decisão na seleção dessas plataformas com base na metodologia proposta.

5.2 Configuração dos Ambientes de Nuvem com as Plataformas OpenStack e CloudStack

Nessa atividade, o estudo implementou e configurou ambientes de nuvem privada com as plataformas OpenStack e CloudStack em hardwares idênticos conforme Figura 5. Foram utilizados nesse estudo dois computadores, um roteador, um switch e um nobreak.

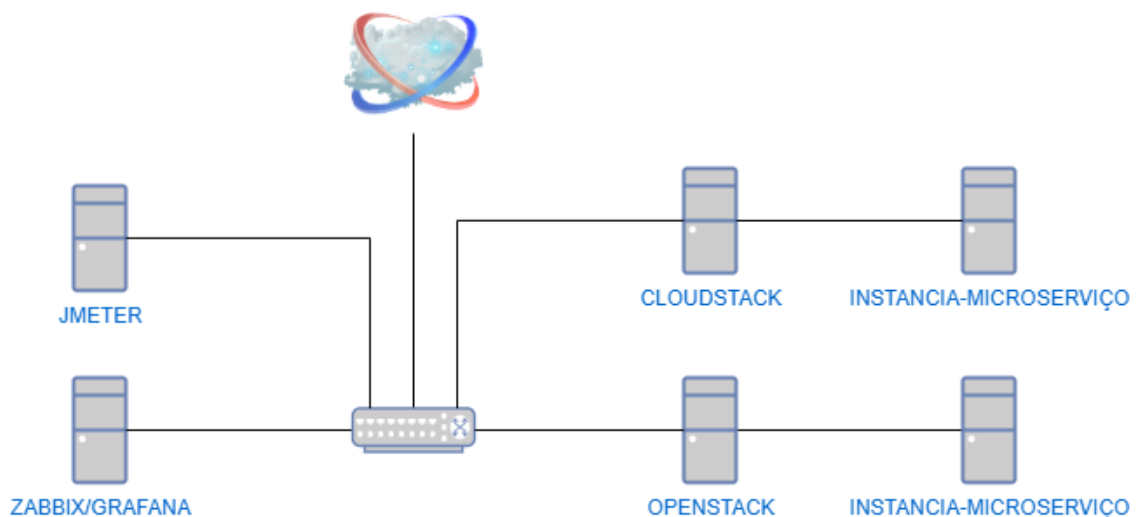


Figura 5 – Topologia

Fonte: O autor

Os servidores seguem as características detalhadas na Tabela 6.

Tabela 6 – Características do servidor OpenStack/Cloudstack

Recurso	Características das Máquinas Físicas
Processador	Intel Core I7
Memória	16GB
Disco	240GB
Sistema Operacional	Ubuntu 24.04.2 LTS

Fonte: O autor

Para os testes, foram definidas três ofertas de serviço para as máquinas virtuais (VMs) na nuvem privada, conforme a Tabela 7.

Tabela 7 – Oferta de Serviço da Nuvem Privada

Oferta de serviço	Configuração das VMs
Small	CPU: 2 Cores; Memória: 4GB; HD: 60GB
Medium	CPU: 4 Cores; Memória: 6GB; HD: 60GB
large	CPU: 8 Cores; Memória: 8GB; HD: 60GB

Fonte: O autor

5.3 Implementação do Microserviço para a Execução dos Testes de Carga

A carga de trabalho foi representada por um microserviço desenvolvido para esta pesquisa. Esta aplicação foi implantada em máquinas virtuais idênticas em ambos os ambientes, OpenStack e CloudStack, funcionando como o *endpoint* que recebia e processava as requisições HTTP geradas pela ferramenta de teste.

O microserviço implementa uma *API RESTful* para o gerenciamento de uma biblioteca. Sua função é gerenciar duas entidades de dados: Livros e Categorias, expondo *endpoints* para operações de criação (POST) e consulta (GET). A aplicação foi desenvolvida para gerar uma carga de trabalho assíncrona e intensiva em operações de I/O (Entrada/Saída).

A aplicação foi construída com os seguintes componentes:

- Linguagem de Programação: Python 3.10.
- *Framework Web*: *FastAPI*, que opera de forma assíncrona sob o padrão *ASGI* (*Asynchronous Server Gateway Interface*).
- Servidor de Aplicação: Gunicorn como gerenciador de processos, utilizando *workers Uvicorn* para executar a aplicação *ASGI*.
- Mapeamento Objeto-Relacional (ORM): *SQLAlchemy* em seu modo de operação assíncrono para interagir com o banco de dados.

O modelo de dados consiste em duas tabelas com um relacionamento de um-para-muitos:

- 1) **Categorias:** Armazena o id e nome de cada categoria.
- 2) **Livros:** Armazena atributos como id, título, autor, ano, isbn e uma chave estrangeira (*categoria_id*) que a associa a uma Categoria.

Os *endpoints* da API geram os seguintes tipos de carga de trabalho:

- **Carga de Escrita:** As requisições POST */livros* e POST */categorias*
- **Carga de Leitura:** As requisições GET */livros* e GET */categorias*

5.4 Planejamento dos Experimentos por Meio da Metodologia DOE

O planejamento da avaliação foi baseado no Planejamento de Experimentos (DOE), uma metodologia estabelecida para a caracterização de desempenho em sistemas computacionais complexos, incluindo ambientes de nuvem (ALI, 2021). Neste trabalho, foi empregado um Planejamento Fatorial Completo, onde todas as combinações dos níveis dos fatores selecionados são sistematicamente testadas.

A abordagem permitiu variar sistematicamente os fatores Oferta de Serviço e Intensidade de Carga (Tabela 8).

Tabela 8 – Fatores e níveis do planejamento de experimentos

Fatores	Níveis		
Ofertas de Serviço	Small	Medium	Large
Intensidade de carga	500	700	900

Fonte: O autor

A combinação desses fatores gerou nove cenários de teste para cada plataforma, (Tabela 9).

Tabela 9 – Cenários gerados pelo planejamento de experimentos

Cenário	Oferta de Serviço	Intensidade de carga
1	Small	500
2		700
3		900
4	Medium	500
5		700
6		900
7	Large	500
8		700
9		900

Fonte: O autor

A Tabela 9 detalha os nove cenários de teste que foram gerados pelo Planejamento de Experimentos (DOE). Cada cenário, numerado de 1 a 9, representa uma combinação única de dois fatores principais: Oferta de Serviço com os níveis Small, Medium e Large e Intensidade de carga com os níveis 500, 700 e 900. A tabela ilustra o resultado de um planejamento fatorial 3x3, onde todas as três configurações de serviço foram testadas em

conjunto com todas as três intensidades de carga, resultando em nove condições de teste distintas.

5.5 Análise de Desempenho

A análise do desempenho, envolveu a medição e a coleta das métricas para cada cenário planejado. Para isso, foram utilizadas duas de ferramentas: Apache JMeter, para geração de carga (O. GÓMEZ; G. PÉREZ; ROJAS, 2022); e o Zabbix/Grafana, para coletar as métricas de utilização de recursos diretamente do servidor. Durante os testes, foram coletadas as métricas de Utilização de Memória (%) e do Processador (%), Tempo de Resposta (s), Tempo de Execução das VMs e Latência (s), e Throughput (vazão) e Taxa de Erro (%) que representam a média aritmética dessas 30 execuções para cada métrica coletada. O experimento com o CloudStack durou 246 minutos, enquanto o OpenStack durou 236 minutos. Utilizando um intervalo de amostragem de 60 segundos, as médias das métricas de processador e memória foram coletadas pelo Zabbix/Grafana, e as de tempo de resposta, latência e throughput, pelo Apache JMeter. No total, foram realizadas 270 execuções para cada plataforma de nuvem privada, sendo 30 por oferta de serviço.

A Tabela 10 apresenta os dados de desempenho consolidados coletados para a plataforma OpenStack. Ela detalha os valores médios resultantes para as métricas de Memória (%), Processador (%), Tempo de Resposta (seg), Latência (seg) e *Throughput* (vazão). Os resultados estão organizados de acordo com os nove cenários experimentais, mostrando o desempenho para cada combinação dos fatores. O tempo de execução foram obtidos somando-se o tempo total das 30 réplicas executadas para cada um dos nove cenários (combinações de oferta de serviço e intensidade de carga), com a duração extraída dos logs “*Start Time*” da ferramenta Apache JMeter. A análise do tempo de execução é um parâmetro crucial para o modelo de Custo Total de Propriedade (TCO). Conforme detalhado na Seção 5.6, o tempo total de execução dos testes (236 minutos para OpenStack e 246 minutos para CloudStack) representa o período de observação completo sobre o qual a utilização média de *CPU* e memória foi aferida. Esses dados de utilização média dos recursos são, por sua vez, aplicados na equação da Potência Média (em Watts), sendo um

componente essencial para o cálculo final do Custo de Energia Elétrica (OPEX) de cada plataforma.

Tabela 10 – Resultado das Métricas de Desempenho - OpenStack

Oferta de Serviço	Intensidade de Carga	Memória (%)	Processador (%)	Tmp Resposta (seg)	Latência (seg)	Throughput (vazão)	Tempo de Execução (minutos)
SMALL	500	27,98	60,11	0,17	0,17	8,30	25
	700	27,84	78,19	0,68	0,68	11,58	22
	900	27,56	86,63	9,33	9,49	12,36	32
MEDIUM	500	22,87	37,44	0,16	0,20	8,30	28
	700	22,87	37,44	0,16	0,20	8,30	30
	900	22,88	57,16	0,20	0,20	15,07	27
LARGE	500	24,33	21,22	0,15	0,15	8,30	17
	700	24,20	23,90	0,15	0,15	11,60	30
	900	23,90	31,37	0,16	0,16	14,90	25

Fonte: O autor

A Tabela 11 apresenta os dados de desempenho consolidados coletados para a plataforma CloudStack. De forma similar à Tabela 10, ela detalha os valores médios resultantes para as métricas de Memória (%), Processador (%), Tempo de Resposta (seg), Latência (seg), Throughput (vazão) e Tempo de Execução.

Tabela 11 – Resultado das Métricas de Desempenho - CloudStack

Oferta de Serviço	Intensidade de Carga	Memória (%)	Processador (%)	Tmp Res-posta (seg)	Latência (seg)	Throughput (vazão)	Tempo de Execução (minutos)
SMALL	500	28,22	4,29	0,01	0,01	8,31	26
	700	26,61	5,88	0,01	0,01	11,75	29
	900	22,40	6,73	0,01	0,01	14,93	28
MEDIUM	500	14,42	2,16	0,01	0,01	8,30	28
	700	14,71	2,84	0,01	0,01	11,60	29
	900	15,00	3,48	0,01	0,01	14,90	27
LARGE	500	11,26	1,06	0,01	0,01	8,28	26
	700	12,08	1,53	0,01	0,01	11,60	29
	900	11,70	1,70	0,01	0,01	14,90	24

Fonte: O autor

5.5.1 Diferença percentual

Esta seção apresenta uma análise comparativa detalhada do desempenho das plataformas OpenStack e CloudStack, com base nos dados coletados nos experimentos das Tabelas 10 e 11. Para isso, serão apresentados e discutidos gráficos que ilustram o aumento (ou diminuição) percentual no consumo de recursos (Memória e *CPU*) e nas métricas de qualidade de serviço (Tempo de Resposta, Latência e Throughput) da plataforma OpenStack em relação à CloudStack, considerando as diferentes ofertas de serviço (Large, Medium e Small) avaliadas.

- Diferença percentual na utilização de memória

A Figura 6 ilustra o aumento percentual no consumo médio de memória RAM pela plataforma OpenStack em comparação com a plataforma CloudStack, para cada oferta de serviço (Large, Medium e Small). Observa-se que a diferença de consumo entre as plataformas é mais acentuada nas configurações Medium e Large. OpenStack utilizou 107% a mais de memória na oferta Large, 55,54% a mais na oferta Medium e 8,94% a mais na oferta Small, em relação ao consumo do CloudStack sob as mesmas condições. Estes dados indicam que, nos cenários testados, o *overhead* de memória imposto pelo OpenStack cresce significativamente com o aumento dos recursos alocados à máquina virtual.

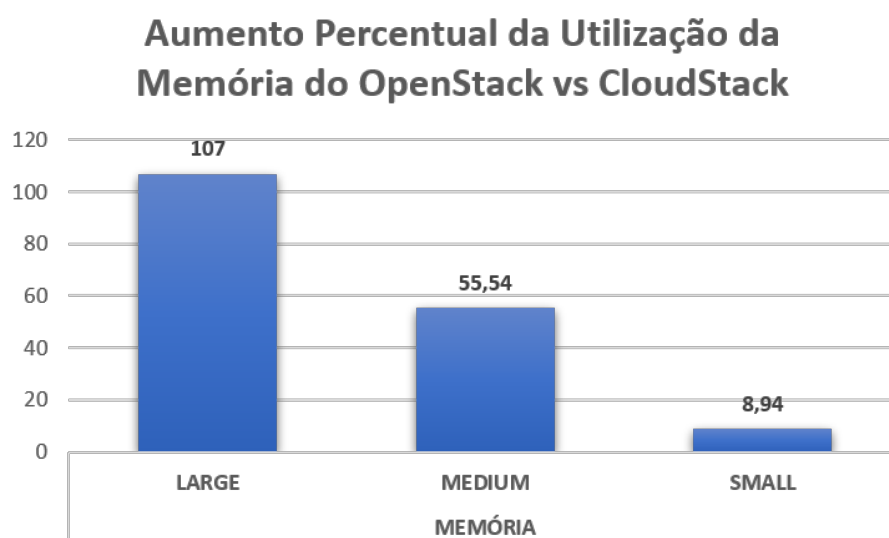


Figura 6 – Utilização Memória
Fonte: O autor

- Diferença percentual na utilização do processador

A Figura 7 demonstra o expressivo aumento percentual no consumo médio de *CPU* pela plataforma OpenStack em comparação com a CloudStack, analisado para cada oferta de serviço. O OpenStack utilizou 1682% a mais de *CPU* na oferta Large, 1456% a mais na oferta Medium e 1231% a mais na oferta Small, sempre em relação ao consumo do CloudStack nas mesmas condições. Esses resultados quantificam um *overhead* computacional maior imposto pelo OpenStack.

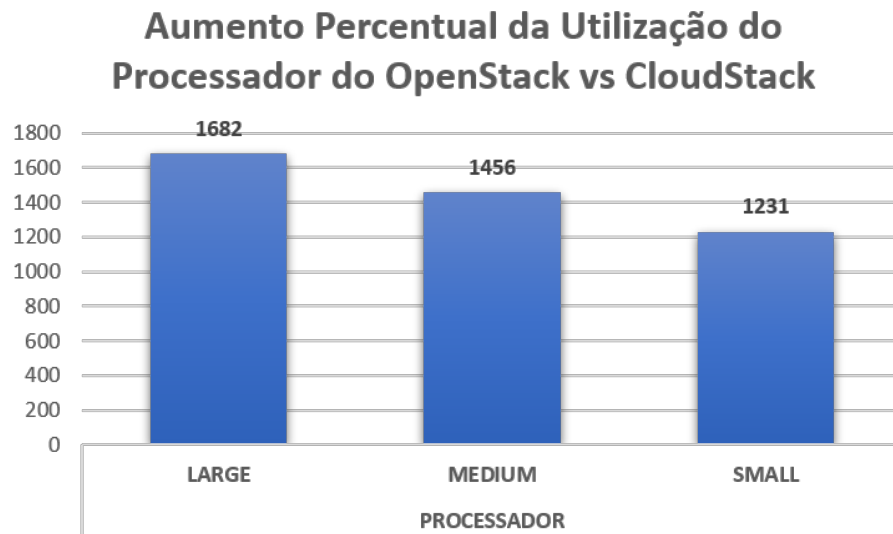


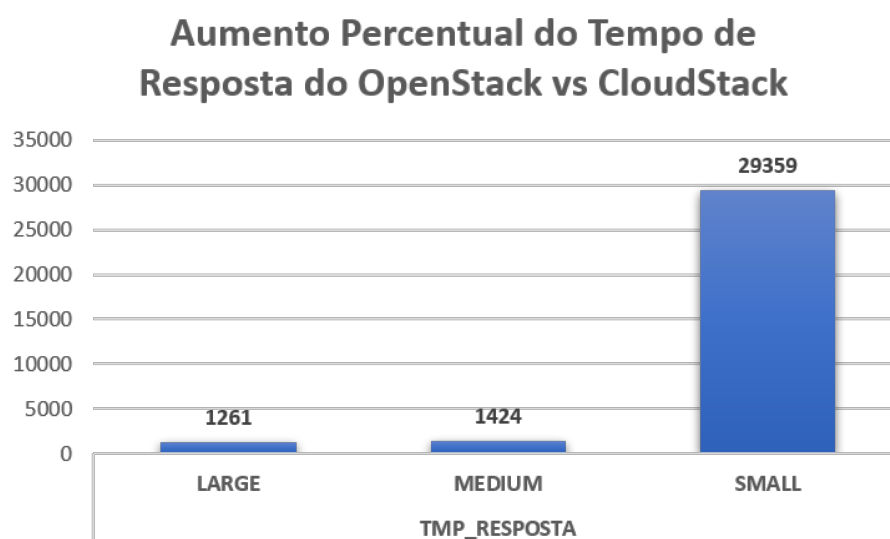
Figura 7 – Utilização Processador

Fonte: O autor

Este elevado consumo de *CPU* pelo OpenStack encontra respaldo na literatura, que o associa à sua filosofia arquitetônica modular. Sendo um conjunto de múltiplos projetos e serviços independentes (ex: Nova, Neutron, Keystone) que se comunicam via APIs, essa abordagem distribuída, embora flexível, apresenta um maior *overhead* sistêmico, pois cada serviço consome ciclos de *CPU* e memória para sua própria execução e para a comunicação constante entre eles (PATEL; PATEL, 2016). Adicionalmente, seu subsistema de rede, o Neutron, é notoriamente complexo e, ao utilizar componentes como o Open vSwitch (OVS), impõe uma carga de processamento de pacotes significativa sobre a *CPU* do *host*, o que se reflete diretamente na maior utilização do processador observada nos experimentos (CHAUDHRY, 2015).

- Diferença percentual no tempo de resposta

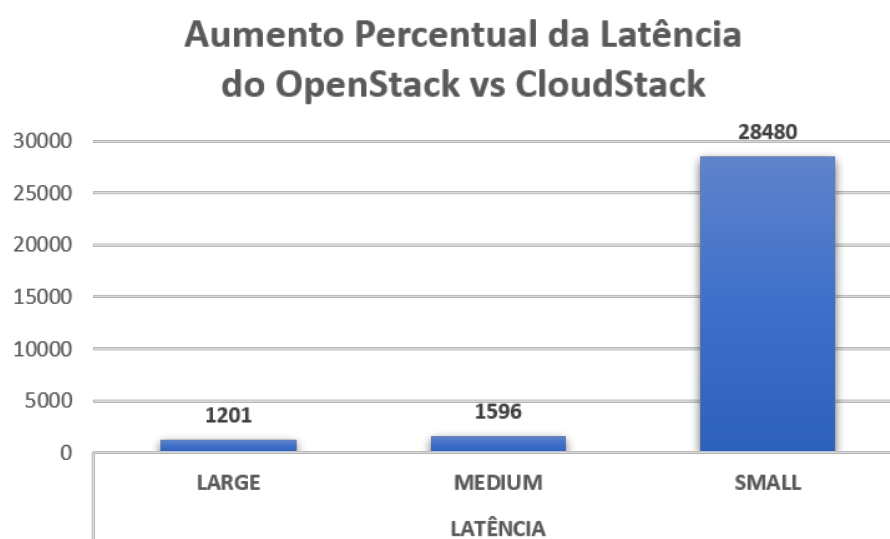
A Figura 8 apresenta o aumento percentual no tempo médio de resposta observado na plataforma OpenStack em relação à CloudStack, para cada oferta de serviço. Na oferta Large, o tempo de resposta do OpenStack foi 1261% maior em relação ao CloudStack. Na oferta Medium, essa diferença aumentou para 1424% (mais de 14 vezes superior). Na oferta de serviço Small, onde o tempo de resposta do OpenStack foi 29359% maior.

**Figura 8 – Tempo de Resposta**

Fonte: O autor

- Diferença percentual da latência

A Figura 9 mostra o aumento percentual na latência média observada na plataforma OpenStack em comparação com a CloudStack, para cada oferta de serviço. Na oferta Large, a latência do OpenStack foi 1201% maior que a do CloudStack, enquanto na oferta Medium essa diferença subiu para 1596%. Na oferta Small, a latência do OpenStack foi 28480% superior à do CloudStack.

**Figura 9 – Latência**

Fonte: O autor

- Diferença percentual do throughput

A Figura 10 apresenta a diferença percentual na vazão média (throughput) alcançada pelo OpenStack em comparação com o CloudStack, para cada oferta de serviço. Na oferta Large, o desempenho foi praticamente idêntico, com o OpenStack registrando um throughput apenas 0,07% superior ao do CloudStack. No entanto, na oferta Medium, sua vazão foi 9,02% menor, e na oferta Small, foi 7,88% menor que a do CloudStack. Estes resultados sugerem que, embora a capacidade máxima de processamento sob carga elevada seja similar entre as plataformas em configurações robustas, o CloudStack manteve uma maior eficiência de vazão nas configurações com menos recursos no cenário testado.

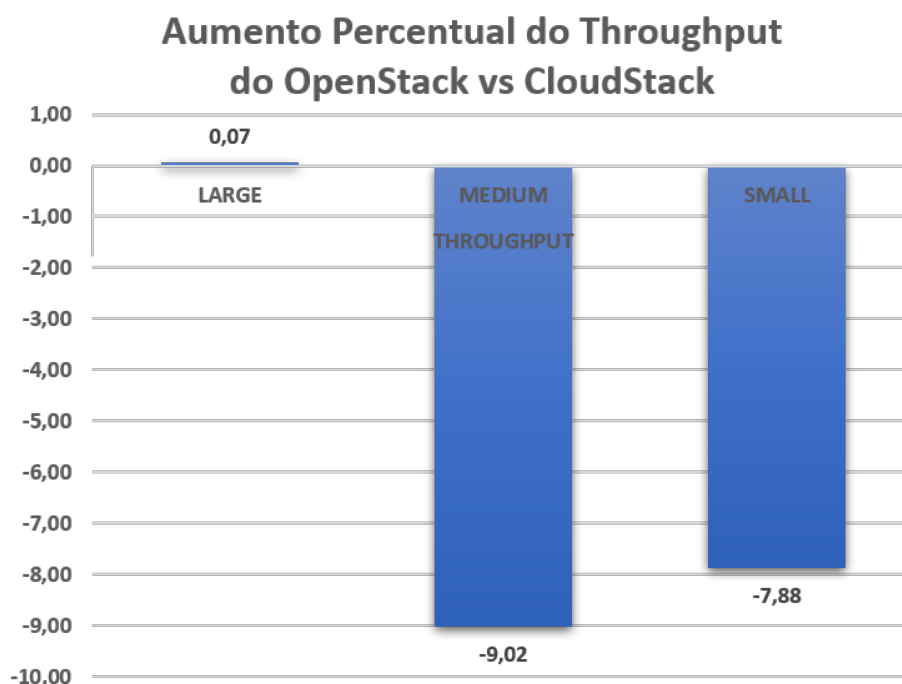


Figura 10 – Throughput
Fonte: O autor

5.5.2 Análise dos Efeitos dos Fatores sobre as Métricas de Desempenho

Nesta seção, é realizada a análise dos efeitos dos fatores carga de trabalho e oferta de serviços nas métricas de desempenho coletadas nas plataformas de Nuvem OpenStack e CloudStack, conforme as Tabelas 10 e 11. A avaliação abrange os indicadores de Utilização de Memória (%), Utilização do Processador (%), Tempo de Resposta (s), Latência (s)

e Throughput (vazão). Para cada métrica, a análise compara o comportamento das plataformas CloudStack e OpenStack, utilizando gráficos de efeitos principais e dados consolidados para avaliar como as variações nos fatores do experimento, Oferta de Serviço e Intensidade de Carga, impactaram os resultados. O objetivo é compreender os padrões de resposta de cada ambiente sob as condições controladas do estudo.

- Utilização de Memória

Avaliando a utilização média de memória (Figura 11), a análise dos gráficos demonstra que o fator Oferta de Serviço provocou a variação mais significativa no consumo para ambas as plataformas, enquanto a Intensidade de Carga teve um impacto comparativamente mínimo. No CloudStack, observou-se um comportamento em que a utilização média de memória diminuiu com o aumento da oferta de serviço, caindo de um consumo de aproximadamente 26% (na oferta Small) para cerca de 11,5% (na oferta Large). Em contraste, o OpenStack apresentou um comportamento não linear. O consumo foi maior na oferta Small (cerca de 28%) e atingiu seu ponto mais baixo na oferta Medium (aproximadamente 22,8%), antes de subir ligeiramente na oferta Large (para 24,2%).

No comparativo, o CloudStack demonstrou um uso de memória mais otimizado e previsível, especialmente nas ofertas Medium e Large, onde seu consumo médio foi significativamente menor que o do OpenStack, indicando um menor *overhead* de memória nessas configurações maiores (Medium e Large).

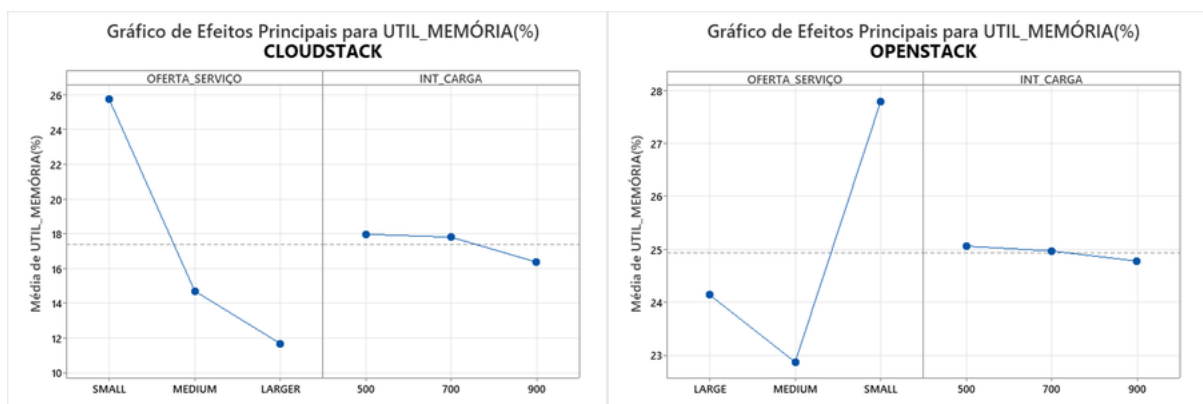


Figura 11 – Memória
Fonte: O autor

- Utilização do Processador

Avaliando a utilização média do processador Figura 12, a análise dos gráficos de efeitos principais demonstra que o fator Oferta de Serviço foi o que provocou a variação mais significativa no consumo de *CPU* para ambas as plataformas, enquanto a Intensidade de Carga teve um impacto comparativamente menor. No CloudStack, observou-se um comportamento na utilização média de *CPU* em que diminuiu consistentemente com o aumento da oferta, caindo de um consumo de aproximadamente 5,6% (na oferta Small) para seu ponto mais baixo de 1,4% (na oferta Large). O OpenStack apresentou um padrão de resposta idêntico, mas em uma escala superior. O consumo também diminuiu com o aumento da oferta, caindo de um consumo médio de 78% (na oferta Small) para cerca de 25% (na oferta Large).

No comparativo geral, CloudStack demonstrou um melhor desempenho. Sua utilização média de *CPU* permaneceu extremamente baixa em todos os cenários (média geral de 3,3%), enquanto OpenStack operou com um consumo muito mais elevado (média geral de 48%). Isso indica que CloudStack exigiu menos recursos de processamento para executar as mesmas cargas de trabalho nas condições testadas.

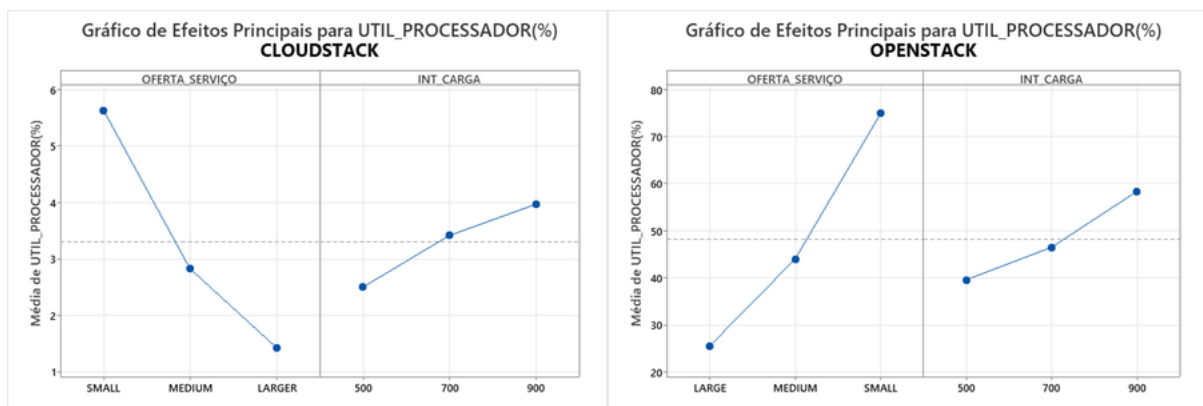


Figura 12 – Processador

Fonte: O autor

- Tempo de Resposta

Ao analisar o tempo médio de resposta (Figura 13), é observado impacto distinto entre as plataformas em resposta às variações nos testes. No CloudStack, tanto a intensidade de carga quanto a oferta de serviço tiveram um impacto mínimo: a mudança na carga de 500 para 900 requisições resultou em uma variação de apenas 0,1 milissegundos

(redução), enquanto a maior variação entre as ofertas de serviço (Small vs Medium) foi de aproximadamente 0,2 milissegundos. Em contrapartida, o OpenStack na intensidade de carga (500 para 900 requisições), ocorreu um aumento no tempo médio de resposta em 3,07 segundos, e a mudança da oferta de serviço Large para Small causou um aumento de 3,25 segundos no tempo de resposta. O CloudStack apresentou tempo de resposta de 11 milissegundos, enquanto o OpenStack apresentou tempo médio de 1,2 segundos, com picos que ultrapassaram 3 segundos nas ofertas de serviço Medium e Large. Isso indica que o CloudStack apresentou tempos de resposta menores nos cenários avaliados.

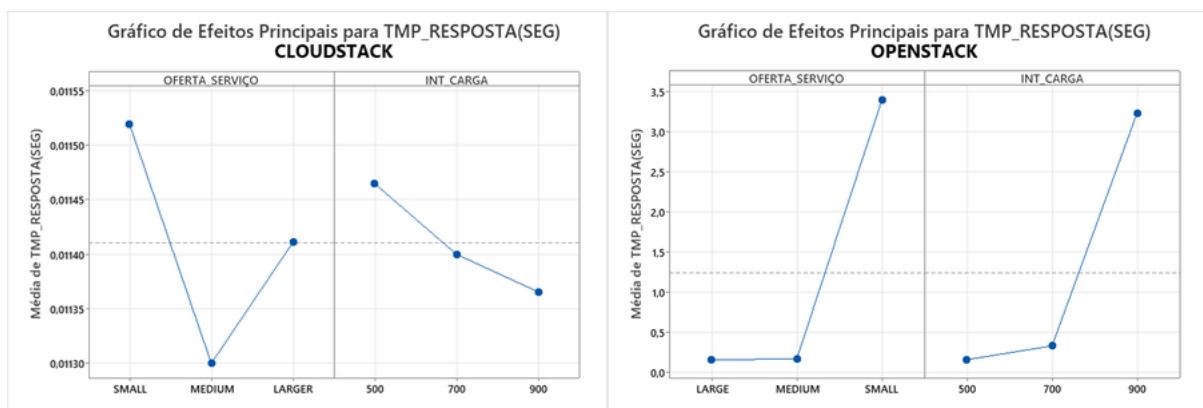


Figura 13 – Tempo de Resposta

Fonte: O autor

- Latência

Ao observar a (Figura 14). O CloudStack apresentou uma redução de latência na oferta de serviço entre Small e Medium. No OpenStack a latência apresentou maior efeito dos fatores oferta de serviço e intensidade de carga de trabalho. O aumento na intensidade de carga de 500 para 900 requisições elevou a latência em 3,12 segundos. A variação em resposta à oferta de serviço ocorreu com a mudança da oferta Small para Large resultando em uma redução de 3,30 segundos. O CloudStack teve latência de aproximadamente 11,9 milissegundos, enquanto o OpenStack apresentou latências médias de 1,28 segundos, com picos que ultrapassaram 3 segundos.

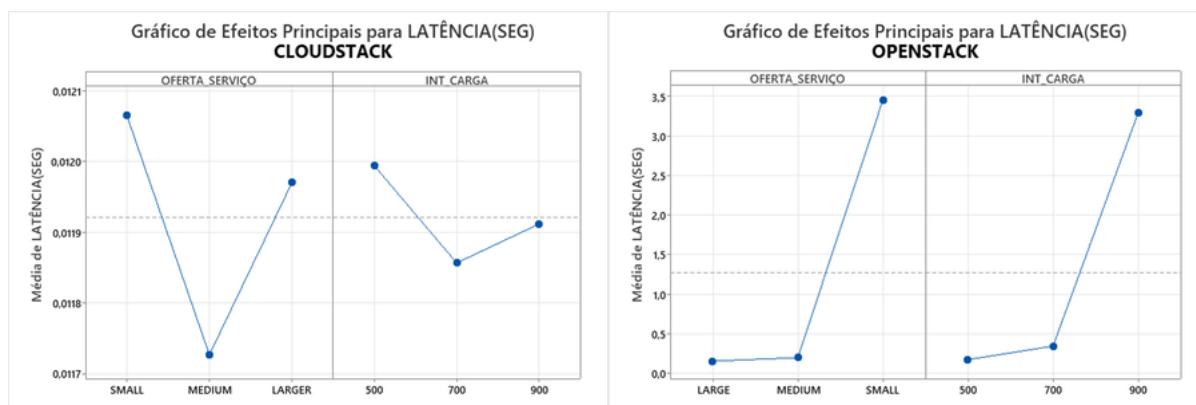


Figura 14 – Latência

Fonte: O autor

- Throughput

Ao analisar (Figura 15) referente a plataforma CloudStack, a intensidade de carga teve um impacto linear com um aumento da vazão de 6,6 requisições por segundo ao passar de 500 para 900 requisições simultâneas. Em contraste, a oferta de serviço demonstrou ter um impacto praticamente nulo, com a vazão média permanecendo em torno de 11,6 req/s. O OpenStack apresentou um comportamento muito similar em relação à carga, com um aumento de aproximadamente 5,8 req/s ao variar a intensidade de 500 para 900 requisições. Contudo, a oferta de serviço apresentou uma variação de 1,1 req/s entre as ofertas de serviço Large e Medium.

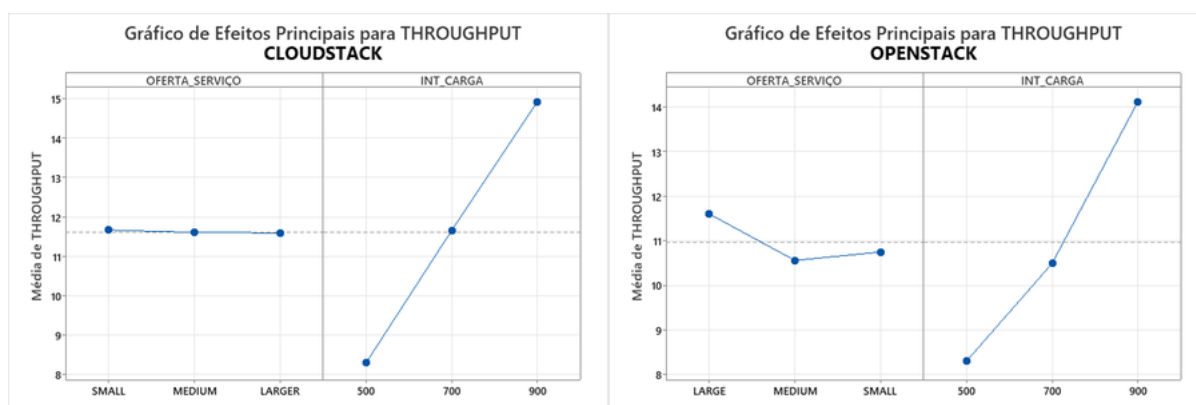


Figura 15 – Throughput

Fonte: O autor

5.6 Análise de Custo

Conforme a metodologia proposta (Seção 4.6), a avaliação de custo foi delineada para quantificar monetariamente a eficiência operacional e a complexidade de gerenciamento de cada plataforma.

É importante ressaltar que o 'Fator Complexidade' utilizado no cálculo do custo de pessoal não é um multiplicador arbitrário, mas sim a tradução direta da exigência técnica de cada plataforma em níveis de senioridade profissional. A literatura estabelece que a arquitetura modular e distribuída do OpenStack demanda um conhecimento aprofundado de múltiplos componentes independentes, exigindo, portanto, um profissional de nível Sênior para garantir sua operabilidade e manutenção (GUPTA; SHARMA, 2023). Em contrapartida, a abordagem integrada (*turnkey*) do CloudStack reduz significativamente a carga de gerenciamento, permitindo que a infraestrutura seja mantida com eficiência por um profissional de nível Pleno (S. FILHO; G. FILHO, 2016). Essa distinção de perfis, fundamentada na complexidade arquitetural intrínseca de cada solução, é a responsável pela diferença base nos custos de pessoal apresentada no modelo.

Para a construção do modelo, foram definidos os parâmetros da Tabela 12. Com base nesses parâmetros e nos dados de desempenho, o TCO mensal para cada plataforma foi calculado, conforme detalhado na Tabela 13.

Tabela 12 – Parâmetros para o Modelo de Custo Total de Propriedade (TCO)

Catego- ria	Parâmetro	Valor	Justificativa/Fonte
Hardware (CAPEX)	Custo de Arquisição do Servidor	R\$ 8.000,00	Estimativa de mercado para um servidor de entrada com as especificações do estudo. (DELL, 2025)
	Período de Amortização	36 Meses	Vida útil padrão de 3 anos para amortização de ativos de TI
Pessoal (OPEX)	Nível de Qualificação (CloudStack)	Pleno	Plataforma com gerenciamento integrado (S. FILHO; G. FILHO, 2016)
	Salário Mensal (Pleno)	R\$ 7.000,00	Média salarial para um profissional de nível pleno na área (Guias salariais 2025)
	Nível de Qualificação (OpenStack)	Sênior	Plataforma com maior complexidade operacional (GUPTA; SHARMA, 2023)
	Salário Mensal (Sênior)	R\$ 10.500,00	Média salarial para um profissional de nível sênior na área (Guias salariais 2025)
Energia (OPEX)	Potência Máxima do Servidor (P_max)	500 W	Valor conservador para um servidor com as características descritas operando sob carga (BARROSO; HOLZLE, 2018)
	Potência Ociosa do Servidor (P_idle)	100 W	Estimativa de 20% da potência máxima para consumo em ociosidade.
	Tarifa de Energia (Pernambuco)	R\$ 0,85 / kWh	Tarifa média comercial para a região da UFRPE. ((ANEEL), 2025)
	Power Usage Effectiveness (PUE)	1.6	Média de Eficiência para datacenters tradicionais (KOOMEY, 2011)

Fonte: O autor

Com base nesses parâmetros e nos dados de desempenho, as equações de Custo de Hardware (Equação 4.1), Custo da Equipe Técnica (Equação 4.2) e Custo de Energia Elétrica (Equação 4.3) foram aplicadas para calcular os valores monetários apresentados. O Custo Total de Propriedade foi consolidado através da (Equação 4.4), conforme detalhado na Tabela 13.

Tabela 13 – Cálculo do Custo Total de Propriedade (TCO) Mensal

Modelos de custo	Fórmula / Lógica	OpenStack	CloudStack
1 - Custo de Hardware (Amortizado)	Custo Servidor / 36	R\$ 222,22	R\$ 222,22
2 - Custo da Equipe Técnica	Salário Base do Nível Definido (Tabela 12)	R\$ 10.500,00	R\$ 7.00,00
3 - Custo de Energia	(Energia kWh/mês) x Tarifa X PUE	R\$ 246,31	R\$ 145,21
Custo Total Mensal (TCO)	Equação 4.4	R\$ 10.968,53	R\$ 7.367,43

Fonte: O autor

O Custo de Energia considerou 720 horas/mês com base na potência média aferida nos experimentos. As utilizações de *CPU* e memória medidas nos experimentos durante o período total de testes (236 minutos para OpenStack e 246 minutos para CloudStack), são utilizados para calcular a Potência Média (em Watts) de cada plataforma.

É fundamental esclarecer que a divergência no custo de energia é um reflexo direto da eficiência computacional de cada plataforma. Uma vez que os parâmetros de hardware ($P_{\text{máxima}}$) e (P_{ociosa}) e a tarifa de energia são constantes para ambos os cenários, a única variável que altera o resultado da equação de consumo é a Utilização Média de Recursos (U_{recursos}). Conforme evidenciado na análise de desempenho, o OpenStack demandou taxas de utilização de processador e memória consistentemente maiores para processar a mesma carga de trabalho. Esse maior 'esforço computacional' eleva o fator (U_{recursos}) na fórmula da potência média, resultando matematicamente em um consumo de Watts superior e, conseqüentemente, em um custo mensal de energia mais elevado.

5.7 Análise dos Cenários de Desempenho e Custo para Cada Plataforma

A análise do TCO (Tabela 13) revela que o custo total mensal para operar o CloudStack (R\$ 7.367,43) é aproximadamente 33% menor que o custo para operar o OpenStack (R\$ 10.968,53). Embora a maior parte da diferença seja impulsionada pelo custo de pessoal associado à maior complexidade do OpenStack, o custo de energia também é significativamente maior (cerca de 70% a mais).

Os dados empíricos demonstram que o CloudStack mantém latência e tempo de resposta invariavelmente baixos, com tempos em milissegundos. Em contrapartida, a latência no OpenStack varia de forma acentuada em resposta a carga de trabalho adotada. Essa disparidade de desempenho encontra respaldo na literatura acadêmica, que a atribui às arquiteturas das duas plataformas a abordagem distribuída do OpenStack, embora flexível, apresenta um maior *overhead* sistêmico em comparação com a arquitetura do CloudStack (PATEL; PATEL, 2016).

5.8 Realização da Tomada de Decisão com o Método AHP

Com o Custo Total de Propriedade (TCO) de cada plataforma calculado e os dados de desempenho aferidos, o Processo de Análise Hierárquica (AHP) foi aplicado para integrar todos os critérios e selecionar a plataforma de nuvem privada com os melhores resultados de desempenho e custo. O AHP é particularmente eficaz em cenários como este, pois decompõe o problema em uma estrutura lógica (SAATY, 1990a). Conforme apresentado na Figura 16, definidos o objetivo, os critérios e as alternativas, a aplicação do método torna possível encontrar a alternativa com melhores dados, desempenho e custo.

A análise a seguir foi estruturada para responder à questão central da pesquisa: **Qual plataforma de gerenciamento (OpenStack ou CloudStack) apresenta o melhor desempenho geral, considerando os critérios de maior importância (custo, memória e processador) ?**

5.8.1 Estruturação do problema de decisão

A primeira etapa do método AHP consiste em estruturar o problema de forma hierárquica, decompondo-o em objetivo, critérios e alternativas.

5.8.1.1 Hierarquia do Problema e Ponderação dos Critérios

A hierarquia para esta decisão é definida da seguinte forma:

- **Objetivo:** Selecionar a Plataforma de Gerenciamento de Nuvem que apresentou melhores resultados.
- **Critérios:** Custo Total de Propriedade (TCO), Utilização de *CPU* (*CPU%*), Utilização de Memória (*MEM%*), Latência (*LAT*), Tempo de Resposta (*TMR*) e Throughput (*THR*).
- **Alternativas:** OpenStack e CloudStack.

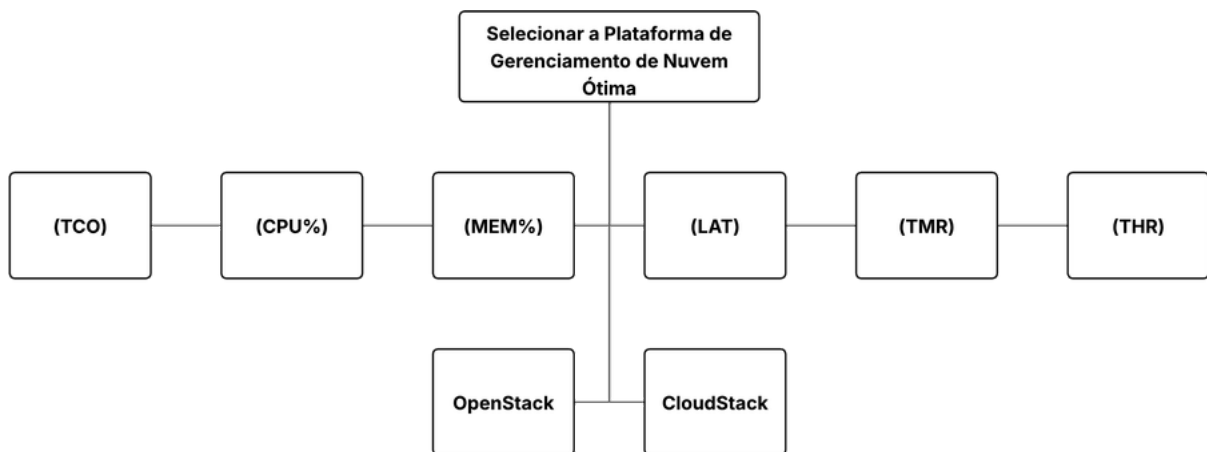


Figura 16 – Hierarquia do Problema
Fonte: O autor

A comparação entre os critérios é realizada utilizando a Escala Fundamental de Saaty, como podemos visualizar na Tabela 14. Segundo o autor, essa escala reflete a capacidade humana de discriminar entre diferentes intensidades de preferência (SAATY, 2008).

Tabela 14 – Escala Fundamental de Saaty para Comparações Paritárias

Intensidade de Importância	Definição
1	Igual importância
3	Moderada importância de um sobre o outro
5	Forte importância
7	Importância muito forte
9	Extrema importância
2, 4, 6, 8	Valores intermediários

Fonte: Adaptado de Saaty (2008)

Com base na premissa da pesquisa, que prioriza o desempenho dos recursos e a viabilidade econômica, os critérios relacionados ao consumo (Custo, *CPU%*, *MEM%*) foram julgados como tendo “importância muito forte” (valor 7) sobre os de qualidade de serviço (*LAT*, *TMR*, *THR*). A Tabela 17 apresenta a matriz de comparação paritária resultante. Por exemplo, a célula que compara Custo com Latência é preenchida com o valor 7. Consequentemente, a célula recíproca (Latência vs. Custo) recebe o valor 1/7, um axioma fundamental do método que garante a coerência dos julgamentos (SAATY, 1990a).

Tabela 15 – Matriz de Comparação Paritária para os Critérios

Critério	CUSTO	CPU%	MEM%	LAT	TMR	THR
CUSTO	1.00	1.00	1.00	7.00	7.00	7.00
CPU%	1.00	1.00	1.00	7.00	7.00	7.00
MEM%	1.00	1.00	1.00	7.00	7.00	7.00
LAT	0.14	0.14	0.14	1.00	7.00	7.00
TMR	0.14	0.14	0.14	1.00	1.00	7.00
THR	0.14	0.14	0.14	1.00	1.00	1.00
Soma	4.42	4.42	4.42	24.00	33.00	36.00

Fonte: O autor

Para extrair as prioridades (pesos), a matriz é primeiramente normalizada. O processo consiste em dividir cada elemento (a_{ij}) da matriz pela soma de sua respectiva coluna ($x_{ij} = \frac{a_{ij}}{\sum_{i=1}^n a_{ij}}$), conforme a fórmula, citada em (SAATY, 1990a):

Onde:

- x_{ij} é o valor normalizado do elemento na linha i e coluna j ;
- a_{ij} é o valor original do julgamento na matriz de comparação (na linha i e coluna j);
- $\sum_{i=1}^n a_{ij}$ representa a soma de todos os elementos da coluna j , onde n é o número total de critérios ou itens que estão sendo comparados.

O peso final de cada critério (w_i) é a média aritmética dos valores de sua linha na matriz já normalizada.

Exemplo de cálculo para o peso do critério CUSTO:

Para isso, somam-se os valores normalizados da linha “CUSTO” e divide-se pelo número de critérios (6):

- $w_{\text{CUSTO}} = (0.226 + 0.226 + 0.226 + 0.292 + 0.233 + 0.194) / 6 = 1.397 / 6 = 0.2328$ ou 23.3%

Como resultado, é criada a tabela de prioridades com os pesos dos critérios (Tabela 16), que agora reflete estes novos valores.

Tabela 16 – Vetor de Prioridades com Pesos dos Critérios

Critério	Peso (Prioridade)
CUSTO (TCO)	23.3%
CPU%	23.3%
MEM%	23.3%
LAT	9.4%
TMR	6.1%
THR	3.3%

Fonte: O autor

Análise de Consistência: Para garantir a validade dos julgamentos, o AHP provê um mecanismo para medir a consistência. O autovetor principal, $\lambda_{\max}$, é uma métrica crucial que, em uma matriz perfeitamente consistente, é igual ao número de itens comparados (n). A magnitude de seu desvio de n é a base para medir a inconsistência do julgamento. Segundo (SAATY, 2008), um grau de inconsistência é esperado no julgamento humano, mas ele não deve exceder um limite razoável, medido pela Razão de Consistência (CR):

- **1 - Cálculo do Autovetor Principal ($\lambda_{\max}$):** $\lambda_{\max} = 6.25$
- **2 - Cálculo do Índice de Consistência (CI):** $CI = (\lambda_{\max} - n) / (n - 1)$
 $= (6.25 - 6) / (6 - 1) = 0.05$
- **3 - Cálculo da Razão de Consistência (CR):** $CR = CI / RI$. O RI (Índice de Aleatoriedade) é um valor tabelado por Saaty, que para $n = 6$ é 1,24 (Tabela 17). Logo, $CR = 0.05 / 1.24 = 0.04$. Um $CR \leq 0,10$ é considerado aceitável (SAATY, 2008), portanto os julgamentos são perfeitamente consistentes.

Tabela 17 – Índice de Aleatoriedade (RI)

n	1	2	3	4	5	6	7	8	9	10
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Fonte: Adaptado de Saaty (2008)

5.8.1.2 Avaliação das Alternativas e Síntese Final

As alternativas (OpenStack e CloudStack) foram avaliadas em relação a cada critério. As pontuações normalizadas (Tabela 18), que medem o desempenho relativo de cada plataforma, não se alteram, pois dependem apenas dos dados de desempenho e custo, não dos pesos.

A Tabela 20 é gerada a partir dos dados de desempenho e custo coletados nas etapas anteriores da pesquisa. Os valores de entrada para este cálculo são os resultados

médios de desempenho de OpenStack (Tabela 11) e CloudStack (Tabela 12) , juntamente com os valores finais do Custo Total de Propriedade (TCO) calculados na Tabela 15.

Tabela 18 – Pontuação Normalizada das Plataformas por Critério

Critério	Pontuação OpenStack	Pontuação CloudStack
CUSTO (TCO)	40.0%	60.0%
CPU (%)	6.4%	93.6%
MEM (%)	40.4%	59.6%
LAT	0.8%	99.2%
TMR	0.8%	99.2%
THR	47.6%	52.4%

Fonte: O autor

A etapa final é a síntese, onde os pesos dos critérios (Tabela 16) são combinados com as pontuações normalizadas das plataformas (Tabela 18) para gerar o resultado final (Tabela 19).

Essa combinação é um cálculo de **soma ponderada**. Para determinar a pontuação final de cada plataforma, o valor da coluna “Peso” de um critério (extraído da Tabela 16) é multiplicado pela “Pontuação” da plataforma (extraída da Tabela 18) para esse mesmo critério.

Por exemplo, a contribuição do critério “*CPU (%)*” para a pontuação final do OpenStack é o resultado de **23.3%** (Peso da Tabela 16) multiplicado por **6.4%** (Pontuação do OpenStack na Tabela 18). Esse cálculo é repetido para todos os seis critérios.

O “SOMA (Resultado Final)” para cada plataforma, visto na Tabela 19, é o somatório desses seis resultados parciais, o que resulta em **22.2%** para OpenStack e **77.8%** para CloudStack

Tabela 19 – Síntese Final para a Escolha da Plataforma

Critério	Peso	Pontuação OpenStack	Pontuação CloudStack
CUSTO (TCO)	23.3%	0.400	0.600
CPU (%)	23.3%	0.064	0.936
MEM (%)	23.3%	0.404	0.596
LAT	9.4%	0.008	0.992
TMR	6.1%	0.008	0.992
THR	3.3%	0.476	0.524
SOMA (Resultado Final)	100%	22.2%	77.8%

Fonte: O autor

O método AHP resultou em uma pontuação final de 77,8% para o CloudStack, consolidando sua posição como a plataforma de gerenciamento com melhores resultados de acordo com os critérios em relação ao OpenStack (22,2%) neste experimento.

5.9 Considerações Finais

A aplicação do método AHP permitiu responder de forma estruturada e justificada à questão central da pesquisa. O resultado final, que atribuiu ao CloudStack uma pontuação de preferência de 77,8% contra 22,2% do OpenStack, não é apenas um *ranking*, mas a validação metodológica de que, para um cenário que prioriza a economicidade de recursos e a previsibilidade, o CloudStack no contexto do experimento demonstrou melhor desempenho.

A análise quantifica a preferência pela plataforma, que não só apresentou melhor desempenho em métricas de qualidade de serviço, mas também demonstrou uma eficiência de recursos e uma simplicidade operacional que se traduzem em um Custo Total de Propriedade significativamente menor. O método AHP permitiu sintetizar esses múltiplos fatores, confirmando que a abordagem mais integrada e de menor complexidade do CloudStack oferece a melhor relação custo-benefício.

A análise comparativa dos indicadores de desempenho, realizada nesta seção,

apresenta a disparidade de eficiência entre as plataformas CloudStack e OpenStack. Os dados empíricos demonstram que o CloudStack mantém uma latência e um tempo de resposta invariavelmente baixos, com média de 11,9 ms.

Em contrapartida, a latência no OpenStack com a instância 'SMALL' ou a aplicação de uma carga de 900 requisições resulta em uma degradação de desempenho de até 20 vezes em comparação com seus cenários ótimos.

Estes resultados empíricos, que apontam para um consumo de recursos de *CPU* e memória consistentemente mais elevado no OpenStack, encontram forte respaldo na literatura acadêmica que aborda a diferença entre as arquiteturas das plataformas. Três fatores principais contribuem para este fenômeno: a complexidade do plano de controle, o *overhead* da virtualização de rede e a natureza de suas arquiteturas.

A arquitetura do OpenStack é, por *design*, um conjunto de múltiplos projetos e serviços independentes (ex: Nova, Neutron, Keystone) que se comunicam via APIs. Essa abordagem distribuída, embora flexível, apresenta um maior *overhead*, pois cada serviço consome ciclos de *CPU* e memória para sua própria execução e para a comunicação constante entre eles (PATEL; PATEL, 2016). Adicionalmente, seu subsistema de rede, o Neutron, é complexo e, ao utilizar componentes como o Open vSwitch (OVS), impõe uma carga de processamento de pacotes significativa sobre a *CPU* do *host*, o que se reflete diretamente na maior utilização do processador e na maior latência observadas nos experimentos (CHAUDHRY, 2015).

O OpenStack foi concebido para ser um “*framework*” de construção de nuvens, oferecendo máxima flexibilidade para a criação de soluções de hiperescala. O CloudStack, por sua vez, origina-se de uma abordagem mais “*turnkey*” (pronta para uso), que prioriza a simplicidade operacional e a eficiência em um conjunto de funcionalidades mais integrado (S. FILHO; G. FILHO, 2016).

Portanto, o maior consumo de recursos do OpenStack é uma consequência direta de sua modularidade, enquanto a arquitetura mais simples do CloudStack apresenta uma melhor eficiência, observada em um ambiente de teste controlado como o realizado nesta pesquisa. Embora a capacidade máxima de vazão de ambas as plataformas tenha sido

comparável (na faixa de 14-15 req/s), a eficiência sistêmica do CloudStack apresentou melhores resultados, pois atingiu essa marca com um custo computacional e de latência inferior. Ao alcançar sua vazão máxima (14,93 req/s), o CloudStack utilizou apenas **6,73%** de *CPU* e manteve uma latência de **0,01** segundos. Em contrapartida, para o OpenStack atingir sua vazão máxima comparável (15,07 req/s), a plataforma exigiu **57,16%** de *CPU* e registrou uma latência de **0,20** segundos.

Esta complexidade de *trade-offs* justifica a aplicação de um método de decisão formal como o AHP, projetado especificamente para estruturar tais problemas (SAATY, 1990b). Ao ponderar os critérios conforme as premissas da pesquisa, que valorizam a eficiência de recursos, a análise AHP quantificou e confirmou as observações empíricas. O resultado final, que atribuiu ao CloudStack uma pontuação de preferência de 76,3% contra 23,7% do OpenStack, não é apenas um *ranking*, mas a validação metodológica de que, para um cenário que prioriza o desempenho de recursos e a previsibilidade, o CloudStack emergiu como a solução tecnicamente viável neste estudo.

6 Conclusões e Trabalhos Futuros

Este capítulo consolida as conclusões obtidas com a pesquisa, destacando os avanços alcançados na avaliação de infraestruturas de nuvem privada e confirmando o cumprimento do objetivo geral de estabelecer um processo decisório estruturado. A realização deste estudo permitiu não apenas identificar a plataforma mais adequada para o cenário proposto, respondendo à questão central da pesquisa sobre a seleção baseada em desempenho e custo, mas também formalizar conhecimentos e métodos que auxiliam na tomada de decisão estratégica em TI. As principais contribuições deste trabalho, que abrangem inovações metodológicas, insights práticos de desempenho e modelagem econômica, são detalhadas a seguir.

- **Metodologia de Apoio à Decisão e Artefato de Software:** A pesquisa apresenta e aplica uma metodologia combinada, utilizando o Planejamento de Experimentos (DOE) para a coleta e análise estatística dos dados e o Processo de Análise Hierárquica (AHP) para a tomada de decisão multicritério. Em alinhamento com os objetivos específicos, foi desenvolvido um microserviço de gerenciamento de biblioteca para atuar como carga de trabalho padronizada. Essa metodologia, validada pelo artefato de software e pelo fluxo experimental, pode ser replicada por outras organizações e pesquisadores para avaliar diferentes aplicações.
- **Diagnóstico de Desempenho e Arquitetura:** O estudo fornece insights diretos e práticos para profissionais de TI, quantificando *trade-offs* de desempenho e eficiência entre arquiteturas distintas. Os resultados demonstram, por exemplo, o alto *overhead* computacional do OpenStack, decorrente de sua modularidade, em comparação com a eficiência de recursos do modelo integrado do CloudStack. Embora tenha sido observada uma paridade de desempenho técnico em vazão máxima entre as duas plataformas, a análise revelou a sensibilidade crítica da latência do OpenStack sob cargas intensas, oferecendo subsídios técnicos claros para o dimensionamento de infraestruturas.
- **Modelagem de Custo (TCO) Aplicada:** A pesquisa propõe e aplica um modelo de

Custo Total de Propriedade detalhado, focado especificamente nos custos de um ambiente de nuvem privada. Esta abordagem inova ao estabelecer uma conexão direta entre o desempenho empírico aferido nos testes de utilização de recursos e o custo operacional real. O modelo traduz a eficiência de hardware (*CPU* e Memória) em economia no Custo de Energia Elétrica (OPEX) e, crucialmente, quantifica o Custo da Equipe Técnica (OPEX) com base na complexidade de gerenciamento de cada plataforma. Isso oferece uma análise tecno-econômica de valor prático para a tomada de decisão em infraestruturas locais, atendendo ao objetivo de justificar investimentos não apenas pela aquisição, mas pela operação .

Por fim, a síntese destes fatores através do método AHP permitiu concluir que, para organizações que priorizam a eficiência de custos e a previsibilidade de desempenho em cenários de recursos limitados, a plataforma CloudStack apresenta-se como a alternativa com melhores resultados de acordo com os critérios, validando a hipótese de que a simplicidade arquitetural impacta positivamente tanto a sustentabilidade econômica quanto a eficiência técnica.

6.1 Limitações

As limitações identificadas concentram-se em três dimensões principais: a infraestrutura de Hardware de Teste utilizada, a Especificidade da Carga de Trabalho aplicada e o Escopo da Implementação da arquitetura. Cada um desses aspectos e seus respectivos impactos são detalhados a seguir.

A principal limitação reside na infraestrutura de hardware utilizada, composta por um único servidor. Embora suficiente para a implementação e o teste funcional das plataformas, este ambiente não representa um cluster de produção de nível empresarial, que normalmente conta com múltiplos nós, hardware de servidor dedicado, armazenamento de alta performance e redes redundantes. O comportamento de ambas as plataformas, especialmente em relação à contenção de recursos e escalabilidade, pode ser diferente em um ambiente mais robusto.

Os resultados são específicos para a carga de trabalho baseada em microsserviços e testes de API utilizados. Aplicações com perfis diferentes (ex: I/O de disco intensivo, processamento de Big Data, cargas de trabalho para GPU) poderiam apresentar resultados e gargalos distintos em cada plataforma.

A análise foi conduzida em uma configuração de nó único para cada plataforma, não avaliando, portanto, aspectos cruciais de um ambiente de produção, como alta disponibilidade, balanceamento de carga entre múltiplos nós e a performance da rede definida por software em um ambiente distribuído.

6.2 Trabalhos Futuros

Com base nas conclusões e limitações deste trabalho, algumas direções para pesquisas futuras podem ser exploradas:

Análise comparativa das nuvens privadas OpenStack e CloudStack configuradas em um cluster de múltiplos nós com hardware de classe empresarial, para validar se os diferenciais de eficiência e os gargalos de desempenho observados se mantêm em um ambiente de produção mais realista.

Executar novos conjuntos de testes com aplicações de perfis distintos (ex: banco de dados transacionais, aplicações de I/O intensivo), para criar um mapa de desempenho mais abrangente e determinar qual plataforma é mais adequada para cada tipo de carga.

Manter as plataformas em operação contínua por períodos extensos (dias a semanas) para simular condições reais de uso prolongado.

Incorporar critérios qualitativos na análise AHP, como facilidade de implantação, qualidade da documentação, suporte da comunidade e complexidade de gerenciamento, para fornecer uma recomendação ainda mais abrangente e alinhada às necessidades de equipes de operações de TI.

Referências

- ALI, S. et al. A systematic review of performance evaluation of cloud computing systems using design of experiments. **Journal of Systems and Software**, v. 173, 2021.
- (ANEEL), A. N. D. E. E. **Resolução Homologatória Nº 3.456, de 22 de abril de 2025**. 2025. Disponível em: <https://biblioteca.aneel.gov.br/resolucoes>. Acesso em: 10/10/2025.
- CHAUDHRY, J. et al. Performance evaluation of OpenStack Neutron with Open vSwitch. In: **2015 IEEE 4TH INTERNATIONAL CONFERENCE ON CLOUD NETWORKING (CLOUDNET)**. Niagara Falls, ON, Canada: IEEE, 2015. v. 258, p. – 253.
- DELL. **Servidor em torre PowerEdge T160**. 2025. Disponível em: https://www.dell.com/pt-br/shop/servidores-armazenamento-rede/servidor-em-torre-poweredge-t160/spd/poweredge-t160/pe_t160_18389_bcc_1. Acesso em: 10/10/2025.
- DRAGOLESCU, A. A.; GOTH, C. A survey on microservices architecture: challenges, solutions, and future trends. **Journal of Cloud Computing**, v. 11, n. 1, 2022.
- FRANCESCO, P. D.; MALAVOLTA, I.; LAGO, P. A systematic literature review on the architectural trade-offs of microservices. **Journal of Systems and Software**, v. 179, 2021.
- HASAN, M. R. et al. A systematic review of the cloud computing adoption process in organizations. **Journal of Open Innovation: Technology, Market, and Complexity**, v. 8, n. 4, p. 196 – 196, 2022.
- HUAMANÍ, G. T. H. Modelo AHP para seleccionar proveedores de Cloud Computing. p. 100 – 106, 2014.
- JAIN, R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. New York: Wiley, 1991.
- LILJA, D. J. Measuring Computer Performance: A Practitioner's Guide. Cambridge: Cambridge University Press, 2000.
- MACIEL, P. R. M. Performance, Reliability, and Availability Evaluation of Computational Systems, Volume I. Recife: Editora UFPE, 2022.
- MISHRA, A.; KUMAR, R. Ranking of cloud services by applying BWM-TOPSIS, BWM-ARAS, and BWM-COPRAS hybrid MCDM methods. **International Journal of Data Science and Analytics**, 2025. Disponível em: <https://doi.org/10.1007/s41060-025-00775-2>.
- PATEL, P. B.; PATEL, A. M. A survey on OpenStack and its components. In: **INTERNATIONAL CONFERENCE ON COMMUNICATION AND SIGNAL PROCESSING (ICCSP)**. Melmaruvathur, India: IEEE, 2016. p. 1386 – 1390.
- S. FILHO, J. V. D.; G. FILHO, E. A. D. M. OpenStack e CloudStack: Uma Análise Comparativa das Plataformas de Orquestração de Nuvem Open Source. In: **COMPUTAÇÃO**,

- S. B. de (coord.). **WORKSHOP DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO DAS INSTITUIÇÕES FEDERAIS DE ENSINO SUPERIOR DO BRASIL (WTICIFES)**. Porto Alegre: [s.n.], 2016. p. 283 – 292.
- SAATY, T. L. Decision making with the analytic hierarchy process. **International Journal of Services Sciences**, v. 1, n. 1, p. 83 – 98, 2008.
- SAATY, T. L. How to make a decision: The Analytic Hierarchy Process. **European Journal of Operational Research**, v. 48, p. 9 – 26, 1990a.
- SAATY, T. L. How to make a decision: The Analytic Hierarchy Process. In: **European Journal of Operational Research**. [S.l.]: European Journal of Operational Research, 1990b. v. 48, n. 1, p. 9 – 26.
- SHARMA, M.; SEHRAWAT, R. Investigating the determinants influencing the adoption decision of cloud computing in the healthcare sector: An integrated ISM-AHP-TOPSIS approach. **Technology in Society**, v. 61, 2020. Disponível em: <https://doi.org/10.1016/j.techsoc.2020.101258>.
- SINGH, P.; KAUR, A. A comparative analysis of OpenStack and CloudStack cloud management platforms. In: **5th International conference on computing methodologies and communication (ICCMC)**. Erode, India: IEEE, 2021. p. 1297 – 1303.
- YOUSEFNEZHAD, N.; FARD, A. M.; HEMMATI, A. A systematic review on cloud computing: A service-oriented perspective. **Journal of Network and Computer Applications**, v. 161, 2020.
- ZIMMERMANN, O. Architectural Blueprints for Microservice-Based Systems: A Pattern- and Decision-Oriented Perspective. **IEEE Transactions on Services Computing**, v. 14, n. 5, p. 1249 – 1261, 2021.