

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

MODELO FUZZY DE CARACTERIZAÇÃO DE APLICAÇÕES EM AMBIENTE  
DE COMPUTAÇÃO NA NUVEM

KILMER PEREIRA BOENTE

2024

MODELO FUZZY DE CARACTERIZAÇÃO DE APLICAÇÕES EM AMBIENTE  
DE COMPUTAÇÃO NA NUVEM

Kilmer Pereira Boente

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Produção.

Orientador: Carlos Alberto Nunes Cosenza, DSc.

RIO DE JANEIRO, RJ-BRASIL

ABRIL DE 2024

MODELO FUZZY DE CARACTERIZAÇÃO DE APLICAÇÕES EM AMBIENTE  
DE COMPUTAÇÃO NA NUVEM

Kilmer Pereira Boente

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA  
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE  
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE  
EM CIÊNCIAS EM ENGENHARIA DE PRODUÇÃO.

Examinada por:

---

Prof. Carlos Alberto Nunes Cosenza, D.Sc.

---

Prof. Mario Cesar Rodriguez Vidal, D.Sc.

---

Prof. Harvey José Santos Ribeiro Cosenza, D.Sc.

---

Prof. Ricardo Marciano dos Santos, D.Sc.

RIO DE JANEIRO, RJ-BRASIL

ABRIL DE 2024

## CIP - Catalogação na Publicação

P671m      Pereira Boente, Kilmer  
              MODELO FUZZY DE CARACTERIZAÇÃO DE APLICAÇÕES EM  
              AMBIENTE DE COMPUTAÇÃO NA NUVEM / Kilmer Pereira  
              Boente. -- Rio de Janeiro, 2023.  
              127 f.

              Orientador: Carlos Alberto Nunes Cosenza.  
              Dissertação (mestrado) - Universidade Federal do  
Rio de Janeiro, Instituto Alberto Luiz Coimbra de  
Pós-Graduação e Pesquisa de Engenharia, Programa de  
Pós-Graduação em Engenharia de Produção, 2023.

              1. Computação em Nuvem. 2. Virtualização. 3. Lógica  
Fuzzy. I. Alberto Nunes Cosenza, Carlos, orient.  
II. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

## **DEDICATÓRIA**

Agradeço a Deus por tudo.

A minha esposa Michelle, pela compreensão, paciência, apoio, incentivo, amor e carinho, dedicados a mim ao longo desta Dissertação.

A minha filha Ana Luísa, que foi o meu combustível para superar todas as adversidades que surgiram ao longo do percurso.

A meus pais, mãe Marinalva, pai Alfredo Boente †, por terem me ensinado que somente com trabalho, determinação e fé que se consegue novas conquistas.

Ao meu irmão Alfredo, por nunca duvidar que eu fosse capaz de concluir o meu Mestrado numa Universidade Federal de renome como a COPPE/UFRJ.

## **AGRADECIMENTOS**

Primeiramente agradeço a DEUS pelo dom da vida, e pela força e fé necessária nos momentos de desânimo quando adversidades apareciam em minha frente.

Ao Professor Doutor Carlos Alberto Nunes Cosenza, pela preciosa e sábia orientação; pela amizade e paciência nos momentos de indecisão; pelo incentivo e pela confiança em mim depositados ao longo de todo o Curso de Mestrado.

Ao Professor Doutor Edison Conde Perez dos Santos, pela amizade, consideração e conhecimentos adquiridos acerca da Lógica Fuzzy e computação em nuvem extremamente úteis para a conclusão desta Dissertação.

Ao Professor Doutor Alfredo Nazareno Pereira Boente, meu irmão e padrinho, pelo incentivo, força e crédito a mim depositados, acreditando desde o início de que eu seria capaz de concluir este Mestrado.

Ao Professor Doutor Vinícius Marques, grande amigo, pela amizade e por ser um grande incentivador e ter acreditado que eu teria potencial e capacidade de construir uma Dissertação de Mestrado independentemente do tipo de pesquisa a ser realizada.

Aos Professores que contribuíram de certa forma para a minha formação profissional em Engenharia de Produção, tornando-me capaz de concluir esta Dissertação.

Aos funcionários de apoio administrativo do Programa de Engenharia de Produção, em especial a Lindalva Barbosa de Araújo e Roberta de Matos.

A todos o meu Muito Obrigado!

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (MSc.)

## MODELO FUZZY DE CARACTERIZAÇÃO DE APLICAÇÕES EM AMBIENTE DE COMPUTAÇÃO NA NUVEM

Kilmer Pereira Boente

Abril/2024

Orientador: Carlos Alberto Nunes Cosenza

Programa: Engenharia de Produção

O presente trabalho propõe um modelo que visa a análise de performance dos recursos computacionais da computação em nuvem, especificamente, carga de CPU, uso de memória e escalabilidade de recursos, abordando a utilização de algoritmos baseados em linguagem de programação Python, bem como fundamentação em método *fuzzy* e algoritmo desenvolvido para auxílio à tomada de decisão na detecção da melhor performance em escalabilidade. O interesse pelas áreas da engenharia da produção, engenharia da computação, ciência da computação, lógicas, gestão, proatividade em estudar a aplicabilidade de Lógica *Fuzzy* com apoio da Ciência de Dados, são fatores que favorecem a elaboração da presente dissertação de mestrado, bem como o importantíssimo acompanhamento da rede de pesquisas dos estudos anteriores desenvolvidos no Programa de Engenharia de Produção da Universidade Federal do Rio de Janeiro.

O presente trabalho poderá servir como apoio a outros estudantes e pesquisadores que se interessam pelo assunto, auxiliando na elaboração de projetos e trabalhos na área da engenharia da produção, engenharia da computação, ciência da computação, administração e áreas correlatas a presente dissertação.

**Palavras-chave:** Computação em Nuvem; Virtualização; Lógica *Fuzzy*; Eficiência de Recursos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (MSc.)

## FUZZY MODEL FOR CHARACTERIZING APPLICATIONS IN A CLOUD COMPUTING ENVIRONMENT

Kilmer Pereira Boente

April/2024

Advisor: Carlos Alberto Nunes Cosenza

Department: Industrial Engineering

This work proposes a model aimed at analyzing the performance of computational resources in cloud computing, specifically CPU load, memory usage, and resource scalability. It focuses on the use of algorithms based on Python programming language, as well as foundations in fuzzy logic and an algorithm developed to assist decision-making in detecting optimal scalability performance. The interest in the fields of production engineering, computer engineering, computer science, logics, management, and proactivity in studying the applicability of Fuzzy Logic with the support of Data Science, are factors that contribute to the elaboration of this master's dissertation, as well as the invaluable participation in the research network of previous studies developed in the Production Engineering Program of the Federal University of Rio de Janeiro. This work may serve as support to other students and researchers interested in the subject, assisting in the development of projects and work in the fields of production engineering, computer engineering, computer science, administration, and related areas to this dissertation.

**Keywords:** Cloud Computing; Virtualization; Fuzzy Logic; Resource Efficiency.

## Sumário

<b>1. INTRODUÇÃO</b>	15
1.1. APRESENTAÇÃO	15
1.2. MOTIVAÇÃO	16
1.3. OBJETIVOS GERAL E ESPECÍFICO	17
1.4. A IMPORTÂNCIA DO TRABALHO	17
1.5. FORMULAÇÃO DO PROBLEMA DE PESQUISA	18
1.6. ORIENTAÇÕES METODOLÓGICAS	19
1.7. ORGANIZAÇÃO DO TRABALHO	19
<b>2. REFERENCIAL TEÓRICO</b>	21
2.1. COMPUTAÇÃO EM NUVEM	22
2.2. ARQUITETURA COMPUTACIONAL EM NUVEM	24
2.3. VIRTUALIZAÇÃO	26
2.4. QoS EM COMPUTAÇÃO EM NUVEM	32
2.5. LÓGICA FUZZY	36
2.5.1. PYTHON	38
<b>3. APLICAÇÃO DO MODELO FUZZY</b>	41
3.1. MONITORAMENTO	42
3.2. CONFIGURAÇÃO	43
3.3. ADMINISTRAÇÃO	43
3.4. SEGURANÇA	43
3.5. PROTEÇÃO	43
3.6. MIGRAÇÃO	44
3.7. METODOLOGIA ADOTADA	44
3.8. APLICAÇÃO DO MODELO	45
3.8.1. <i>Obtenção de Dados</i>	46
3.8.2. <i>Determinação das variáveis linguísticas do modelo</i>	47
3.8.3. <i>Escolha dos termos linguísticos a serem utilizados</i>	47
3.8.4. <i>Criação das funções de pertinências para os termos fuzzy.</i>	50
3.8.5. <i>Aplicação dos questionários estruturados.</i>	53
3.8.6. <i>Coleta e tabulação de dados.</i>	55
<b>4. TÉCNICAS EMPREGADAS</b>	57
4.1. APLICAÇÃO DE ALGORITMO EM LINGUAGEM PYTHON BASEADO EM LÓGICA FUZZY	58
4.2. DIAGRAMA DE CASO DE USO	61
4.3. DIAGRAMA DE CLASSES	62
4.4. PROJETO DE BANCO DE DADOS	63
<b>5. SOLUÇÃO PARA O PROBLEMA PROPOSTO</b>	64
<b>6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS</b>	74
<b>REFERÊNCIAS</b>	77
<b>APÊNDICE</b>	80



## LISTA DE FIGURAS

Figura 1 - Papéis na computação em nuvem	23
Figura 2 - Comparativo entre a lógica clássica e a lógica fuzzy	37
Figura 3 - Etapas da construção de uma aplicação	40
Figura 4 - Área de gerenciamento do Azure	42
Figura 5 - Metodologia adotada	45
Figura 6 - Descrição do modelo fuzzy	46
Figura 7 - Conjuntos fuzzy dos termos relacionados aos graus de utilização da CPU e memória das máquinas virtuais em computação em nuvem	50
Figura 8 - Conjuntos fuzzy dos termos relacionados aos graus escalonamento de recursos das máquinas virtuais em computação em nuvem	51
Figura 9 - Conjunto fuzzy triangular (0, 0, 1)	51
Figura 10 - Conjunto fuzzy triangular (0, 1, 2)	52
Figura 11 - Conjunto fuzzy triangular (1, 2, 3)	52
Figura 12 - Conjunto fuzzy triangular (2, 3, 4)	52
Figura 13 - Conjunto fuzzy triangular (3, 4, 4)	53
Figura 14 - Fórmula do somatório dos Itens e avaliações dos especialistas de serviços em nuvem	55
Figura 15 - Dataframe ou tabela de dados obtida do Kaggle	56
Figura 16 - Fluxo de funcionamento e técnicas empregadas	57
Figura 17 - Fluxo do funcionamento do modelo MVC	58
Figura 18 - Mapa Mental da aplicação do algoritmo	59
Figura 19 - Diagramas de Caso de Uso da aplicação	61
Figura 20 - Diagrama de Classes da aplicação	62
Figura 21 - Código que define a variáveis, funções de pertinências e termos linguísticos fuzzy	66
Figura 22 - Código que define a inferências fuzzy	66
Figura 23 - Código que define o carregamento dos dados a serem tratados	67
Figura 24 - Código que define a média de consumo ou utilização de recursos de computação em nuvem	67
Figura 25 - Código que define a exibição do gráfico de consumo ou utilização de recursos de computação em nuvem por mês	68
Figura 26 - Código que define a exibição do gráfico de barras de consumo ou utilização de recursos de computação em nuvem por mês	69
Figura 27 - Código que define a exibição do gráfico de distribuição de consumo ou utilização de recursos de computação em nuvem por mês	70
Figura 28 - Código que define a exibição do gráfico triangular de utilização da CPU em nuvem	71
Figura 29 - Código que define a exibição do gráfico triangular de utilização da memória em nuvem	72
Figura 30 - Código que define a exibição do gráfico triangular de escalonamento de recursos	73

## **LISTA DE TABELAS**

Tabela 1 – Peso dos especialistas

54

## LISTA DE QUADROS

Quadro 1 - Vantagens e desvantagens da Virtualização	27
Quadro 2 - Tipos de Virtualização	29
Quadro 3 - Função de Pertinência da variável linguística CPU	47
Quadro 4 - Função de Pertinência da variável linguística Memória	48
Quadro 5 - Função de Pertinência da variável linguística Escalabilidade	49
Quadro 6 - Restrições na escolha da tecnologia (algoritmo)	64

## **LISTA DE GRÁFICOS**

Gráfico 1 - Gráfico de barras referente ao peso dos especialistas

54

## 1. INTRODUÇÃO

Este capítulo apresenta uma visão geral do trabalho desenvolvido, sua importância e os motivos pelo qual ele foi realizado.

### 1.1. APRESENTAÇÃO

A partir da concepção da internet em ambientes corporativos e residenciais, houve uma crescente demanda por maior conectividade entre os usuários, resultando no surgimento de uma ampla gama de serviços *online*. Esse desenvolvimento impulsionou a necessidade de armazenamento e processamento online. Nesse contexto, a computação em nuvem, também conhecida como *Cloud* ou *Cloud Computing*, emerge como uma abordagem tecnológica que visa proporcionar maior mobilidade, portabilidade e conveniência ao ambiente computacional. Como resultado, ocorreram investimentos em *hardware* de alta capacidade, mas com custos reduzidos, possibilitando o acesso a um público mais amplo.

Em termos simplificados, a computação em nuvem refere-se à interconexão de serviços de Tecnologia da Informação que oferecem armazenamento de dados, capacidade de processamento, conectividade, aplicativos e serviços por meio da rede. Essa tecnologia permite que os usuários acessem remotamente arquivos, aplicativos e dados de outros computadores usando apenas um computador pessoal, sem a necessidade de instalação ou *download*.

Devido ao advento da computação em nuvem (*Cloud Computing*), os modelos de programação distribuída em larga escala estão se tornando cada vez mais atrativos para os usuários finais. A abordagem baseada em serviços é considerada uma das principais razões para a ampla aceitação e adoção das nuvens computacionais. Conforme ressaltado por Governo do Brasil (2022), o modelo de computação em nuvem busca fornecer serviços computacionais sob demanda, assim como serviços comuns do cotidiano, como distribuição de energia elétrica ou serviços de telefonia. Essa abordagem permite que as empresas se concentrem mais em suas atividades principais, transferindo a responsabilidade de administrar serviços básicos de infraestrutura e serviços.

No contexto da computação em nuvem, é possível oferecer serviços de armazenamento de dados, por exemplo, sem que o cliente precise possuir uma

infraestrutura dedicada. Conforme destacado por (GOLIGHTLY et al., 2022), nesse modelo, tanto a infraestrutura quanto a interface da aplicação são disponibilizadas pela nuvem. Os serviços podem ser personalizados de acordo com as necessidades do cliente, sem que ele precise participar da instalação, configuração ou manutenção do produto. Além disso, recursos como máquinas virtuais podem ser acessados pelo cliente em qualquer lugar e a qualquer momento, sem a necessidade de envolvimento na gestão da infraestrutura.

A evolução tecnológica tem desempenhado um papel significativo no sucesso das nuvens computacionais. A utilização da virtualização em conjunto com boas técnicas de provisionamento de recursos permite uma melhor utilização dos recursos disponíveis, proporcionando maior economia e flexibilidade na configuração de ambientes. Segundo relatório de Pereira (2014, p. 69), o aumento da largura de banda, as taxas de transferência de dados mais rápidas em redes de comunicação e o incremento na capacidade de processamento dos computadores atuais têm contribuído para um melhor desempenho, permitindo que os provedores de serviços garantam níveis de qualidade de serviço compatíveis com os requisitos dos usuários.

Nesse sentido, este estudo visa relatar uma análise abrangente sobre a computação em nuvem, englobando conceitos, tecnologias existentes e os modelos de serviços oferecidos, com o intuito de aprofundar o entendimento dessa área em constante evolução.

Neste cenário é apresentado o presente trabalho que propõe uma contribuição para o setor de serviço tecnológico, associando o mesmo ao uso do método COPPE-COSENZA para apoio à tomada de decisão.

## 1.2. MOTIVAÇÃO

O presente trabalho surge do interesse pelo setor de serviço tecnológico e da possibilidade de aplicar os conhecimentos adquiridos no mestrado em Engenharia de Produção do programa de pós-graduação da COPPE/UFRJ.

O interesse surge na época da graduação, com o intuito de tentar entender essa área em crescente expansão no Brasil e no mundo e de destaque nas empresas e na economia dos países.

Ao aplicar os conhecimentos adquiridos durante o mestrado em Engenharia de Produção, há a possibilidade de colaborar de forma inovadora para a pesquisa no setor tecnológico brasileiro, alcançando assim uma realização pessoal.

### 1.3. OBJETIVOS GERAL E ESPECÍFICO

Propor um modelo *fuzzy* de caracterização de aplicações em ambientes de computação nas nuvens. A partir deste modelo, será possível discernir características da aplicação que sejam capazes de ser classificadas por taxonomia de *software* e lógica *fuzzy*, visto que cada categorização disponibiliza uma solução de arquitetura sugerida ou sinalizada. Este modelo propõe possibilidades alternativas para que os provedores da nuvem sejam capazes de controlar, configurar e escalar os ambientes computacionais conforme a necessidade do cliente ou conforme a necessidade de negócio que o provedor da nuvem possa vir a demandar.

Em relação aos objetivos específicos, buscou-se identificar os critérios da avaliação das aplicações em ambiente de computação em nuvens, perspectiva do consumo de recursos, perspectiva da influência das máquinas virtuais em aplicações distribuídas, paralelas e transacionais, discutir a subjetividade da categorização de arquiteturas computacionais de provedores (teoria dos conjuntos *fuzzy*) ao conhecimento e experiência em seus juízos de valor. Discutir as implicações destes resultados frente a modelos de arquitetura computacionais de problematização, bem como infraestrutura, máquinas virtuais e consumo de recurso.

### 1.4. A IMPORTÂNCIA DO TRABALHO

Nos últimos anos, a computação na nuvem emergiu como um paradigma dominante para a entrega de serviços solicitados sob uma arquitetura de computação distribuída, como IaaS<sup>1</sup>, PaaS<sup>2</sup> e SaaS<sup>3</sup> (HAYAT; KIM, 2017, p. 589).

A computação em nuvem revolucionou a maneira como interagimos com a tecnologia, trazendo inúmeras vantagens e transformações significativas para o mundo dos negócios e da pesquisa, pois através dela, é possível acessar serviços e aplicativos de qualquer lugar do mundo, a qualquer momento, apenas com uma conexão à internet,

---

<sup>1</sup> IaaS - Infraestrutura como serviço.

<sup>2</sup> PaaS - Plataforma como serviço.

<sup>3</sup> SaaS - *Software* como serviço.

oferecendo flexibilidade sem precedentes, especialmente em um mundo cada vez mais móvel e conectado.

A escalabilidade é uma das maiores vantagens da computação em nuvem, visto que empresas podem aumentar ou diminuir seus recursos conforme a demanda, sem a necessidade de grandes investimentos em infraestrutura física, o que torna a nuvem ideal para *startups* e pequenas empresas, que podem acessar tecnologias de ponta e armazenamento de dados com custos reduzidos.

Além disso, a computação em nuvem oferece uma segurança robusta, geralmente superior à que muitas empresas podem proporcionar localmente, bem como os provedores de serviços em nuvem investem pesadamente em sistemas de segurança, protocolos de criptografia e conformidade regulatória, protegendo os dados dos usuários contra perdas, roubos e outros tipos de ciberataques.

Por fim, a sustentabilidade é um aspecto frequentemente destacado. Ao utilizar a infraestrutura da nuvem, as empresas podem reduzir o consumo de energia e a emissão de carbono associados à manutenção de grandes centros de dados locais.

Em resumo, a computação em nuvem não apenas transformou a infraestrutura de TI, mas também democratizou o acesso a tecnologias avançadas, promovendo inovação, colaboração e sustentabilidade em uma escala global.

O estudo possui potencial de contribuição para as áreas de ciência da computação, engenharia de produção, engenharia de software, administração e lógica *fuzzy*, para endereçar as incertezas e subjetividades inerentes à caracterização de aplicações em nuvens computacionais. A natureza destas incertezas é amplamente reconhecida como um desafio central nas ciências aplicadas e na engenharia de sistemas, fazendo com que a pesquisa sobre modelos *fuzzy* seja de interesse crítico.

## 1.5. FORMULAÇÃO DO PROBLEMA DE PESQUISA

Como os serviços em ambientes de computação em nuvem podem atingir a eficiência e eficácia efetiva por meio da caracterização de aplicações de computação em nuvem?

## 1.6. ORIENTAÇÕES METODOLÓGICAS

As orientações metodológicas são baseadas na compreensão dos conceitos sobre logística, lógica *fuzzy*, computação em nuvem, no atual panorama da arquitetura computacional no mundo, na aplicabilidade de algoritmo para auxílio à tomada de decisão em um problema de performance, obtidos por meio de livros, artigos, internet, jornais e revistas, além das sugestões dos orientadores do presente trabalho e orientações de *experts* das áreas envolvidas, caracterizando, portanto uma pesquisa bibliográfica.

Os experimentos realizados por meio de programação Python e aplicação do modelo *fuzzy* proposto, caracteriza uma pesquisa de cunho experimental.

As interpretações e tratamento de dados discretos e com base na opinião dos especialistas, fuzificados e defuzificados, caracteriza uma pesquisa quali-quantitativa.

#### 1.6.1. Roteiro Metodológico

O roteiro metodológico para o presente trabalho é assim apresentado:

- a) Resumo sobre conceitos relacionados ao problema, como computação em nuvem, arquitetura computacional, virtualização, QoS<sup>4</sup> em computação em nuvem, lógica *fuzzy* e algoritmo;
- b) Formulação do Problema Proposto;
- c) Técnicas empregadas para a solução do problema;
- d) Solução teórica para análise de performance de recursos para serviços em nuvem utilizando modelo *fuzzy*;
- e) Solução teórica apresentada para o problema proposto; e
- f) Considerações finais e trabalhos futuros.

#### 1.7. Organização do Trabalho

O trabalho proposto apresenta através de um algoritmo escrito em linguagem Python, alternativas de tomada de decisão multicritério baseada em um modelo de apoio à performance de melhores condições de escalabilidade e utilização dos recursos fundamentada nas técnicas da lógica *fuzzy*. Em cada capítulo é observado o

---

<sup>4</sup> QoS – Qualidade de Serviço ou *quality of service*.

desenvolvimento do trabalho, fazendo com que seja possível compreender o que se elabora no mesmo e seu resultado. A introdução aparece descrita no Capítulo 1.

No Capítulo 2 é apresentado o referencial teórico, com definições, conceitos e ideias sobre os temas encontrados no presente trabalho: computação em nuvem, arquitetura computacional, virtualização, qualidade de serviço (QoS) na computação em nuvem, lógica *fuzzy* e algoritmo.

A formulação do problema é abordada no Capítulo 3, onde é mostrado o problema de forma mais detalhada, com os resultados experimentais, as restrições inerentes ao problema, os critérios considerados importantes.

O Capítulo 4 apresenta as técnicas empregadas para a resolução do problema proposto.

A solução para o problema proposto é abordada em detalhes no Capítulo 5, enfatizando o uso de algoritmo desenvolvido na linguagem de programação Python, para a resolução do presente trabalho.

No Capítulo 6 estão descritas as considerações finais e trabalhos futuros.

## 2. REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico descrito para o presente trabalho. São abordados assuntos que auxiliam a uma análise mais detalhada do problema proposto, tais como computação em nuvem, arquitetura computacional, virtualização, qualidade de serviço (QoS) na computação em nuvem, lógica *fuzzy* e algoritmo.

Direta ou indiretamente, entendemos que esta pesquisa é importante para as áreas supracitadas, pois trata das incertezas e subjetividades inerentes à caracterização de tecnologias em nuvens computacionais, cujos alicerces de força e organização são interesses centrais de todas as ciências de grande campo atualmente, e do crescente campo da engenharia de sistemas.

A dinâmica evolutiva da computação em nuvem é marcada pela diversidade de abordagens e políticas adotadas por cada provedor de serviços, sendo predominantemente representada por três estruturas principais: Plataforma como Serviço (PaaS), Infraestrutura como Serviço (IaaS) e Software como Serviço (SaaS). Essas estruturas capacitam os provedores a disponibilizarem desde máquinas virtuais simples até interfaces específicas para plataformas ou produtos para uso (RONG; ZHAO, 2021).

Do ponto de vista dos provedores de infraestrutura em nuvem, a vantagem está na variedade de equipamentos e arquiteturas computacionais para executar aplicações específicas, sobretudo quando há suporte para processamento distribuído e paralelo. Isso possibilita a criação de soluções flexíveis e escaláveis, gerenciando máquinas virtuais conforme a demanda dos clientes. Entretanto, a computação em nuvem enfrenta desafios notáveis, como desempenho, confidencialidade, escalabilidade, segurança e armazenamento de dados. Superar tais obstáculos requer aprimoramento dos mecanismos de provisionamento de custos, desempenho e segurança, garantindo a integridade, confidencialidade e auditoria dos dados nas máquinas virtuais, fundamentais para aplicações comerciais eficazes.

De acordo com (AZEVEDO et al., 2018), os contratos de nível de serviço (SLA) desempenham um papel crucial ao regulamentar as atividades dos provedores de nuvem, assegurando qualidade e disponibilidade do ambiente para ambas as partes. Quando há necessidade de recorrer a ambientes computacionais de terceiros, isso deve ser explicitado no contrato para informar os clientes sobre possíveis transferências de máquinas virtuais. Após essa definição, os clientes têm liberdade para utilizar as

máquinas conforme suas necessidades, implementando métodos para monitorar o desempenho e gerenciar falhas.

Do lado do cliente da nuvem, há uma mudança na abordagem de *design* e arquitetura das soluções computacionais, permitindo a configuração de aplicações corporativas em infraestruturas privadas ou migração para a nuvem em momentos de escassez de recursos. Isso possibilita a combinação de infraestruturas públicas e privadas, além da criação de ambientes redundantes.

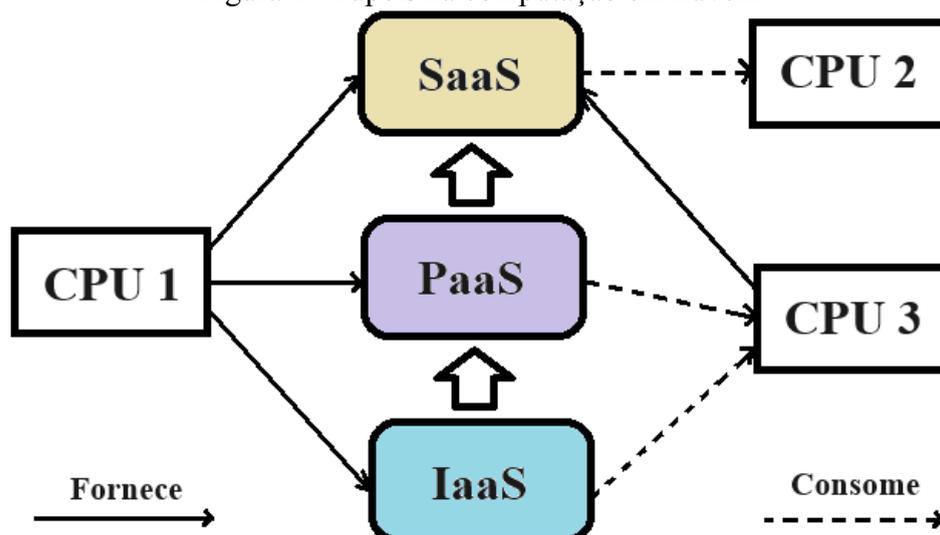
Os provedores também enfrentam mudanças, precisando mensurar o uso das máquinas virtuais e converter isso em tarifas, viabilizando o acesso aos serviços de infraestrutura sem a necessidade de aquisição de recursos próprios. Com a responsabilidade pelo processamento, monitoramento e escalabilidade na nuvem, os clientes se desvencilham da gestão direta da infraestrutura para suas aplicações.

Em suma, após a criação da máquina virtual, a transferência e configuração dos aplicativos na nuvem pública, via interface web, demandam um estudo aprofundado do desempenho em ambientes virtuais para garantir sua eficácia, é neste contexto da performance de recursos em nuvem que trabalharemos a pesquisa desta dissertação.

## 2.1. COMPUTAÇÃO EM NUVEM

A computação em nuvem pública é uma abordagem que permite que as organizações executem serviços em uma infraestrutura compartilhada, proporcionando uma mudança do modelo de custos de capital para custos operacionais. Empresas como *Google*, *Microsoft* e *Amazon* oferecem serviços de *cloud* pública, visando melhorar a eficiência e reduzir os custos operacionais (CÂNDIDO; ARAÚJO JÚNIOR, 2022).

Figura 1 - Papéis na computação em nuvem



Fonte: Adaptado de Neto, 2015.

Conforme ilustração da Figura 1, existem três abordagens principais na computação em nuvem: Infraestrutura como um Serviço (IaaS), *Software* como um Serviço (SaaS) e Plataforma como Serviço (PaaS). No modelo IaaS, o usuário aluga recursos de *hardware*, como servidores e firewalls, e configura remotamente a infraestrutura de acordo com suas necessidades. No modelo PaaS, o cliente aluga uma plataforma para desenvolver e implementar seus aplicativos, sem se preocupar com a infraestrutura subjacente. No modelo SaaS, o cliente utiliza um serviço fornecido pelo provedor, sem precisar se preocupar com a infraestrutura ou o *software* subjacente (RONG; ZHAO, 2021).

A computação em nuvem pode ser implementada em diferentes modelos, como nuvem privada, nuvem pública, nuvem comunitária e nuvem híbrida. Cada modelo possui características específicas de controle, gerenciamento e compartilhamento de recursos.

Um modelo de computação em nuvem deve apresentar características como autoatendimento sob demanda, amplo acesso a serviços de rede, pool de recursos compartilhados, elasticidade rápida e serviços mensuráveis. Essas características permitem que os usuários obtenham recursos computacionais conforme suas necessidades, pagando apenas pelo que utilizam.

Na arquitetura de computação em nuvem, existem diferentes papéis desempenhados pelos provedores e pelos usuários. O provedor é responsável por

disponibilizar, gerenciar e monitorar a infraestrutura, enquanto os programadores e os usuários finais são liberados dessas responsabilidades.

Os fornecedores de computação em nuvem oferecem um conjunto de serviços que podem ser alugados pelos clientes, permitindo que eles utilizem recursos distribuídos sem se preocupar com a tecnologia subjacente. A responsabilidade pelo gerenciamento e segurança varia de acordo com o modelo de serviço, sendo que no modelo SaaS, o provedor é responsável por essas questões, enquanto no modelo IaaS, o cliente assume mais responsabilidades.

Em resumo, a computação em nuvem pública é uma abordagem que oferece serviços de infraestrutura, plataforma e *software* de forma flexível e escalável. Os usuários podem utilizar esses serviços conforme suas necessidades, pagando apenas pelo que utilizam, sem se preocupar com a infraestrutura subjacente. Essa abordagem traz benefícios como redução de custos, maior eficiência operacional e acesso a recursos computacionais sob demanda.

## 2.2. ARQUITETURA COMPUTACIONAL EM NUVEM

A arquitetura computacional em nuvem é um campo emergente que combina princípios tradicionais de arquitetura de sistemas com os recursos expansíveis da computação em nuvem.

A evolução para infraestruturas baseadas na nuvem reflete a necessidade de escalabilidade, resiliência e agilidade para atender às demandas de negócios dinâmicas e ao desenvolvimento acelerado, como inteligência artificial (IA) e otimizações de operações e fluxo de trabalho. (INTEL, 2023).

Um projeto bem-sucedido de nuvem deve começar com a avaliação das cargas de trabalho atuais e necessidades corporativas, pois a infraestrutura de nuvem de *back-end*<sup>5</sup> consiste em *hardware* de *data center*, virtualização, aplicativos e serviços. As tecnologias contemporâneas, como as oferecidas pela Intel, são projetadas para otimizar o desempenho e maximizar a utilização de recursos (INTEL, 2023).

A arquitetura em nuvem deve ser projetada considerando perguntas fundamentais sobre as cargas de trabalho existentes, a utilização geral da nuvem, gargalos de desempenho e estratégias de virtualização, pois a infraestrutura em nuvem

---

<sup>5</sup> Back-end - Camada de acesso a dados de um software ou infraestrutura física, também trabalha codificação em linguagem de programação diversa.

abrange tanto a interface do usuário (*front-end*<sup>6</sup>) quanto a execução dos serviços (*back-end*), que inclui o hardware do data center, virtualização e camadas de aplicativos e serviços, todos comunicando-se através do *middleware*<sup>7</sup>.

Quando se trata de provedores de computação em nuvem, as três principais grandes empresas competidoras, atualmente no mercado de computação em nuvem, são Microsoft Azure, também chamado de Azure, Google Cloud e AWS. Essas grandes empresas estão investindo em seus próprios funcionários e engenheiros, oferecendo certificações em suas plataformas. Embora a *AWS* seja a líder de mercado, o *Google Cloud* e o *Azure* estão crescendo rapidamente e mostrando um crescimento significativo, sendo assim, cada provedor de nuvem tem sua própria oferta de serviços e recursos.

A *AWS*, por exemplo, oferece serviços como *Amazon EC2* para computação, *Amazon S3* e *Amazon EBS* para armazenamento, *Amazon VPC* para rede e *Amazon RDS* e *Amazon DynamoDB* para bancos de dados. A infraestrutura da *AWS* é dividida em regiões e zonas de disponibilidade, garantindo a disponibilidade e independência dos serviços.

O Azure, por sua vez, associa regiões isoladas fisicamente em conjuntos de disponibilidade e incentiva os usuários a arquitetar seus sistemas em torno dessas regiões. Seus serviços incluem máquinas virtuais do Azure, *Azure App Service*, *Azure Kubernetes Service*, *Azure Blob Storage*, *Azure Virtual Network* e várias opções de bancos de dados, como *Azure SQL Database* e *CosmosDB* (INTEL, 2023).

O *Google Cloud* também adota uma estratégia de isolamento e disponibilidade, e embora não prescreva pareamentos regionais específicos, os usuários precisam arquitetar seus aplicativos em várias regiões para alcançar alta disponibilidade. Os principais serviços do *Google Cloud* incluem *Compute Engine*, *App Engine*, *Google Kubernetes Engine*, *Cloud Storage*, *Cloud SQL* e *Cloud Spanner*.

Cada provedor de nuvem tem suas próprias vantagens e recursos exclusivos, e a escolha do provedor certo depende das necessidades específicas do usuário. É importante considerar fatores como eficiência, quantidade de serviços disponíveis e custos associados a cada plataforma.

---

<sup>6</sup> Front-end - Camada de abstração que visa oferecer uma interface amigável ao usuário.

<sup>7</sup> Middleware é uma camada de software que fornece serviços comuns e capacidades de comunicação entre o front-end e o back-end.

Em resumo, o mercado de computação em nuvem está aquecido, e além dos três principais provedores, existem outras opções emergentes no mercado. Cada provedor tem sua própria oferta de serviços e recursos, e os usuários devem avaliar cuidadosamente suas necessidades para escolher a plataforma que melhor atenda a seus requisitos específicos.

### 2.3. VIRTUALIZAÇÃO

O uso de uma aplicação dentro de uma plataforma em nuvem, como o *Azure* da Microsoft, oferece diversas vantagens e funcionalidades. Para começar, a criação de uma conta Azure é simples e pode ser feita gratuitamente, o que permite aos usuários explorar os serviços disponíveis por um período de 12 meses. Ao entrar no portal do *Azure*, é possível visualizar uma variedade de serviços e funcionalidades oferecidos, pois os serviços mais populares são destacados, facilitando o acesso rápido. Além disso, há uma ampla gama de serviços disponíveis em diferentes categorias, conforme ilustrado na tela do portal.

No caso da implementação de uma aplicação que requer uma máquina virtual, o *Azure* fornece a flexibilidade necessária para escolher as configurações adequadas. É possível selecionar a região desejada e o sistema operacional, como *Linux* ou *Windows Server*. A configuração da máquina virtual pode ser ajustada de acordo com a necessidade de recursos, e o Azure também oferece opções de segurança, como chaves de acesso SSH.

Após a configuração da máquina virtual, é possível acompanhar o monitoramento dos recursos, como segurança, recuperação de dados e *backups*. A plataforma do *Azure* é intuitiva e fornece todas as informações necessárias para gerenciar e monitorar a máquina virtual (MICROSOFT AZURE, 2023).

Para ter acesso total à máquina virtual, é necessário utilizar um *software* SSH, como o *PuTTY*. A chave PEM, fornecida pelo Azure durante a criação da máquina virtual, é essencial para acessar o servidor. Por meio do SSH, é possível ter controle total sobre o sistema operacional, realizar instalações de aplicativos e configurar o ambiente de acordo com as necessidades do projeto.

Ao longo do processo de utilização do servidor na nuvem, é possível monitorar todos os recursos e garantir o pleno funcionamento da aplicação. O *Azure* oferece

ferramentas de monitoramento que permitem acompanhar o desempenho do servidor e realizar ajustes quando necessário.

As vantagens e desvantagens do processo de virtualização podem ser melhor compreendidas através do Quadro 1.

Quadro 1 - Vantagens e desvantagens da Virtualização

<b>Vantagens</b>	<b>Descrição</b>	<b>Desvantagens</b>	<b>Descrição</b>
<b>Consolidação de Servidores</b>	Possibilita que múltiplas instâncias virtuais coexistam em um único servidor físico, otimizando a utilização de recursos e mitigando a necessidade de operar e manter um parque de servidores físicos extenso, resultando em economia de espaço físico, energia e despesas operacionais associadas à refrigeração.	<b>Sobrecarga de Desempenho na Virtualização</b>	A virtualização incute uma inércia no desempenho devido à interposição de uma camada de software adicional. Apesar da minimização dessa sobrecarga ao longo do tempo, ela pode ainda influenciar o desempenho de aplicações intensivas em recursos.
<b>Eficiência de Custos</b>	Essa abordagem diminui os custos capitais e operacionais relacionados à aquisição e manutenção de <i>hardware</i> .	<b>Centralização de Riscos em Ambientes Virtuais</b>	A virtualização, apesar de ampliar a disponibilidade sistêmica, pode introduzir uma centralização de riscos no servidor hospedeiro. Uma falha neste ponto pode comprometer todas as máquinas virtuais associadas.
<b>Isolamento</b>	Cada máquina virtual opera como uma entidade autônoma, proporcionando um isolamento eficaz e reforçando a segurança entre diferentes aplicações e sistemas operativos. A independência das máquinas virtuais, assegura que falhas ou vulnerabilidades em uma não se propaguem para nas demais.	<b>Complexidade de Gestão Virtual</b>	A administração de ambientes virtualizados exige um conhecimento especializado em tecnologias de virtualização, representando uma complexidade acentuada em comparação com infraestruturas convencionais.

<p><b>Flexibilidade e Escalabilidade</b></p>	<p>Caracteriza-se pela sua agilidade e capacidade de escalabilidade. A gestão dessas instâncias virtuais, incluindo a criação, clonagem e migração, é realizada com eficiência, permitindo um ajuste dinâmico de recursos e uma expansão rápida conforme a demanda operacional flutua.</p>	<p><b>Implicações Financeiras de Soluções Virtualizadas</b></p>	<p>Soluções de virtualização podem acarretar encargos financeiros associados ao licenciamento, particularmente para funcionalidades de alta complexidade ou para uso corporativo.</p>
<p><b>Recuperação de Desastres</b></p>	<p>Ferramenta facilitadora no desenvolvimento de estratégias de recuperação de desastres, permitindo a realização de backups de maneira simplificada e a restauração de sistemas virtuais de forma ágil, o que é essencial para a continuidade operacional em situações adversas.</p>	<p><b>Disputa por Recursos em Virtualização</b></p>	<p>A gestão ineficaz de máquinas virtuais pode resultar em competição por recursos computacionais no servidor hospedeiro, culminando em degradação do desempenho e outras ineficiências.</p>
<p><b>Teste e Desenvolvimento</b></p>	<p>Oferecem um ambiente ideal para operações de teste e desenvolvimento, com a possibilidade de capturar estados de máquinas virtuais antes de atualizações ou mudanças significativas. Se alterações resultarem em falhas, é possível reverter para um estado anterior de maneira rápida e sem riscos.</p>	<p><b>Interoperabilidade em Ambientes Virtualizados</b></p>	<p>Aplicações e softwares nem sempre são compatíveis com ambientes virtualizados, podendo resultar em questões de interoperabilidade que exigem atenção adicional para resolução.</p>
<p><b>Gestão de Recursos</b></p>	<p>Proporciona um gerenciamento refinado e detalhado dos recursos computacionais, permitindo a distribuição e alocação equitativa de recursos entre as máquinas virtuais, o que contribui para uma eficiência operacional otimizada</p>	<p><b>Vulnerabilidades de Segurança em Virtualização</b></p>	<p>A virtualização proporciona um isolamento efetivo entre as máquinas virtuais, porém, persistem riscos de segurança, especialmente se o sistema hospedeiro for vulnerável a comprometimentos de segurança.</p>

Fonte: Adaptado da Intel (2023).

"[...] a computação em nuvem se beneficiará das técnicas de virtualização, virtualizando seu hardware, permitindo o compartilhamento de acesso e utilizando os recursos de forma eficiente. Portanto, a computação em nuvem maximizará seus lucros e servirá a diversas aplicações. (ALMURISI, 2022).

Quanto aos tipos de virtualização, podem ser melhor compreendidos através do Quadro 2.

Quadro 2 - Tipos de Virtualização

Tipos de Virtualização	Descrição
<p><b>Virtualização da área de trabalho</b></p>	<p>A virtualização da área de trabalho permite executar vários sistemas operacionais de área de trabalho, cada um em sua própria VM no mesmo computador.</p>
<p><b>Virtualização de rede</b></p>	<p>A virtualização de rede usa um software para criar uma "visualização" da rede que um administrador pode usar para gerenciar a rede a partir de um único console. Ela abstrai elementos e funções de hardware (por exemplo, conexões, comutadores, roteadores etc.) e ela o faz em software em execução em um <i>hypervisor</i>.</p>
<p><b>Virtualização de armazenamento</b></p>	<p>A virtualização de armazenamento possibilita que todos os dispositivos de armazenamento na rede, estejam eles instalados em servidores individuais ou em unidades de armazenamento independentes, sejam acessados e gerenciados como um único dispositivo de armazenamento.</p>

<b>Virtualização de dados</b>	A virtualização de armazenamento possibilita que todos os dispositivos de armazenamento na rede, estejam eles instalados em servidores individuais ou em unidades de armazenamento independentes, sejam acessados e gerenciados como um único dispositivo de armazenamento.
<b>Virtualização de aplicativos</b>	A virtualização de aplicativos executa um software de aplicativo sem instalá-lo diretamente no SO do usuário.
<b>Virtualização de data center</b>	A virtualização de data center abstrai a maior parte do hardware de um data center em software, permitindo de forma eficaz que um administrador divida um único data center físico em vários data centers virtuais para diferentes clientes.
<b>Virtualização de CPU</b>	A virtualização de CPU (unidade central de processamento) é a tecnologia fundamental que viabiliza <i>hypervisores</i> , máquinas virtuais e sistemas operacionais. Ela permite que uma única CPU seja dividida em várias CPUs virtuais para uso por várias Máquinas Virtuais.

<p><b>Virtualização de GPU</b></p>	<p>Uma GPU (unidade de processamento gráfico) é um processador especial de múltiplos núcleos que melhora o desempenho geral da computação ao assumir o processamento gráfico ou matemático pesado. A virtualização de GPU permite que várias <i>VMs</i> usem toda ou parte da capacidade de processamento de uma única GPU para executar vídeos, inteligência artificial (IA) e outros aplicativos que exigem alta capacidade para executar gráficos ou cálculos matemáticos.</p>
<p><b>Virtualização de Linux</b></p>	<p>O Linux inclui o seu próprio <i>hypervisor</i>, chamado de máquina virtual com base em kernel (KVM), que oferece suporte às extensões do processador de virtualização da Intel e da AMD de modo que seja possível criar <i>VMs</i> com base em x86 a partir de dentro de um SO Linux convidado.</p>
<p><b>Virtualização de cloud</b></p>	<p>Ao virtualizar servidores, armazenamento e outros recursos de data centers físicos, provedores de <i>cloud computing</i> podem oferecer uma variedade de serviços aos clientes, incluindo IaaS, PaaS e SaaS.</p>

**Fonte: Adaptado da IBM (2023).**

A virtualização em nuvem, ao concluir uma análise abrangente de sua aplicabilidade e impacto, revela-se como uma tecnologia transformadora na era digital, pois ela não só oferece soluções escaláveis e econômicas para o gerenciamento de

infraestrutura de TI, mas também impulsiona a inovação e a agilidade nos negócios, assim como é capaz de isolar e distribuir recursos computacionais, garantindo ao mesmo tempo eficiência e segurança, ressaltando seu papel central em ambientes corporativos. Contudo, desafios como a gestão de complexidade, custos de licenciamento e questões de segurança permanecem como áreas cruciais para o desenvolvimento futuro. Portanto, a virtualização em nuvem, enquanto catalisadora de mudanças operacionais e estratégicas, continua a ser um campo fértil para pesquisa e desenvolvimento contínuos.

Em suma, o uso de uma aplicação dentro de uma plataforma em nuvem, proporciona flexibilidade, facilidade de configuração, controle total do ambiente e monitoramento eficiente dos recursos. Essas vantagens mostram como a virtualização da computação em nuvem pode oferecer uma solução eficaz para as necessidades de hospedagem e execução de aplicações.

#### 2.4. QoS EM COMPUTAÇÃO EM NUVEM

Qualidade de Serviço (QoS) é um conceito fundamental na área de redes e computação, com o objetivo de garantir a eficiência e a confiabilidade dos serviços oferecidos. A medição da latência é uma forma de avaliar a QoS e é comumente realizada utilizando pontos de referência entre os nós que solicitam um serviço e os nós que o fornecem. De acordo com (HASSANZADEH-NAZARABADI; TAHERI-BOSHROOYEH; ÖZKASAP, 2022), em ambientes de Computação em Nuvem, os protocolos baseados em DHT (*Distributed Hash Table*) podem reduzir a latência na busca de informações, pois esses protocolos possuem heurísticas para selecionar os nós mais próximos uns dos outros, resultando em uma menor latência de comunicação.

A QoS pode ser definida como a percepção do usuário em relação à eficiência de um determinado serviço. Isso inclui garantir requisitos como perdas mínimas, desempenho máximo, tempo de resposta rápido e precisão e consistência dos dados recebidos. No entanto, prover uma QoS adequada não é uma tarefa fácil devido à dinamicidade e aos desafios da *www*. Além disso, é um desafio determinar se um provedor de serviço está cumprindo o nível de QoS desejado pelo usuário. Para resolver esse problema, propõe-se o estabelecimento de acordos, como os Acordos de Nível de Serviço (SLAs), entre os usuários e os provedores (AMAZON WEB SERVICES, 2023).

No nível de aplicação, o gerenciamento de QoS requer o mapeamento e a negociação dos requisitos, o estabelecimento de SLAs e a monitoração contínua. O mapeamento dos melhores atributos de QoS não é uma tarefa simples e depende dos objetivos do usuário. Os usuários podem solicitar determinados níveis de qualidade que o provedor pode não estar preparado para atender, como uma disponibilidade de 99,99%, o que é considerada ótima, segundo o mesmo autor. Existem vários atributos de qualidade de serviço que podem ser considerados, como tempo de resposta, taxa de transferência, interoperabilidade e confiabilidade. A confiabilidade, por exemplo, envolve propriedades como garantir a entrega correta dos dados, a disponibilidade do sistema em determinado momento e a segurança e proteção dos dados contra falhas e ameaças maliciosas.

Uma vez que os atributos de QoS foram identificados, é necessário determinar os níveis desses atributos. Isso requer uma negociação entre os usuários e os provedores de serviços. Normalmente, é estabelecido um SLA que define os níveis desejados pelo consumidor e os serviços que serão fornecidos pelo provedor, juntamente com os custos associados.

No estado atual da computação em nuvem, há uma falta de suporte ou limitações na dinâmica de negociação de SLAs. Além disso, o fornecimento de serviços sob demanda na nuvem complica ainda mais a entrega de QoS, pois a alocação dinâmica de recursos deve levar em consideração a quantidade de recursos necessários para manter o nível de QoS estabelecido no SLA. Para lidar com essas questões, é necessário um monitoramento contínuo da QoS (AMAZON WEB SERVICES, 2023).

Além dos atributos de qualidade de serviço, também é importante considerar o prazo e o orçamento. Em sistemas de tempo real, o requisito de QoS está relacionado a qualidade de serviço (QoS) desempenha um papel crucial na área de redes e computação, buscando garantir a eficiência e a confiabilidade dos serviços oferecidos. A medição da latência é um método comumente utilizado para avaliar a QoS, envolvendo a utilização de pontos de referência para medir a latência entre os nós que solicitam um serviço e os nós que o fornecem (LOPES e BAQUERO, 2003). Em ambientes de computação em nuvem, o uso de protocolos baseados em DHT pode contribuir para a redução da latência na busca de informações, uma vez que esses protocolos possuem heurísticas que selecionam nós próximos uns dos outros, diminuindo a latência de comunicação entre eles (BASET; SCHULZRINNE; SHIM, 2006).

A QoS pode ser definida como a percepção do usuário em relação à eficiência de um determinado serviço. Ela engloba a garantia de requisitos como perdas mínimas, desempenho máximo, tempo de resposta reduzido e a precisão e consistência dos dados recebidos (KUMAR et al., 2014). No entanto, oferecer uma QoS adequada não é uma tarefa trivial, especialmente considerando a natureza dinâmica e os desafios impostos pela rede da internet. Além disso, determinar se um provedor de serviços atende ao nível de QoS desejado pelo usuário é um problema em si. Para solucionar essa questão, propõe-se o estabelecimento de acordos, como os Acordos de Nível de Serviço (SLAs), entre os usuários e os provedores de serviços.

No contexto da aplicação, o gerenciamento de QoS deve abranger requisitos como mapeamento e negociação da QoS, estabelecimento de SLAs e monitoramento contínuo (GRANVILLE, 2001). O mapeamento dos melhores atributos de QoS não é uma tarefa simples e depende dos objetivos do usuário. Muitas vezes, os usuários podem solicitar níveis de qualidade que o provedor não está preparado para fornecer, como uma disponibilidade de 99,99%. Existem diversos atributos de qualidade de serviço que podem ser considerados, como tempo de resposta, taxa de transferência, interoperabilidade e confiabilidade. A confiabilidade, por exemplo, abrange propriedades como a capacidade de garantir a entrega correta dos dados, a disponibilidade do sistema em momentos específicos, além da segurança e proteção dos dados contra falhas, ameaças maliciosas e erros (LOPES e BAQUERO, 2003).

Uma vez que os atributos de QoS foram mapeados, surge a necessidade de determinar os níveis desses atributos. Nesse ponto, a negociação entre os usuários e os provedores de serviços se torna essencial. Normalmente, essa negociação resulta na elaboração de um acordo de nível de serviço (SLA), que define os níveis desejados pelo consumidor e os serviços que serão fornecidos pelo provedor a um custo específico.

No estado atual da computação em nuvem, o suporte à dinâmica de negociação de SLAs ainda é limitado (VACCARO, 1997). Além disso, a oferta de serviços. A garantia de uma qualidade mínima nos serviços oferecidos por sistemas computacionais tornou-se um tema cada vez mais relevante tanto no âmbito acadêmico quanto na indústria. No contexto das tecnologias de rede, a qualidade de serviço (QoS) desempenha um papel fundamental em todos os níveis da arquitetura de protocolos, com avanços significativos sendo realizados nas camadas inferiores. No entanto, na camada de aplicação, os esforços nesse sentido são mais recentes.

As redes IPv6 constituem a infraestrutura de milhares de redes de computadores em todo o mundo, com uma expectativa constante de crescimento. A disseminação desse padrão é resultado da rápida expansão da Internet e da adoção do protocolo TCP/IP como suporte para aplicações em redes. O predomínio de computadores que utilizam o TCP/IP em suas comunicações tornou o IP um padrão universal de suporte para aplicações na Internet, estando presente em milhões de máquinas em todo o mundo.

As redes IP foram desenvolvidas com requisitos de confiabilidade pouco exigentes e com o objetivo de serem utilizadas em diversos tipos de meios físicos. Devido à sua simplicidade, o IP apresenta algumas restrições, como a falta de garantias no trânsito de pacotes e atrasos. Com o aumento das aplicações, como a transmissão de voz e multimídia, torna-se necessário abordar aspectos de QoS mais rigorosos.

Na área da computação em nuvem, surgiu a necessidade de utilização de serviços entre os participantes desse novo ambiente, criando uma organização colaborativa de computação em nuvem. Os provedores de computação em nuvem podem estar localizados geograficamente em diferentes pontos da Internet, buscando fornecer serviços de melhor qualidade aos clientes, incluindo aspectos como segurança, confiabilidade e tolerância a falhas (SHARMA; GUPTA; ACHARYA, 2020).

Nesse contexto, é fundamental que a comunicação entre as entidades colaborativas seja acompanhada de características que garantam aspectos de QoS. Os usuários de computação em nuvem precisam de mecanismos para identificar os participantes capazes de atender às suas necessidades em relação às métricas de QoS.

Os provedores de computação em nuvem devem implementar mecanismos e políticas para distribuir a carga de trabalho entre diferentes data centers, a fim de determinar a localização ideal para hospedar os serviços e alcançar níveis razoáveis de QoS. Em organizações de computação em nuvem, a interconexão entre os diversos provedores pode ser realizada utilizando uma rede *peer-to-peer* (P2P). No entanto, é essencial que o sistema P2P seja bem definido para garantir um desempenho adequado durante as chamadas de comunicação entre as entidades.

Em resumo, a QoS desempenha um papel fundamental na garantia da qualidade e eficiência dos serviços em redes e sistemas computacionais. No contexto da computação em nuvem, é necessário um esforço contínuo para aprimorar a QoS em todos os níveis, tanto na aplicação quanto na infraestrutura de rede, visando atender às demandas dos usuários e proporcionar uma experiência satisfatória no uso dos serviços.

## 2.5. LÓGICA FUZZY

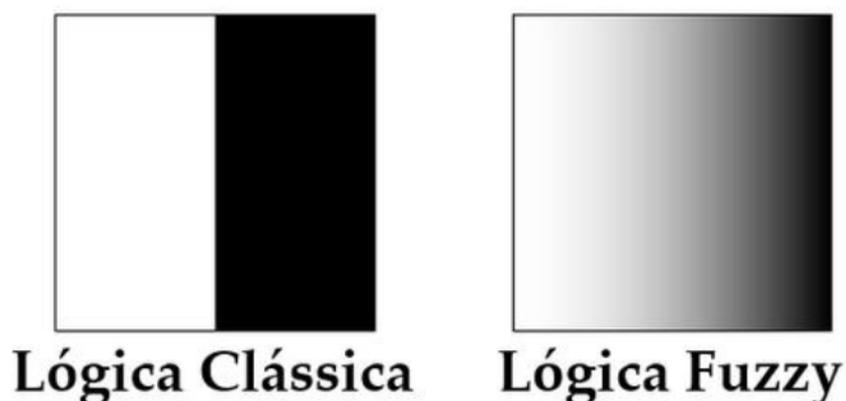
No contexto de automação de decisões que possuem natureza ambígua, surge a necessidade de desenvolver, implementar e testar aplicações capazes de lidar com estímulos complexos de forma adequada. Como podemos enfrentar o desafio de simular o comportamento de um motorista de carro ou analisar dados provenientes de redes sociais para compreender conflitos de ideias? Essas situações exigem habilidades de interpretação subjetiva e lidam com informações que não podem ser quantificadas com precisão por valores numéricos.

A lógica *fuzzy*, proposta inicialmente em 1965 pelo professor Lofti Zadeh da Universidade da Califórnia, Estados Unidos, surge como uma solução para lidar com a ambiguidade e a incerteza inerentes a essas situações. Zadeh introduziu o conceito de variáveis linguísticas e os comparou a conjuntos *fuzzy* (ou *Fuzzy Sets*).

Segundo Moré (2004), grande parte da linguagem natural contém ambiguidades e múltiplos significados. Adjetivos, por exemplo, frequentemente levam a interpretações subjetivas e imprecisas. Por exemplo, quando dizemos que alguém é "alto", não é possível afirmar de forma definitiva quem é alto e quem não é, pois a altura é uma característica relativa. Além disso, em áreas como engenharia, adjetivos que descrevem estados ou condições geralmente estão relacionados a quantidades. Assim, adjetivos como "alto" ou "idoso" podem ser dimensionados e quantificados.

Em contraste com a lógica clássica, conforme ilustração da Figura 2, em que uma proposição é verdadeira ou falsa, a lógica *fuzzy* permite que uma proposição seja parcialmente verdadeira ou parcialmente falsa. Cada elemento de um conjunto *fuzzy* possui um grau de pertinência definido no intervalo de 0 a 1, indicando o quão compatível o elemento é com o conjunto (BOENTE, 2013).

Figura 2 - Comparativo entre a lógica clássica e a lógica *fuzzy*



Fonte: Boente, 2013.

A lógica *fuzzy* representa o conhecimento comum em uma linguagem matemática, utilizando a teoria de conjuntos *fuzzy* e suas funções de pertinência associadas.

Os números *fuzzy* são usados para quantificar atributos físicos imprecisos ou conceitos vagos. Eles representam conjuntos difusos normalizados e convexos no conjunto dos números reais, utilizando funções de pertinência. Os números *fuzzy* permitem uma representação intuitiva de números ou intervalos aproximados. Eles desempenham um papel fundamental na determinação dos estados das variáveis difusas e são aplicados em controle *fuzzy*, raciocínio aproximado, tomada de decisão e estatística.

As variáveis linguísticas são formalizadas como conjuntos *fuzzy* discretos que consistem no nome do conjunto, nos valores linguísticos associados a eles e nas funções de pertinência correspondentes. Essas variáveis são definidas em um universo de discurso e representadas por uma quintupla que inclui o nome da variável, o conjunto de termos linguísticos, o conjunto universo, a regra sintática para geração dos termos e a regra semântica associada a cada termo.

Existem diferentes tipos de conjuntos *fuzzy*, como triangular, gaussiano e trapezoidal, que são utilizados de acordo com a necessidade e a aplicação específica. No trabalho proposto, são empregados números difusos triangulares devido à sua adaptação adequada à resolução do problema em questão (BOJADZIEV; BOJADZIEV, 2007).

Os conjuntos *fuzzy* permitem a representação de conceitos vagos e imprecisos, expressos na linguagem natural, e sua utilização depende do contexto em que são

aplicados. Eles possibilitam que os elementos pertençam parcialmente a diferentes conjuntos *fuzzy* dentro do mesmo universo.

Segundo Cosenza *et al.* (2006), o modelo matemático foi estruturado de forma a avaliar alternativas de localização empregando lógica *fuzzy*. As variáveis linguísticas utilizadas foram definidas de forma a permitir ao tomador de decisões uma maior afinidade com as hierarquias estabelecidas para operar neste ambiente *fuzzy*. O primeiro passo é confrontar as situações de demanda industrial e as de oferta territorial de fatores gerais (basicamente infraestrutura).

Em suma, a lógica *fuzzy* fornece uma abordagem matemática para lidar com a incerteza e a ambiguidade na tomada de decisões. Ela incorpora a natureza da linguagem natural, permitindo a representação e a manipulação de conhecimento complexo. Ao utilizar variáveis linguísticas, conjuntos *fuzzy* e funções de pertinência, a lógica *fuzzy* oferece uma estrutura formal para lidar com a interpretação subjetiva e a incerteza inerentes a muitos problemas do mundo real.

### 2.5.1. Python

Python, uma linguagem de programação notável por sua sintaxe clara e concisa, promove a legibilidade e a eficiência na escrita de código-fonte. Esta linguagem incorpora estruturas de dados de alto nível, como listas e dicionários, e uma extensa biblioteca de módulos prontos para uso, além de suportar a adição de frameworks externos. Com características de linguagens modernas, como geradores e introspecção, Python é uma linguagem multiparadigma, suportando programação modular, funcional e orientada a objetos. Sua natureza interpretada, operando via *bytecode* em uma máquina virtual Python, confere portabilidade ao código. Python, sendo um software de código aberto sob uma licença menos restritiva que a GPL, permite sua incorporação em produtos proprietários. A Python Software Foundation mantém a especificação da linguagem. Utilizada amplamente tanto como linguagem principal quanto em scripts para softwares diversos, Python oferece automação e extensibilidade. Sua capacidade de integração com outras linguagens, como C e Fortran, e semelhanças com outras linguagens dinâmicas como Perl e Ruby, destacam sua versatilidade e aplicabilidade em diversos contextos de desenvolvimento de *software*.

De acordo com Rossum (2015), A linguagem Python foi desenvolvida em 1990 por Guido van Rossum no Instituto Nacional de Pesquisa para Matemática e Ciência da

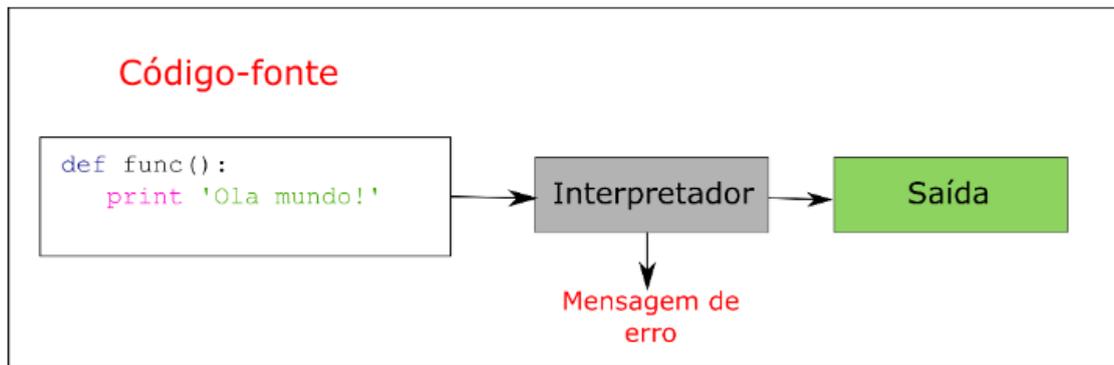
Computação da Holanda (CWI), inicialmente direcionada para uso em campos como física e engenharia. Originária da linguagem ABC, Python evoluiu para se tornar uma escolha popular em diversos setores da indústria de alta tecnologia, sendo adotada por empresas como Google, Yahoo, Microsoft (com o IronPython para .NET), Nokia e Disney para aplicações que variam de web a animações 3D. Python é mantido pela Python Software Foundation (PSF) e sua implementação oficial, escrita em C e conhecida como CPython, está disponível para download em várias plataformas. Além do CPython, existem implementações para .NET (IronPython), JVM (Jython) e Python (PyPy), demonstrando sua flexibilidade e adaptabilidade a diferentes ambientes de desenvolvimento.

Atualmente, a linguagem é bem aceita na indústria por empresas de alta tecnologia, tais como:

- Google (aplicações web e sistemas internos);
- Facebook (back-end e processamento de dados);
- Instagram (gerenciamento da sua plataforma de mídia social);
- Spotify (Utiliza principalmente para análise de dados e back-end); e
- Netflix (Aplica para análise de dados, segurança e servidor back-end).

Na esfera das linguagens de programação de alto nível, como o Python, existe a necessidade de um mecanismo intermediário para traduzir o código em linguagem de baixo nível, adequada para execução por hardware. Esses mecanismos são geralmente categorizados em dois tipos: interpretadores e compiladores. Especificamente, Python é classificado como uma linguagem interpretada (Borges, 2010), o que implica que seu código é executado através de um processo iterativo de leitura, interpretação e execução de instruções, conforme ilustrado na Figura 3.

Figura 3 - Etapas da construção de uma aplicação



**Fonte: Borges (2010).**

Este método de execução oferece flexibilidade e facilita a depuração, embora possa impactar a velocidade de execução quando comparado aos compiladores.

Basicamente, trata-se de uma linguagem de programação versátil, capaz de desenvolver uma ampla gama de sistemas. Sua aplicabilidade se estende desde a coleta de dados de formulários online até a geração de páginas web dinâmicas.

Uma característica distintiva de Python é seu robusto suporte a uma extensa variedade de sistemas de gerenciamento de banco de dados, incluindo, mas não se limitando a, dBase, mSQL, Interbase, SQL Server, MySQL, Sybase, Oracle e PostgreSQL. Esta compatibilidade com múltiplos bancos de dados facilita significativamente a integração de dados em aplicações baseadas na web, tornando a construção de páginas web baseadas em banco de dados uma tarefa simplificada e eficiente com o uso de Python.

### 3. APLICAÇÃO DO MODELO FUZZY

O presente estudo foi concebido no âmbito do Programa de Engenharia de Produção da COPPE, na Universidade Federal do Rio de Janeiro, especificamente na Área de Engenharia de Decisão e Gestão. Este ambiente acadêmico estimulante propiciou a elaboração de projetos focados na lógica *fuzzy*, contando com a orientação e participação ativa dos orientadores. O objetivo principal da pesquisa era abordar e solucionar uma problemática prática e relevante no setor de serviços tecnológicos, propondo uma solução inovadora e eficaz que se alinhasse às condições e exigências específicas do problema identificado.

Na atualidade, os impasses relacionados ao setor de serviços são considerados desafiadores, especificamente os serviços referentes a computação em nuvem que é o foco da formulação do problema. Eles envolvem diversas variáveis, diversos aspectos a serem analisados, como qualidade, eficiência, capacidade, entre outros.

Em ambientes de computação em nuvem, a alocação de recursos e a otimização do desempenho são desafios significativos devido à natureza dinâmica e heterogênea das aplicações. As técnicas tradicionais de caracterização de aplicações podem não ser suficientemente flexíveis ou precisas para lidar com a incerteza e a variabilidade inerentes a esses ambientes. Portanto, surge a necessidade de um modelo que possa caracterizar aplicações em ambientes de computação na nuvem de forma mais eficaz, levando em conta a incerteza e a variabilidade das demandas de recursos e comportamento das aplicações.

Para esta dissertação, o estudo do problema envolve os serviços de computação em nuvem da Microsoft Azure<sup>8</sup> que reflete parte integrante do portfólio de serviços da Microsoft, representa uma plataforma abrangente de computação em nuvem, que oferece uma variedade de serviços e soluções para empresas e indivíduos. Seu desenvolvimento e expansão refletem uma resposta direta à crescente demanda por soluções de computação em nuvem flexíveis, escaláveis e eficientes.

Lançada inicialmente em 2010, conforme afirma (TWDI, 2021), a Microsoft Azure evoluiu de uma plataforma de serviços em nuvem simples para um ecossistema robusto e diversificado. Inicialmente focada em serviços de PaaS (*Platform as a Service*), ela expandiu rapidamente suas ofertas para incluir soluções IaaS (*Infrastructure as a Service*) e SaaS (*Software as a Service*). Essa expansão reflete uma

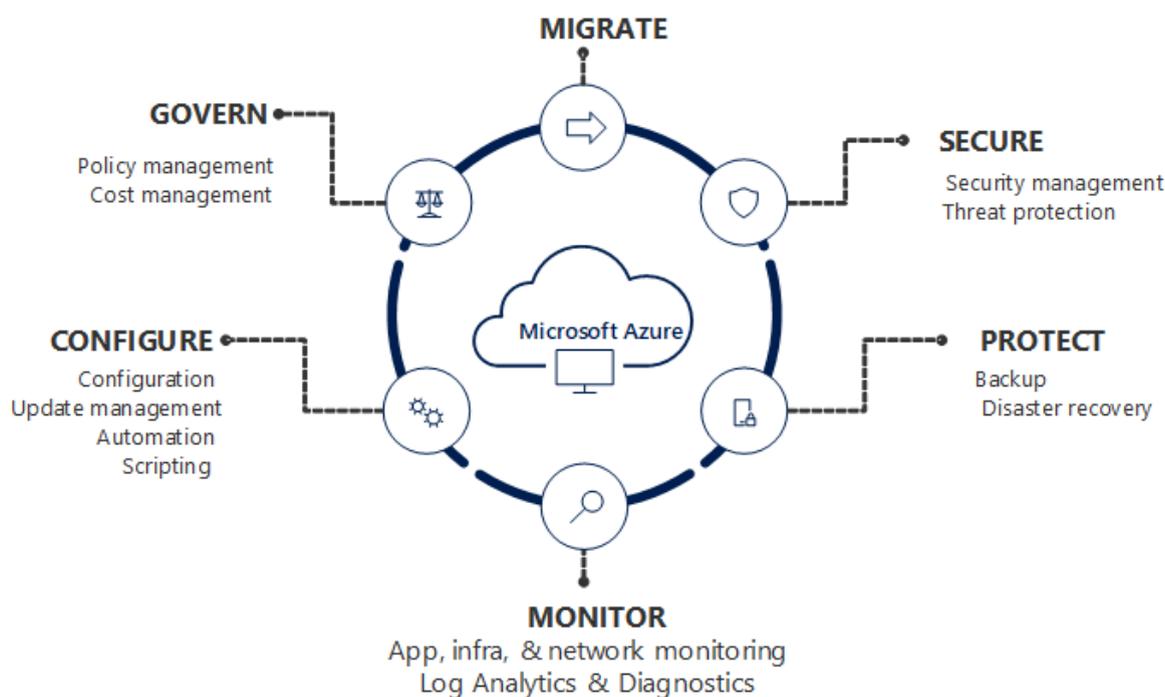
---

<sup>8</sup> Microsoft Azure - Plataforma abrangente de computação em nuvem criada pela Microsoft

adaptação às necessidades em constante mudança do mercado de TI, bem como uma resposta aos concorrentes, como Amazon Web Services (AWS) e Google Cloud Platform (GCP).

Pode-se observar com detalhes, as áreas de conhecimento de gerenciamento do Azure por meio da Figura 4.

Figura 4 - Área de gerenciamento do Azure



Fonte: (DAVIDSMATLAK, 2023).

A plataforma Microsoft Azure oferece uma série de serviços integrados, cada um contribuindo de forma única para atender a requisitos específicos de gerenciamento de TI. Estes serviços, ao operarem em conjunto, fornecem uma solução abrangente, embora nenhum serviço individualmente atenda completamente a todas as necessidades de uma área de gerenciamento específica.

### 3.1. MONITORAMENTO

O monitoramento envolve a coleta e análise de dados para avaliar o desempenho, a saúde e a disponibilidade de recursos. Uma abordagem efetiva de monitoramento é essencial para compreender as operações dos componentes e otimizar a disponibilidade, através de notificações proativas. Segundo (STEPHENSON, 2023),

dentre os serviços do Azure para essa função, destacam-se o *Application Insights*, para monitoramento focado em aplicações web, e o Azure Monitor Logs, que armazena dados de gerenciamento para uso em análises diversas.

### 3.2. CONFIGURAÇÃO

A configuração diz respeito à implantação e ajuste inicial de recursos, bem como à sua manutenção contínua. A automação destas tarefas é fundamental para reduzir redundâncias, melhorar a precisão e eficiência e minimizar esforços manuais. Neste contexto, o Azure Automation oferece um conjunto abrangente de ferramentas para automatizar tarefas de configuração, com *runbooks* para automação de processos e ferramentas para gerenciamento de atualizações e configurações (KAUR, 2023).

### 3.3. ADMINISTRAÇÃO

A governança, dentro do contexto do Azure, se refere à implementação de mecanismos e processos para exercer controle efetivo sobre aplicativos e recursos. Isso inclui o planejamento de iniciativas e a definição de prioridades estratégicas. Serviços como o Azure *Policy*, que permite a criação, atribuição e gestão de políticas para regulamentar recursos, e o Gerenciamento de Custos do Azure, que monitora o uso e os custos dos recursos, são essenciais para uma governança eficaz (MICROSOFT, 2023).

### 3.4. SEGURANÇA

Gerenciar a segurança dos recursos e dados é uma prioridade no Azure. Este aspecto envolve a avaliação de ameaças, a coleta e análise de dados de segurança e a garantia da conformidade de aplicativos e recursos. O Microsoft Defender para Nuvem é um serviço chave, oferecendo monitoramento de segurança e análise de ameaças, juntamente com gerenciamento de segurança unificado e proteção avançada contra ameaças para cargas de trabalho em nuvens híbridas (MICROSOFT, 2023).

### 3.5. PROTEÇÃO

A proteção no Azure foca em manter aplicativos e dados acessíveis e seguros, mesmo diante de interrupções imprevistas. Isso é assegurado por serviços como o Backup do Azure, que oferece soluções de backup e recuperação, e o Azure Site

*Recovery*, que proporciona continuidade dos negócios e recuperação de desastres (MICROSOFT, 2023).

### 3.6. MIGRAÇÃO

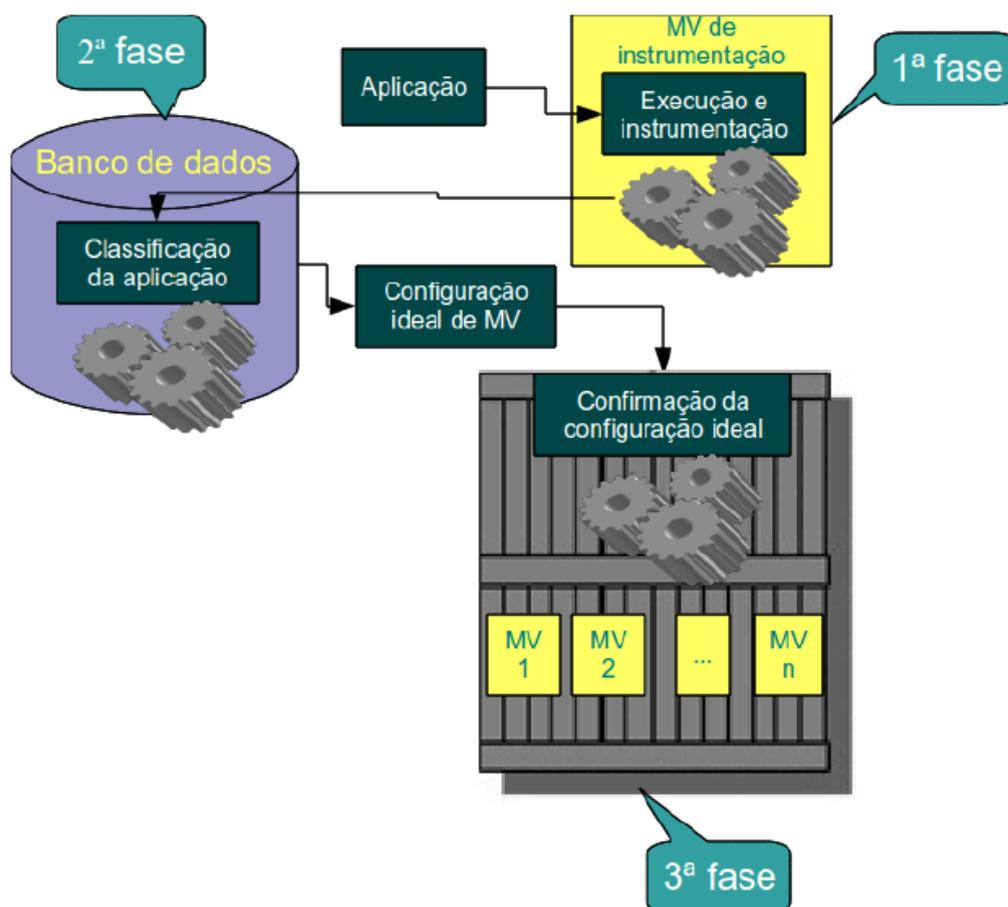
Migração envolve a transição de cargas de trabalho de ambientes locais para a nuvem do Azure. Para isso, a Azure oferece serviços como o Azure Site *Recovery*, que facilita a migração de máquinas virtuais, e o Serviço de Migração de Banco de Dados do Azure, que auxilia na transição de fontes de banco de dados para plataformas do Azure Data. Cada um desses serviços desempenha um papel vital na gestão eficaz de recursos e aplicações na plataforma Azure, demonstrando a abordagem holística e integrada da Microsoft na oferta de soluções de computação em nuvem.

### 3.7. METODOLOGIA ADOTADA

A metodologia adotada neste trabalho, conforme ilustra a Figura 5, tem o objetivo de apresentar uma solução de caracterização de aplicações por intermédio de um plano de instrumentação e classificação da performance de recursos computacionais em ambientes de máquinas virtuais tratando as subjetividades, imprecisões e possíveis ambiguidades.

A partir de certa aplicação, na fase 1, denominada máquina virtual (MV) de instrumentação, ocorre a execução e a instrumentação dos dados que, em seguida, faz decorrer a classificação da aplicação, onde seus dados são armazenados em banco de dados, na fase 2. Após esse processo, na fase 3, ocorre a confirmação da configuração ideal de máquina virtual, gerando inúmeras instâncias, ora classificadas como  $MV_1$ ,  $MV_2$ , ...,  $MV_n$ .

Figura 5 - Metodologia adotada

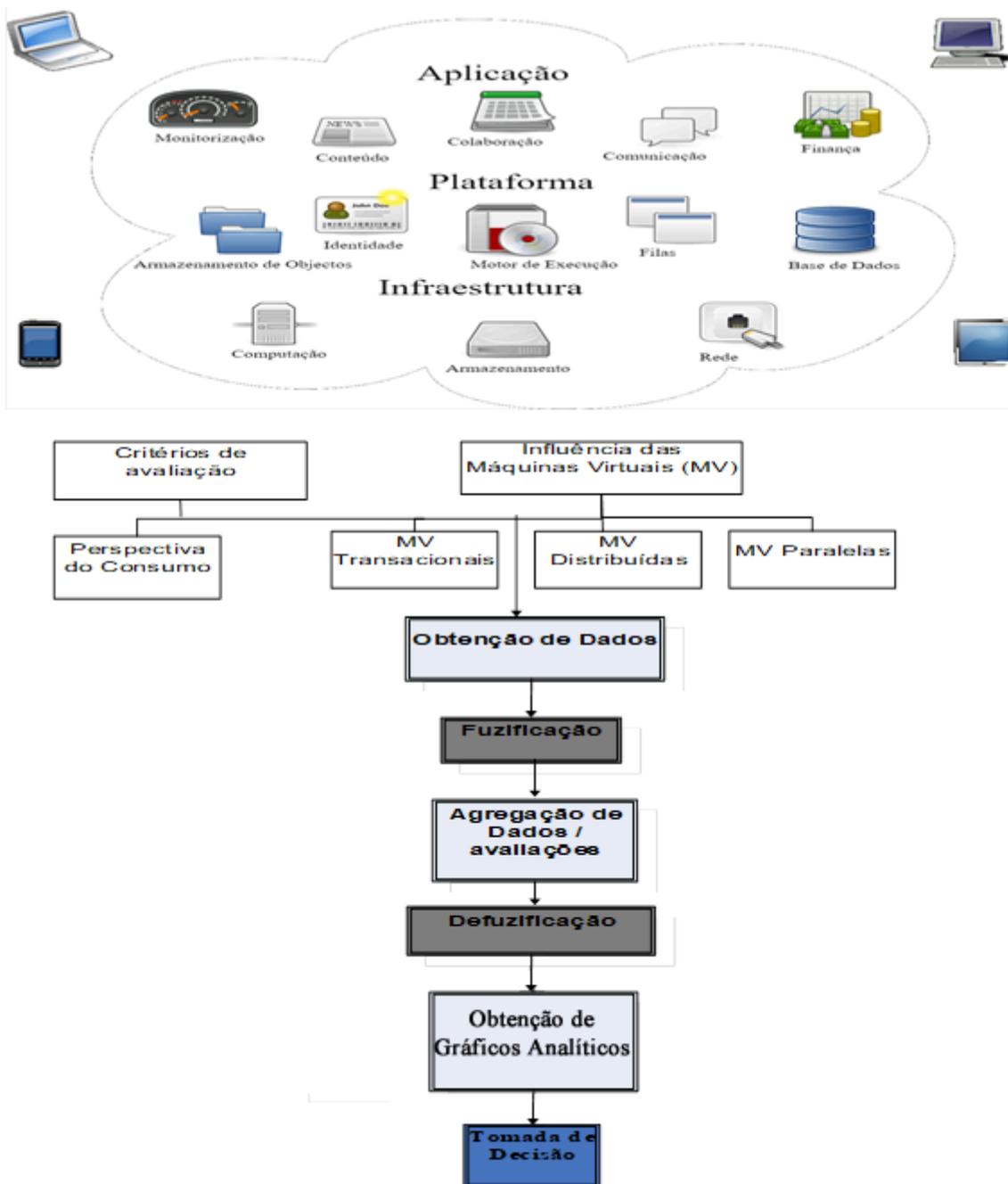


Fonte: Ogura, 2011.

Esses modelos viabilizam a configuração de cenários e a classificação de distintos conjuntos para uma ampla gama de aplicações, conseqüentemente, essa estrutura pode ser aprimorada à medida que o número de aplicações integradas aumenta, possibilitando a inclusão de outras variáveis comparativas para uma categorização mais precisa dos conjuntos (OGURA, 2011). Isso permite a identificação da configuração mais adequada ou recomendada para cada aplicação.

### 3.8. APLICAÇÃO DO MODELO

Para a aplicação do modelo proposto, foi necessário utilizar seis (06) etapas conforme descrição do modelo *fuzzy* proposto a seguir, ilustrado na Figura 6:

Figura 6 - Descrição do modelo *fuzzy*

Fonte: Elaboração Própria.

### 3.8.1. Obtenção de Dados

Considerando que o comportamento e características de consumo de CPU, disco e a influência do monitor de máquinas virtuais em aplicações distribuídas,

paralelas e transacionais apresentam aspectos de representação imprecisos, inerentes às características subjetivas como ambiguidade, relativização, imprecisões e incoerências, sua avaliação torna-se preditivamente complexa. Propomos, portanto, o uso da teoria dos conjuntos *fuzzy* para a avaliação da perspectiva de consumo e influência das máquinas virtuais. Espera-se que este estudo possa fundamentar a proposta de um modelo que permita a obtenção de gráficos analíticos para tomada de decisão.

### 3.8.2. Determinação das variáveis linguísticas do modelo

Nesta fase do estudo, as variáveis linguísticas foram estabelecidas com base em uma revisão bibliográfica extensiva, que possibilitou a identificação de 15 construtos fundamentais relacionados à qualidade e importância dos serviços de computação em nuvem. Esses construtos incluem aspectos como desempenho do sistema, avaliação de eficiência, autoavaliação de desempenho da aplicação, segurança de dados, gestão de recursos computacionais, coordenação e gerenciamento de serviços na nuvem, suporte técnico, administração de sistemas e infraestrutura de rede. Além disso, foram considerados 6 construtos adicionais ligados à satisfação do usuário, abrangendo a qualidade percebida dos serviços, expectativas dos usuários, valor percebido, satisfação do usuário, feedbacks e reclamações dos usuários e lealdade ou fidelidade do usuário aos serviços de computação em nuvem.

### 3.8.3. Escolha dos termos linguísticos a serem utilizados

Para aferir a qualidade dos serviços em nuvem e a presença dos critérios de utilização da CPU, foram escolhidos 3 termos linguísticos, baixa, média e alta, conforme ilustrado no Quadro 3.

Quadro 3 - Função de Pertinência da variável linguística CPU

<b>Termos Linguísticos</b>	<b>Função de Pertinência</b>	<b>Descrição</b>
Baixa	0 a 50	Carga de CPU considerada baixa.
Média	25 a 75	Carga de CPU considerada média.
Alta	50 a 100	Carga de CPU considerada alta.

**Fonte: Elaboração própria.**

**Baixa** (0 a 50): Este conjunto representa uma carga de CPU baixa, variando de 0 a 50. Na lógica *fuzzy*, a CPU com valor menor ou igual a 50 de carga é considerada como operando em um estado 'baixo' de utilização. Esta categoria é útil para identificar situações em que a CPU está subutilizada.

**Média** (25 a 75): Este conjunto indica uma carga de CPU média, abrangendo uma zona de transição onde a CPU não está nem subutilizada nem sobrecarregada. É uma faixa de operação comum para muitas aplicações.

**Alta** (50 a 100): Este conjunto representa uma alta carga de CPU, variando de 50 a 100. Na lógica *fuzzy*, isso indica uma utilização intensa da CPU, onde a ela pode estar se aproximando ou atingindo sua capacidade máxima.

Não se pode esquecer de um ponto primordial no contexto da computação em nuvem, a qualidade do serviço prestado ao cliente, pois através da SLA é possível melhorar, cada vez mais, os serviços de computação em nuvem oferecidos aos clientes.

Para aferir a qualidade dos serviços em nuvem e a presença dos critérios de utilização da memória das máquinas virtuais, foram escolhidos 3 termos linguísticos, baixa, média e alta, conforme ilustrado no Quadro 4.

Quadro 4 - Função de Pertinência da variável linguística Memória

<b>Termos Linguísticos</b>	<b>Função de Pertinência</b>	<b>Descrição</b>
Baixa	0 a 40	Uso de memória considerado baixo.
Média	30 a 70	Uso de memória considerado médio.
Alta	60 a 100	Uso de memória considerado alto.

**Fonte:** Elaboração própria.

**Baixa** (0 a 40): Este conjunto *fuzzy* indica baixo uso de memória. É útil para identificar quando há uma ampla disponibilidade de memória.

**Média** (30 a 70): Representa um uso médio de memória. Esta categoria é aplicada quando a memória está sendo utilizada de forma moderada.

**Alta** (60 a 100): Indica um alto uso de memória. Esta categoria é crucial para identificar situações de potencial sobrecarga de memória.

Para aferir a qualidade dos serviços em nuvem e a presença dos critérios da ação de escalonamento de recursos das máquinas virtuais, foram escolhidos 5 termos linguísticos, diminuir muito, diminuir pouco, manter, aumentar pouco e aumentar muito, conforme ilustrado no Quadro 5.

Quadro 5 - Função de pertinência da variável linguística Escalabilidade

<b>Termos Linguísticos</b>	<b>Função de Pertinência</b>	<b>Descrição</b>
Diminuir Muito	-10 a -5	Ação de escalonamento sugere diminuir significativamente os recursos.
Diminuir Pouco	-6 a 0	Ação de escalonamento sugere diminuir ligeiramente os recursos.
Manter	-1 a 1	Ação de escalonamento sugere manter os recursos atuais.
Aumentar Pouco	0 a 6	Ação de escalonamento sugere aumentar ligeiramente os recursos.
Aumentar Muito	5 a 10	Ação de escalonamento sugere aumentar significativamente os recursos.

**Fonte:** Elaboração própria.

**Diminuir Muito** (-10 a -5): Sugere uma redução significativa nos recursos. É aplicado em situações em que a demanda por recursos é muito baixa.

**Diminuir Pouco** (-6 a 0): Indica uma redução leve nos recursos, útil em ajustes finos para otimizar o desempenho.

**Manter** (-1 a 1): Implica na manutenção dos níveis atuais de recursos, indicando que os recursos atuais são adequados.

**Aumentar Pouco** (0 a 6): Sugere um aumento leve nos recursos, aplicável quando há uma necessidade moderada de mais recursos.

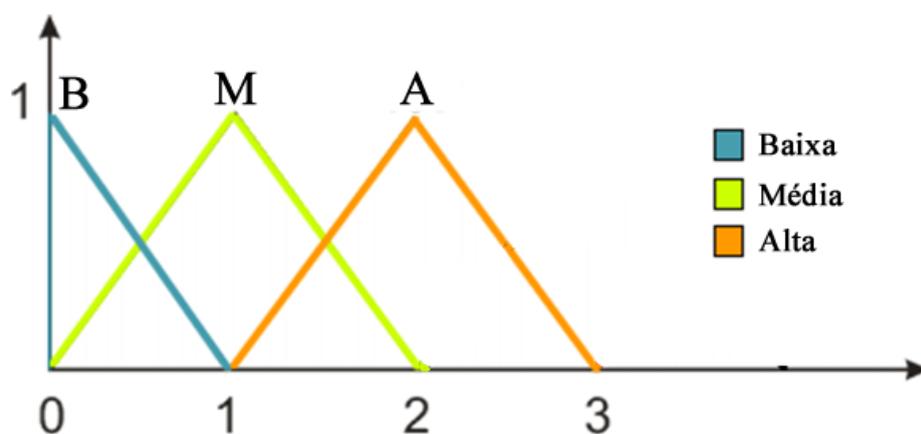
**Aumentar Muito** (5 a 10): Indica a necessidade de aumentar significativamente os recursos, geralmente em situações de alta demanda.

#### 3.8.4. Criação das funções de pertinências para os termos *fuzzy*.

Para representar as avaliações imprecisas e subjetivas dos dados de recursos em nuvem, tais como poder de processamento ou carga na CPU, memória e escalabilidade dos serviços foram escolhidos os conjuntos *fuzzy* triangulares pela capacidade que possuem de representar essa incerteza e para que seja computada através de um algoritmo facilmente pelo computador. Sua representação gráfica está ilustrada nas Figuras 7, 8, 9, 10, 11, 12 e 13, respectivamente.

Os conjuntos *fuzzy* correspondentes aos termos relacionados aos graus de utilização da CPU e memória das máquinas virtuais em nuvem, Baixa (B), Média (M) e Alta (A), estão ilustrados na Figura 7.

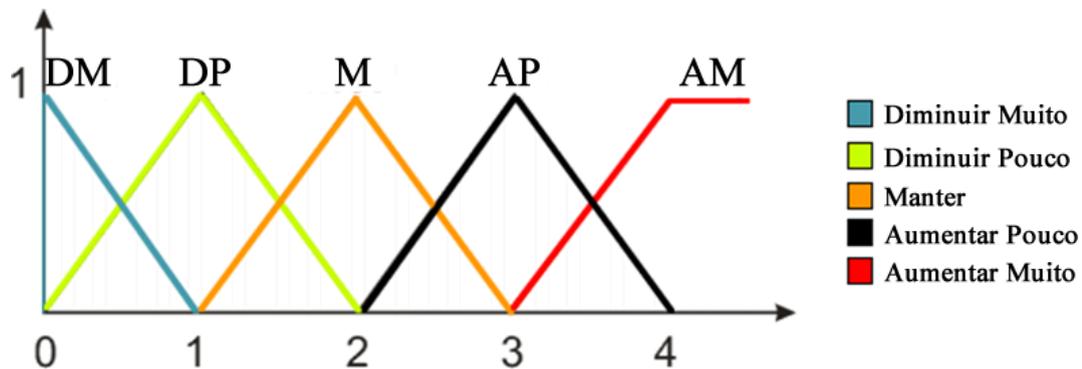
Figura 7 - Conjuntos *fuzzy* dos termos relacionados aos graus de utilização da CPU e memória das máquinas virtuais em computação em nuvem



Fonte: Adaptado de FERREIRA (2015).

Os conjuntos *fuzzy* correspondentes aos graus de escalonamento dos recursos e serviços em nuvem foram: Diminuir Muito (DM), Diminuir Pouco (DP), Manter (M), Aumentar Pouco (AP) e Aumentar Muito (AM), conforme ilustrado na Figura 8.

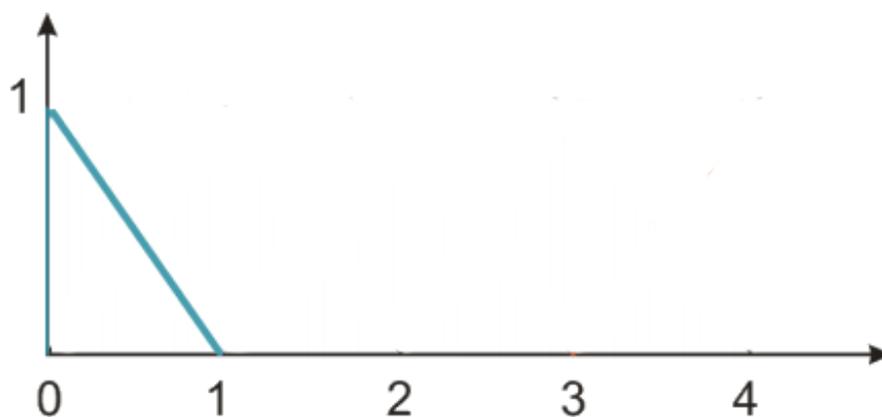
Figura 8 - Conjuntos *fuzzy* dos termos relacionados aos graus escalonamento de recursos das máquinas virtuais em computação em nuvem



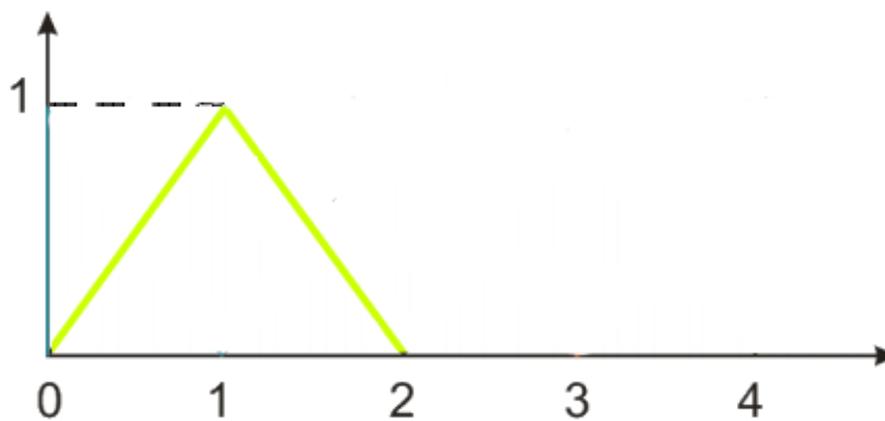
Fonte: Adaptado de FERREIRA (2015).

As Figuras 9, 10, 11, 12 e 13, ilustram os números triangulares *fuzzy* correspondente aos conjuntos *fuzzy* escolhidos nesta pesquisa.

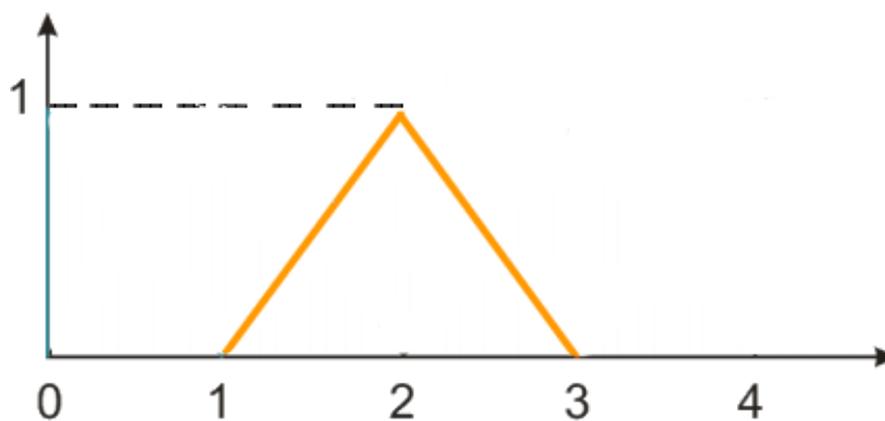
Figura 9 - Conjunto *fuzzy* triangular (0, 0, 1)



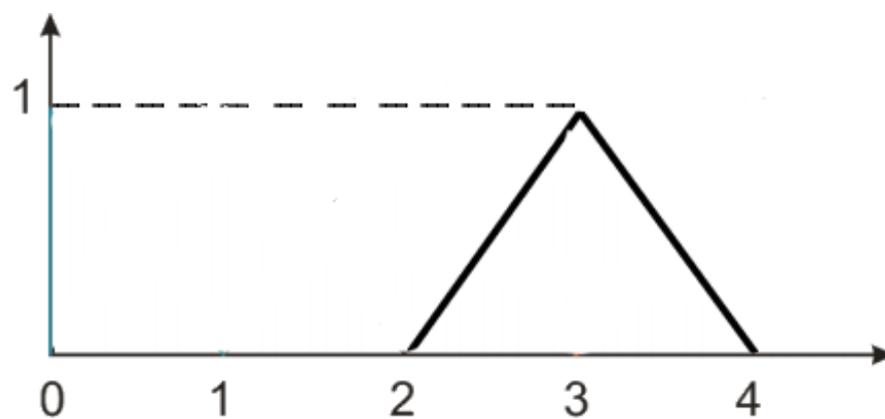
Fonte: Elaboração própria.

Figura 10 - Conjunto *fuzzy* triangular (0, 1, 2)

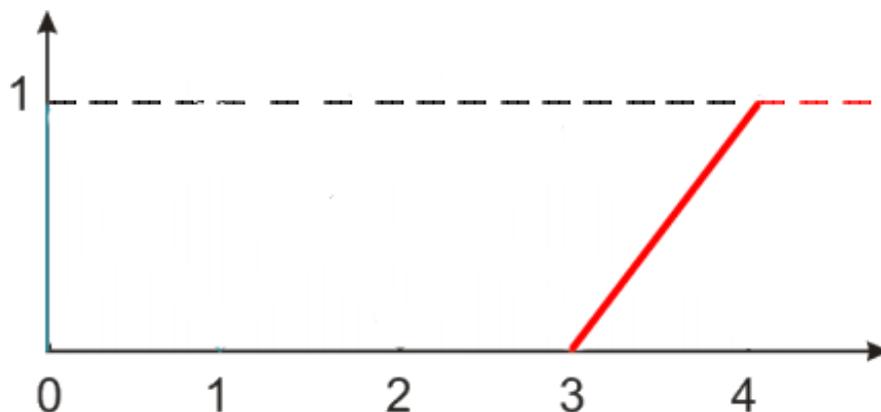
Fonte: Elaboração própria.

Figura 11 - Conjunto *fuzzy* triangular (1, 2, 3)

Fonte: Elaboração própria.

Figura 12 - Conjunto *fuzzy* triangular (2, 3, 4)

Fonte: Elaboração própria.

Figura 13 - Conjunto *fuzzy* triangular (3, 4, 4)

Fonte: Elaboração própria.

### 3.8.5. Aplicação dos questionários estruturados.

Nesta etapa, foi aplicado um questionário estruturado à amostra apresentada e baseada em 7 especialistas atuantes na utilização de serviços em nuvem, caracterizados pelos clientes que se utilizam dos serviços da computação em nuvem (ver Apêndice).

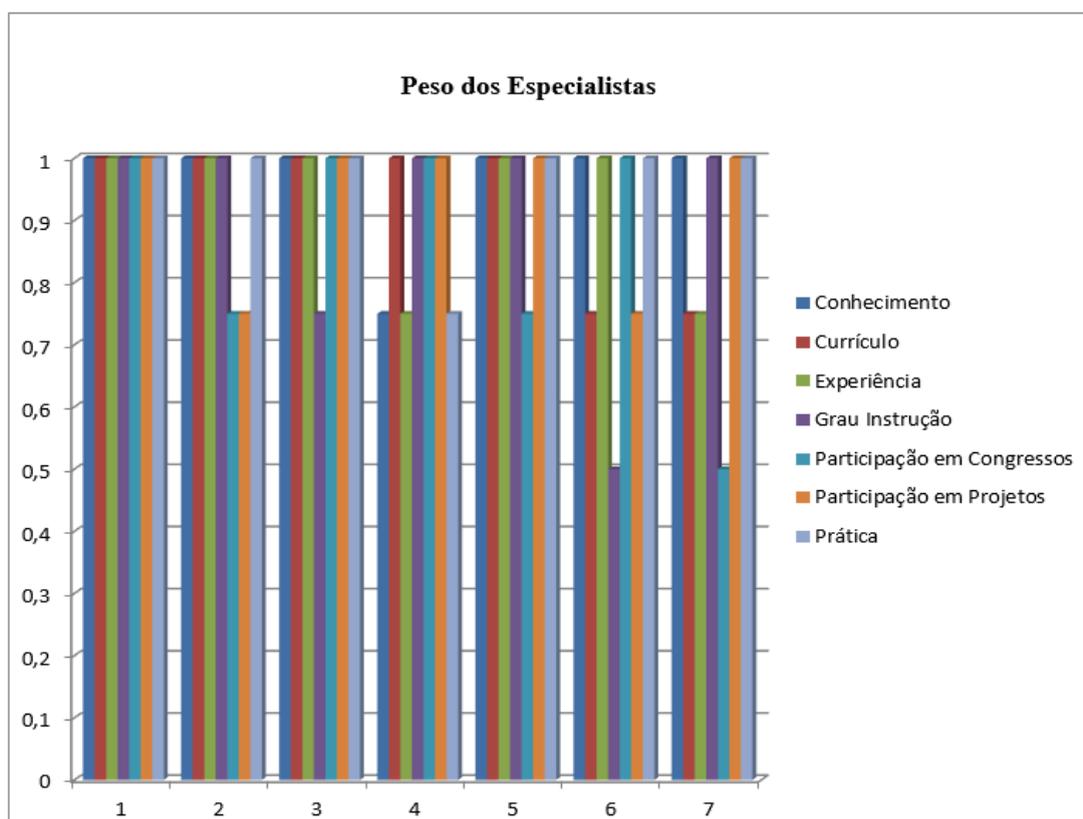
Quanto a identificação dos pesos dos especialistas, o coeficiente de importância e satisfação, foram utilizadas dicotomias de escalas de medição, atribuídas de acordo com os itens de interesse levantados (nível de conhecimento, currículo, nível de experiência, grau de instrução, participação em congressos, participação em projetos e prática de cada especialista). Cada item aferido obteve uma pontuação subjetiva entre 0 e 1. Conforme disponível no Tabela 1 e no Gráfico 1, são apresentados os valores dados a cada especialista distintos (E01, E02, E03, E04, E05, E06 e E7).

Tabela 1 - Peso dos especialistas

ITENS	E1	E2	E3	E4	E5	E6	E7	
Conhecimento	1,00	1,00	1,00	0,75	1,00	1,00	1,00	
Currículo	1,00	1,00	1,00	1,00	1,00	0,75	0,75	
Experiência	1,00	1,00	1,00	0,75	1,00	1,00	0,75	
Grau Instrução	1,00	1,00	0,75	1,00	1,00	0,50	1,00	
Participação em Congressos	1,00	0,75	1,00	1,00	0,75	1,00	0,50	
Participação em Projetos	1,00	0,75	1,00	1,00	1,00	0,75	1,00	
Prática	1,00	1,00	1,00	0,75	1,00	1,00	1,00	
<b>Σ Itens dos especialistas</b>	<b>7,00</b>	<b>6,50</b>	<b>6,75</b>	<b>6,25</b>	<b>6,75</b>	<b>6,00</b>	<b>6,00</b>	<b>Σ= 45,25</b>
<b>Peso dos especialistas</b>	<b>0,15</b>	<b>0,14</b>	<b>0,15</b>	<b>0,14</b>	<b>0,15</b>	<b>0,13</b>	<b>0,13</b>	<b>Σ= 1,00</b>

Fonte: Elaboração própria.

Gráfico 1 - Gráfico de barras referente ao peso dos especialistas



Fonte: Elaboração própria.

O coeficiente de satisfação e importância, referente aos pesos dos especialistas, foram calculados a partir da divisão e soma das pontuações subjetivas e recebidas por

item da soma total das pontuações recebidas de todos os respondentes, respectivamente ( $\Sigma=45,25$  e  $\Sigma=38,25$ ), de acordo com a fórmula ilustrada na Figura 14:

Figura 14 - Fórmula do somatório dos Itens e avaliações dos especialistas de serviços em nuvem

$$CI_{\text{especialistas}_i} = \frac{\sum_{i=1}^7 \text{Avaliações Itens Especialistas}_i}{\sum_{i=1}^7 \text{Avaliações Especialistas}}$$

Fonte: Elaboração própria.

### 3.8.6. Coleta e tabulação de dados.

O processo de coleta e tabulação de dados para a caracterização de aplicações em ambiente de computação na nuvem, particularmente quando a base de dados é obtida do *Kaggle*, envolve várias etapas críticas. Primeiramente, é necessário identificar e extrair conjuntos de dados relevantes do *Kaggle*<sup>9</sup>, que neste caso incluem métricas como identificação de VM, uso de CPU, uso de memória, tráfego de rede, consumo de energia, número de instruções executadas, tempo de execução, eficiência energética, tipo de tarefa, prioridade da tarefa e status da tarefa, esses dados precisam ser limpos e pré-processados para garantir a consistência e a qualidade dos dados, podendo incluir o tratamento de valores ausentes, normalização de formatos de data e hora, e a conversão de dados categóricos em formatos numéricos para análise.

Após a limpeza, a tabulação é realizada, organizando os dados de forma estruturada para facilitar a análise, pois envolve a criação de tabelas ou *dataframe*, gráficos e outras formas de representação visual que ajudam a identificar padrões e tendências.

Outrossim, a análise desses dados tabulados permite a caracterização de aplicações na nuvem, fornecendo *insights* sobre o desempenho, a eficiência e o comportamento das aplicações em diferentes condições, visto que essa análise pode ser

<sup>9</sup> *Kaggle* - é uma plataforma popular para cientistas de dados e pesquisadores que oferece uma grande variedade de conjuntos de dados, competições de *machine learning*, notebooks e discussões.

utilizada para otimizar recursos, melhorar a eficiência energética e informar decisões estratégicas em ambientes de computação na nuvem.

A agregação dos dados mediante arquivo importado na aplicação Python, é ilustrado na Figura 15:

Figura 15 - *Dataframe* ou tabela de dados obtida do *Kaggle*

▼ Base dados obtida do Kaggle

```
1 df = pd.read_csv('vmCloud_data.csv')
2 df.head(10)
```

	vm_id	timestamp	cpu_usage	memory_usage	network_traffic	power_consumption	num_executed_instructions	execution_time	energy_efficiency
0	c5215826-6237-4a33-9312-72c1df909881	2023-01-25 09:10:54	54.881350	78.950861	164.775973	287.808986	7527.0	69.345575	0.553589
1	29690bc6-1f34-403b-b509-a1ecb1834fb8	2023-01-26 04:46:34	71.518937	29.901883	NaN	362.273569	5348.0	41.396040	0.349856
2	2e55abc3-5bad-46cb-b445-a577f5e9bf2a	2023-01-13 23:39:47	NaN	92.709195	203.674847	231.467903	5483.0	24.602549	0.796277
3	e672e32f-c134-4fbc-992b-34eb63bef6bf	2023-02-09 11:45:49	54.488318	88.100960	NaN	195.639954	5876.0	16.456670	0.529511
4	f38b8b50-6926-4533-be4f-89ad11624071	2023-06-14 08:27:26	42.365480	NaN	NaN	359.451537	3361.0	55.307992	0.351907
5	ad14d5d1-4e96-48ca-9893-1baaa47e4cab	2023-02-06 16:44:41	64.589411	62.008016	580.569836	115.931610	4766.0	63.137671	NaN
6	5651505d-14a2-41d6-989b-ee0aac9b922	2023-06-15 18:04:04	43.758721	22.456429	429.139640	272.960417	9008.0	60.153904	0.461246
7	e5970e5c-0350-4131-bd6c-4bd7e2c98df1	NaN	NaN	85.438155	685.282813	NaN	NaN	14.191356	0.242538
8	7fc95f4c-f561-4dd5-8465-aec2b388bd5b	2023-05-20 19:09:26	96.366276	4.388441	902.827717	367.979056	9984.0	21.876694	NaN
9	7fd9a664-1da8-4ae9-8a04-3c341d32d35c	2023-07-10 11:32:51	38.344152	16.441881	779.791345	382.756696	2989.0	42.160847	0.139187

**Fonte: Elaboração própria.**

O problema apresentado nesta dissertação, objetiva apoiar a tomada de decisão multicritério baseada na utilização de técnicas de ciência de dados e inteligência artificial.

O problema ainda conta com algumas restrições, apresentadas a seguir:

- Para aferição dos dados obtidos no *Kaggle*, levou-se em conta que os serviços e recursos em nuvem são provenientes da Microsoft Azure.
- O foco da aferição dos recursos e serviços em nuvem, consistem no uso de memória, poder de processamento de CPU e escalabilidade.

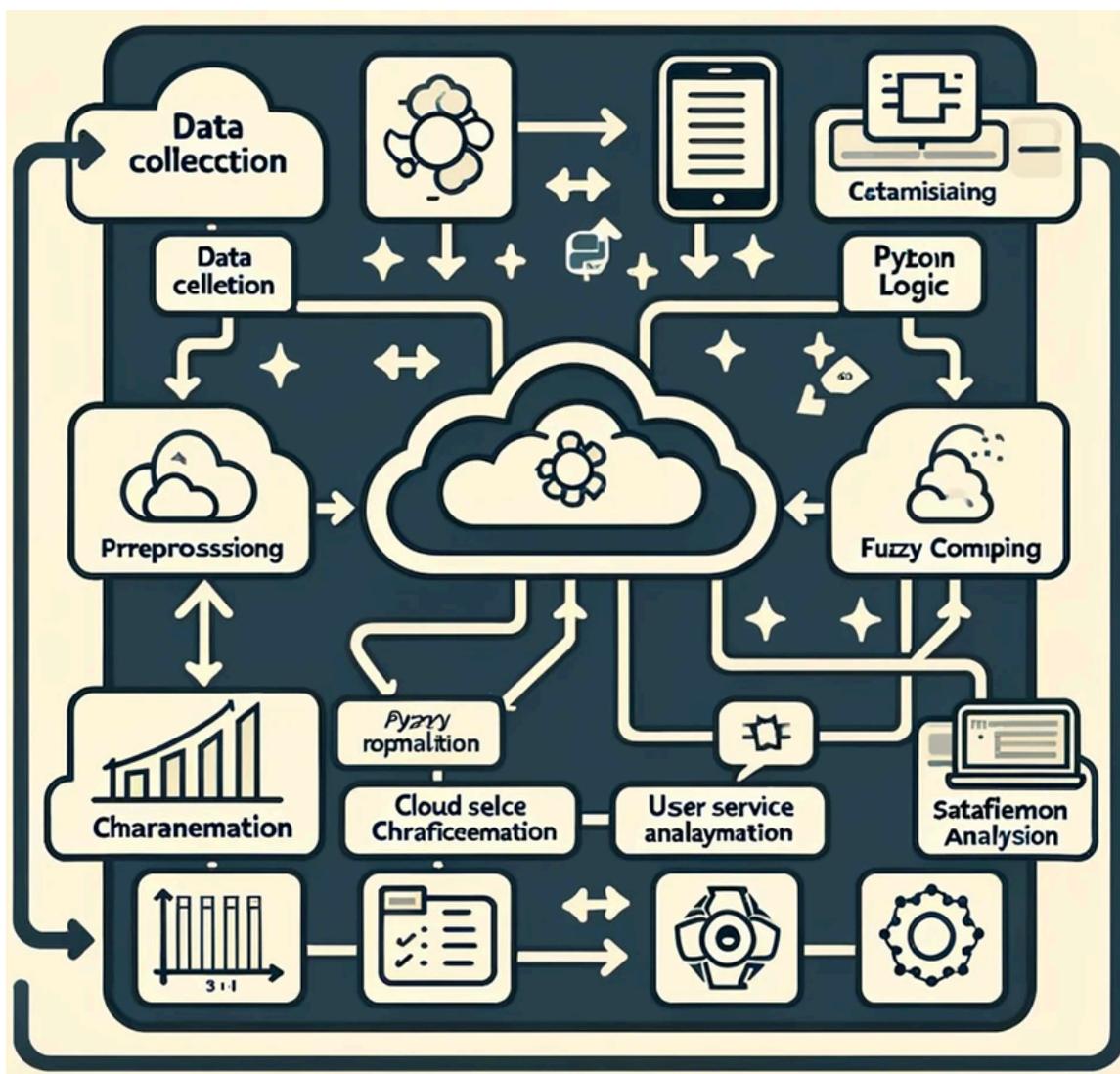


#### 4. TÉCNICAS EMPREGADAS

Neste capítulo são comentadas as técnicas empregadas na dissertação de mestrado. As técnicas aqui apresentadas visam ir ao encontro de um problema real, que necessita de uma solução, e que essa solução seja a melhor possível dentro das restrições estabelecidas no problema.

O entendimento do fluxo de funcionamento e das técnicas empregadas, está ilustrado na Figura 16.

Figura 16 - Fluxo de funcionamento e técnicas empregadas



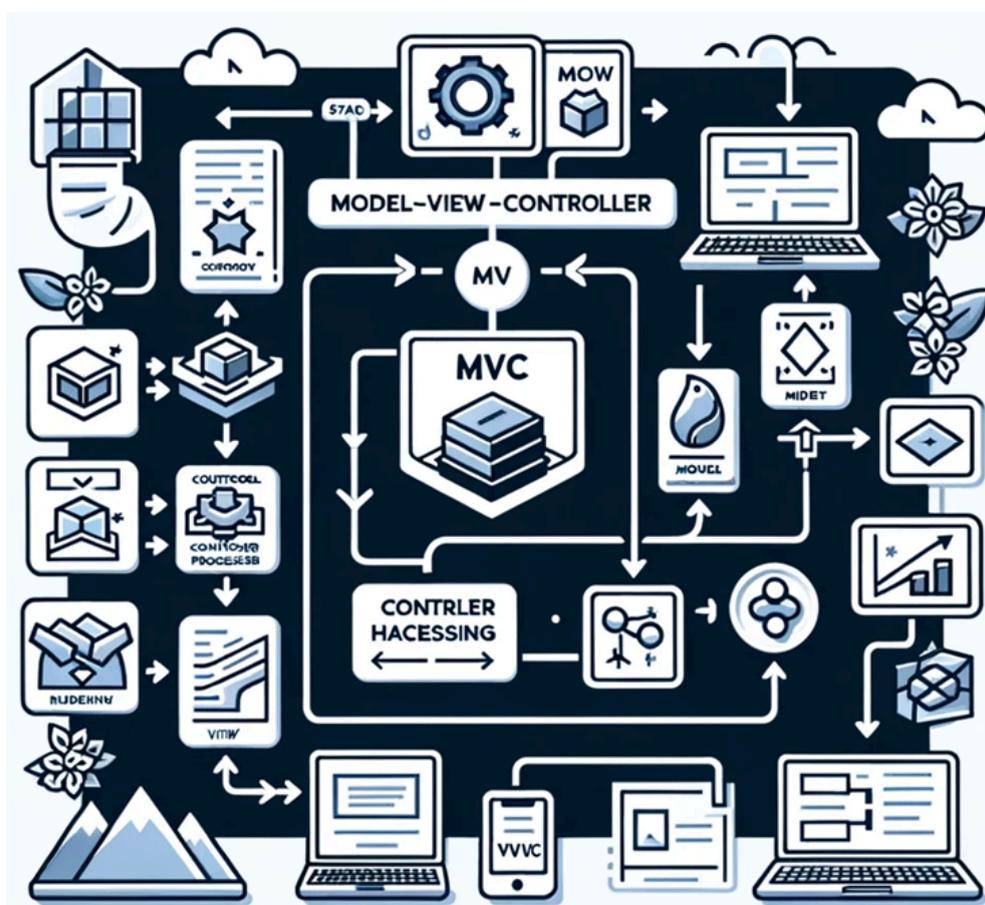
Fonte: Elaboração própria.

### 3.1. APLICAÇÃO DE ALGORITMO EM LINGUAGEM PYTHON BASEADO EM LÓGICA FUZZY

O algoritmo da aplicação é de código aberto foi elaborado durante a pesquisa e se baseia nos conceitos e padrões da programação orientada a objeto. É uma poderosa ferramenta que auxilia na tomada de decisão multicritério baseado em lógica *fuzzy*, seja através da *web* quanto intranet ou extranet, levando em consideração algumas das características de técnicas em ciência e análise de dados, sendo este, executado diretamente via *browser*<sup>10</sup>, permite o desenvolvimento colaborativo.

Este *software* foi elaborado com fins acadêmico e é essencialmente fundamentado com a tecnologia *server side* Python, independente de seu funcionamento em plataformas distintas dos mais diversos sistemas operacionais utilizados no mercado, suportando os principais sistemas gerenciadores de banco de dados do mercado.

Figura 17 - Fluxo do funcionamento do modelo MVC

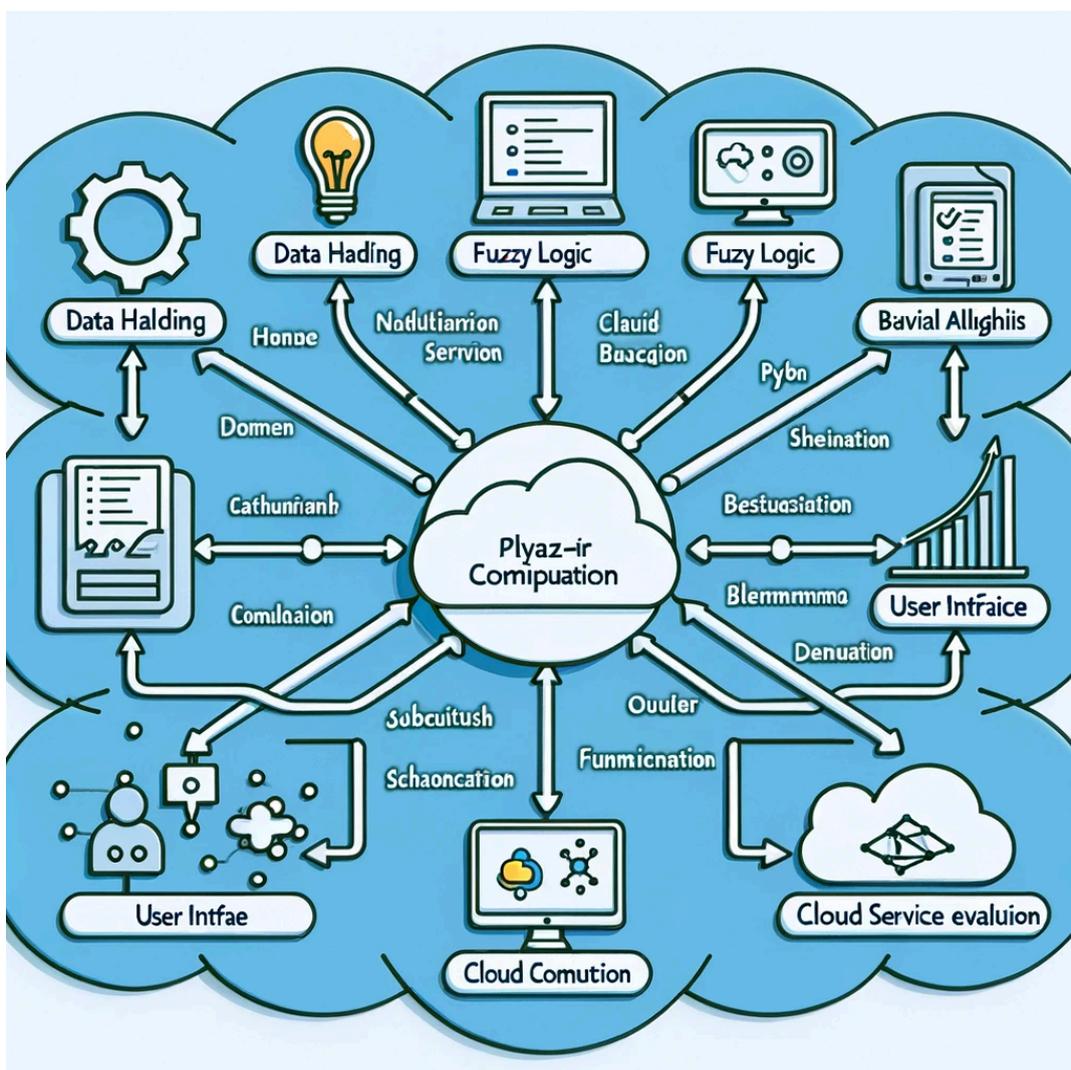


<sup>10</sup> Browser – Navegador de internet.

Fonte: Elaboração própria.

A aplicação do algoritmo de lógica *fuzzy* em linguagem Python para a caracterização de aplicações em ambiente de computação na nuvem, conforme ilustrado na Figura 17, o fluxo do funcionamento do modelo MVC, é uma abordagem sistemática que integra princípios de engenharia de software com modelagem matemática avançada, bem como o modelo MVC (*Model-View-Controller*) que é uma arquitetura de *software* que separa a lógica de negócios da interface do usuário, facilitando a gestão e a manutenção do código.

Figura 18 - Mapa Mental da aplicação do algoritmo



Fonte: Elaboração própria.

Na Figura 18, pode-se observar no mapa mental, os métodos e funcionalidades da aplicação baseada em lógica *fuzzy* são categorizados em módulos principais, incluindo:

**Manipulação de Dados:** Responsável pela importação, formatação e validação dos dados de entrada, que são essenciais para alimentar o modelo com informações precisas e relevantes para a análise.

**Algoritmos de Lógica *Fuzzy*:** Contém as funções que implementam os algoritmos de lógica *fuzzy*, permitindo a modelagem de dados incertos e proporcionando uma análise mais flexível e realista das características dos serviços de computação na nuvem.

**Avaliação do Serviço de Computação na Nuvem:** Utiliza os algoritmos de lógica *fuzzy* para avaliar diferentes aspectos dos serviços de computação na nuvem, como desempenho, segurança e escalabilidade.

**Interface do Usuário:** Onde os resultados da análise são apresentados de maneira intuitiva e acessível, permitindo aos usuários interagirem com o sistema e compreenderem os resultados da caracterização dos serviços na nuvem.

O uso do padrão MVC, juntamente com a implementação de lógica *fuzzy* em Python, proporciona uma arquitetura robusta que suporta a tomada de decisões multicritério e facilita a caracterização detalhada de aplicações em nuvem. Esta abordagem é reforçada pelas vantagens que os padrões de projeto trazem para o desenvolvimento de software, como destacado por (PRIETO; PENTEADO, 2000), incluindo a melhoria da qualidade, a uniformidade e a redução da complexidade do sistema.

Neste viés, procura-se obter produtos de software voltados para a *web* desenvolvidos com a devida qualidade de *software* requerida, pois de acordo com Boente e Dória (2012), a qualidade de *software* não pode ser avaliada isoladamente. No desenvolvimento de produtos de *software*, um método pobre ou a ausência de uma metodologia pode ser a causa da baixa qualidade. A avaliação da qualidade está diretamente relacionada com a qualidade de processos e metodologias utilizadas no desenvolvimento do produto de *software*.

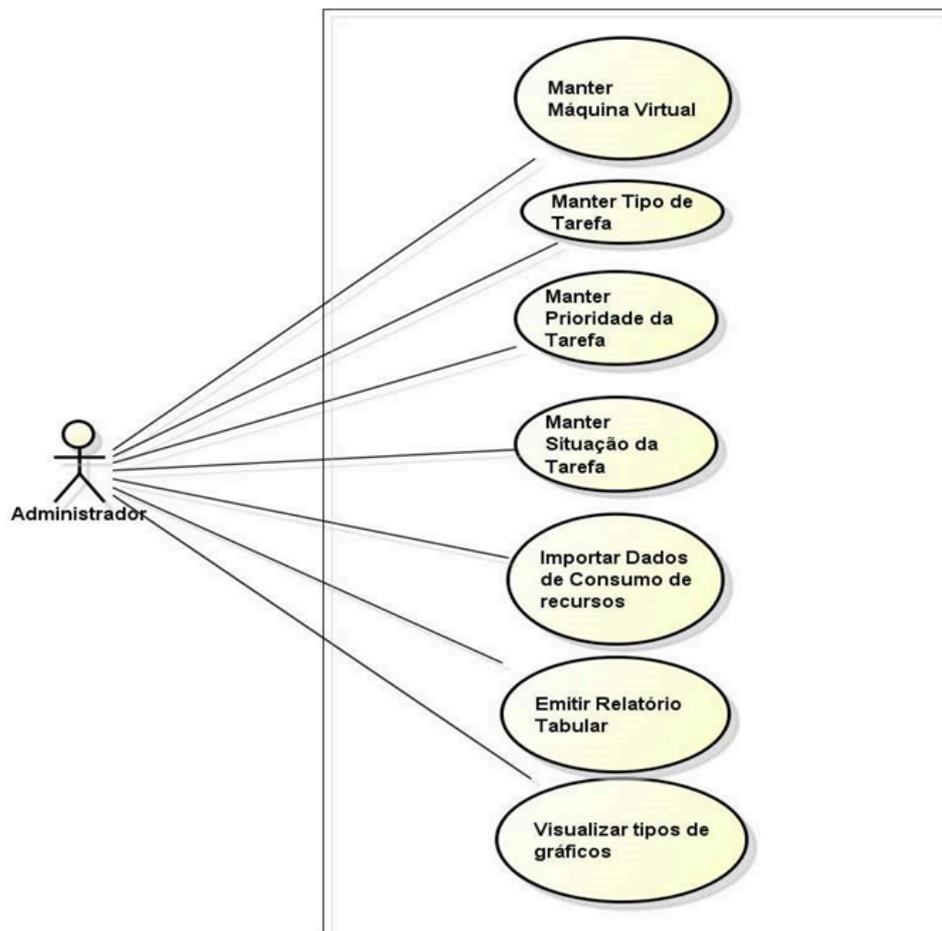
Para o desenvolvimento da aplicação proposta, foi necessário utilizar basicamente três técnicas documentais que colaboram para um *software* de qualidade, Diagrama de Caso de Uso, Diagrama de Classes e diagrama de Entidade Relacionamento.

Segundo Ferreira (2022), a avaliação da qualidade está diretamente relacionada com a qualidade de processos e metodologias utilizadas no desenvolvimento do produto de *software*.

### 3.2. DIAGRAMA DE CASO DE USO

Segundo Jacobson, Booch e Rumbaugh (2005, p. 230), um caso de uso é uma descrição de um conjunto de sequências de ações, inclusive variantes, que um sistema executa para produzir um resultado de valor observável por um ator. O diagrama de caso de uso para a aplicação é ilustrado na Figura 19.

Figura 19 - Diagramas de Caso de Uso da aplicação



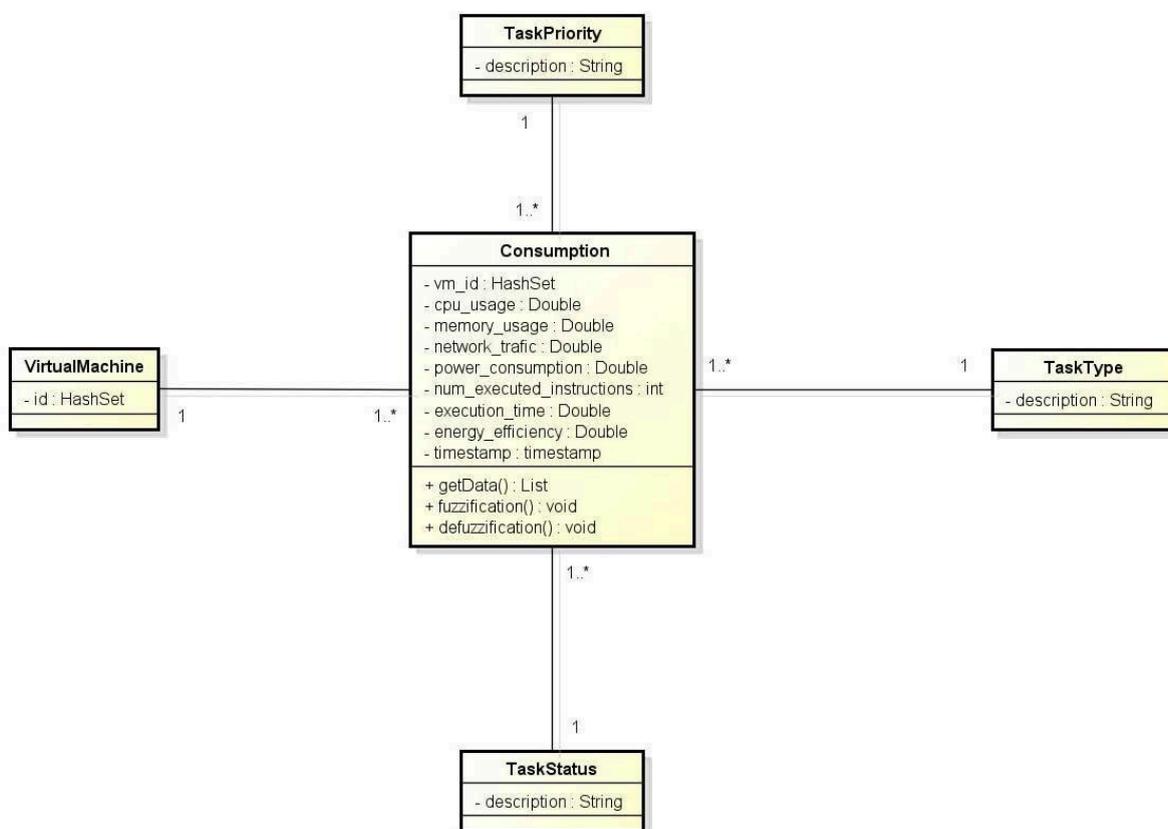
Fonte: Elaboração própria.

### 3.3. DIAGRAMA DE CLASSES

Segundo Bezerra (2015), o diagrama de classes é um detalhamento do modelo de especificação. Esse modelo corresponde a implementação das classes em alguma linguagem de programação, normalmente uma linguagem orientada a objetos.

Para este estudo, utilizou-se a linguagem de programação Python sob o paradigma orientado a objetos, cuja descrição das classes é ilustrada na Figura 20.

Figura 20 - Diagrama de Classes da aplicação



Fonte: Elaboração própria.

O diagrama de classes representa a estrutura geral do sistema, é o resultado do processo de abstração para identificar os objetos com relevância para do sistema.

Então, o diagrama de classes serve para descrever os tipos de objetos do sistema, seus atributos, métodos e relacionamentos.

### 3.4. PROJETO DE BANCO DE DADOS

O projeto de banco de dados tem grande importância no desenvolvimento de qualquer sistema de informação, pois o banco de dados é onde são armazenados todos os dados.

Etapa onde é criado o esquema conceitual do banco de dados no esquema lógico, que é a descrição da estrutura do banco que será processada pelo Sistema Gerenciador de Banco de Dados (SGBD).

Em programação baseada no paradigma orientado a objetos, utiliza-se o modelo de banco de dados orientado a objetos, representado pelo próprio diagrama de classes, considerando apenas as características das classes, ou seja, seus atributos, descartando-se seus métodos.

#### 4. SOLUÇÃO PARA O PROBLEMA PROPOSTO

A solução para o problema proposto dispõe de um *software* elaborado através de conhecimento obtido de lógica *fuzzy*, método Delphi, linguagem estruturada de consulta estruturada (SQL), PHP, bem como algoritmo baseado na linguagem de programação Python.

Antes de descrever as tecnologias empregadas, é importante descrever o que era necessário para a elaboração do algoritmo, em relação à tecnologia, para auxiliar a decisão do problema: um algoritmo para gerar a melhor solução possível dentro das condições estabelecidas no problema. Para isso ocorrer, foram elaboradas restrições para escolher o que seria usado como tecnologia, cujo detalhes estão descritos no Quadro 6.

Quadro 6 - Restrições na escolha da tecnologia (algoritmo)

<b>Restrição</b>	<b>Descrição</b>
Especificidade	O algoritmo deveria ser específico para resolver este problema, ou seja, só seria inserido o que fosse considerado importante para a resolução do problema.
Acessibilidade	Para que o usuário pudesse acessar o algoritmo e modificá-lo quando fosse preciso de forma rápida e sem maiores transtornos.
Praticidade	Para que o usuário tivesse condições de ver a solução que foi obtida salva em seu computador ou em banco de dados.
Rastreabilidade	Para que o usuário possa mapear o que está sendo desenvolvido, como está e de que forma, quando desejar.
Alto Desempenho	O melhor desempenho possível para possibilitar a melhor solução.

Baixo Custo	Uma tecnologia gratuita ou com baixo custo para a resolução do trabalho.
Fácil Aprendizado	Um algoritmo que permitisse aos usuários sem amplo conhecimento desenvolvê-lo sem grandes problemas.

**Fonte: Adaptado de FERREIRA (2015).**

Após analisar as tecnologias disponíveis, ficou estabelecido que as tecnologias utilizadas para a elaboração do algoritmo seriam Python e arquivos de extensões xls ouxlsx ou csv. Segundo (THOMSON; WELLING, 2005), uma das maiores qualidades do PHP e do MySQL é o fato de eles funcionarem com qualquer dos maiores sistemas operacionais e muitos dos menores.

Os algoritmos desenvolvidos nos meses de agosto a dezembro de 2023, objetiva o auxílio na tomada de decisão multicritérios da caracterização de aplicações em ambiente de computação na nuvem, estando inserido em um sistema elaborado especificamente para este trabalho, onde é possível manter os dados obtidos via *Kaggle*, gerar relatórios, visualizar e imprimir o resultado defuzificado.

O algoritmo possui um processo de fuzificação das respostas caracterizada como variável, dando prosseguimento, ocorre o processo de defuzificação e normalização dos valores obtidos para uma possível tomada de decisão multicritério.

A linguagem do algoritmo desenvolvida para determinar uma tomada de decisão multicritério, Python, se apresenta descrita detalhadamente a seguir, por meio das Figuras 21, 22, 23, 24, 25, 26, 27, 28, 29 e 30, respectivamente.

Figura 21 - Código que define as variáveis, funções de pertinências e termos linguísticos *fuzzy*

```

1 import numpy as np
2 import skfuzzy as fuzz
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 from skfuzzy import control as ctrl
7
8 # Passo 1: Criar as variáveis do universo
9 carregar_cpu = ctrl.Antecedent(np.arange(0, 101, 1), 'carregar_cpu')
10 memoria_utilizada = ctrl.Antecedent(np.arange(0, 101, 1), 'memoria_utilizada')
11 escalabilidade = ctrl.Consequent(np.arange(-10, 11, 1), 'escalabilidade')
12
13 # Passo 2: Definir funções de pertinência
14 # Para Carga CPU
15 carregar_cpu['baixa'] = fuzz.trimf(carregar_cpu.universe, [0, 0, 50])
16 carregar_cpu['media'] = fuzz.trimf(carregar_cpu.universe, [25, 50, 75])
17 carregar_cpu['alta'] = fuzz.trimf(carregar_cpu.universe, [50, 75, 100])
18
19 # Para Uso de Memória
20 memoria_utilizada['baixa'] = fuzz.trimf(memoria_utilizada.universe, [0, 0, 40])
21 memoria_utilizada['media'] = fuzz.trimf(memoria_utilizada.universe, [30, 50, 70])
22 memoria_utilizada['alta'] = fuzz.trimf(memoria_utilizada.universe, [60, 80, 100])
23
24 # Para Ação de Escalonamento
25 escalabilidade['diminuir_muito'] = fuzz.trimf(escalabilidade.universe, [-10, -10, -5])
26 escalabilidade['diminuir_pouco'] = fuzz.trimf(escalabilidade.universe, [-6, -3, 0])
27 escalabilidade['manter'] = fuzz.trimf(escalabilidade.universe, [-1, 0, 1])
28 escalabilidade['aumentar_pouco'] = fuzz.trimf(escalabilidade.universe, [0, 3, 6])
29 escalabilidade['aumentar_muito'] = fuzz.trimf(escalabilidade.universe, [5, 10, 10])

```

Fonte: Elaboração própria.

Figura 22 - Código que define as inferências *fuzzy*

```

32 # Passo 3: Criar as regras fuzzy
33 rule1 = ctrl.Rule(carregar_cpu['alta'] & memoria_utilizada['alta'], escalabilidade['aumentar_muito'])
34 rule2 = ctrl.Rule(carregar_cpu['baixa'] | memoria_utilizada['baixa'], escalabilidade['diminuir_muito'])
35 rule3 = ctrl.Rule(carregar_cpu['media'] & memoria_utilizada['media'], escalabilidade['manter'])
36
37 # Passo 4: Criar e simular o sistema de controle fuzzy
38 scaling_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
39 scaling = ctrl.ControlSystemSimulation(scaling_ctrl)
40
41 # Entrar com valores para CPU e memória
42 scaling.input['carregar_cpu'] = 85 # Exemplo: 85% de carga CPU
43 scaling.input['memoria_utilizada'] = 70 # Exemplo: 70% de uso de memória
44
45 # Calcular o resultado
46 scaling.compute()
47
48 print(scaling.output['escalabilidade'])

```

Fonte: Elaboração própria.

Figura 23 - Código que define o carregamento dos dados a serem tratados

```

v Base dados obtida do Kaggle

[ ] 1 df = pd.read_csv('vmCloud_data.csv')
     2 df.head(10)

```

Fonte: Elaboração própria.

Figura 24 - Código que define a média de consumo ou utilização de recursos de computação em nuvem

```

v Calcula as médias mensais das colunas cpu_usage, memory_usage e
  power_consumption.

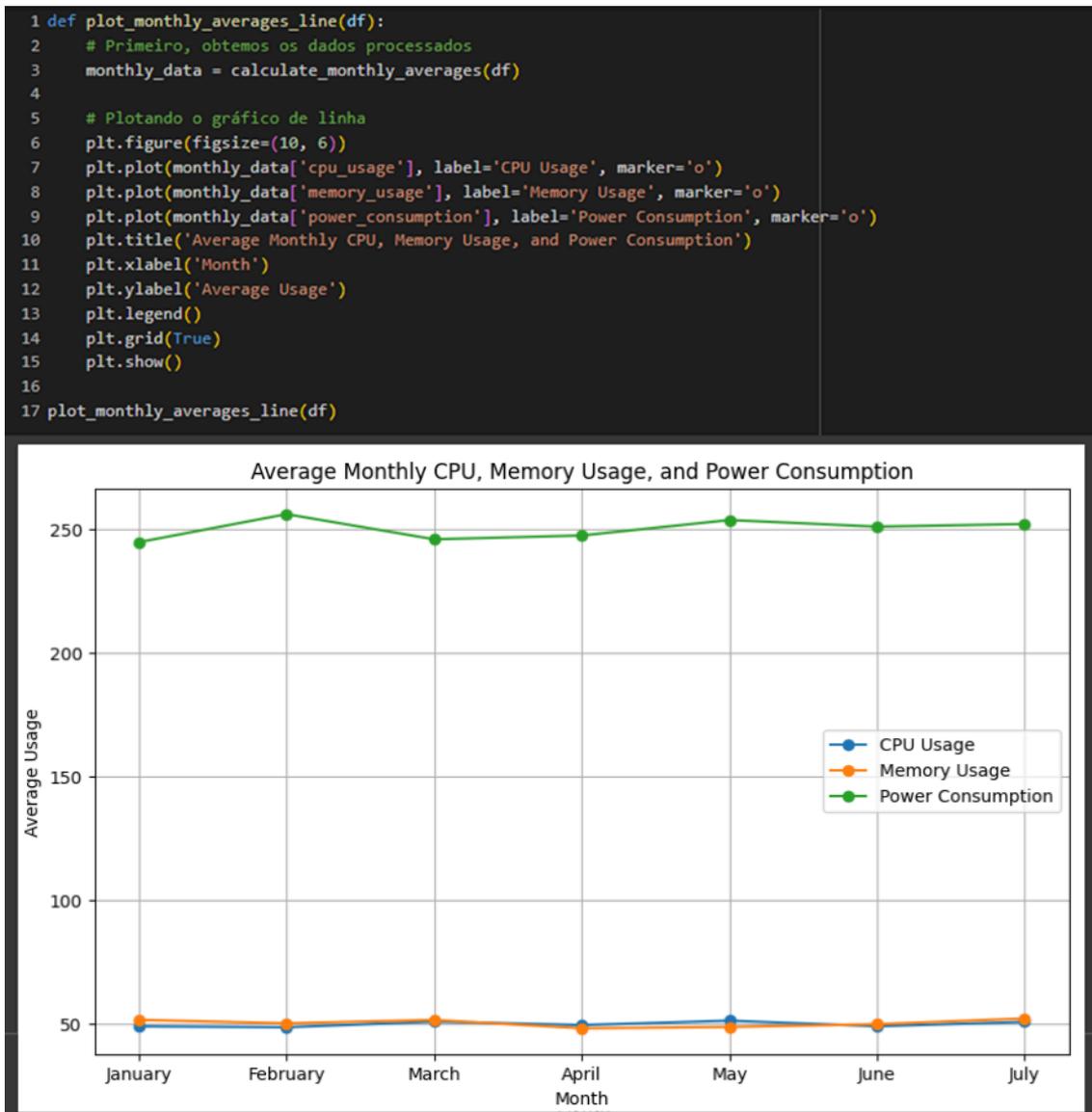
[ ] 1 # Parâmetros: df (pandas.DataFrame): DataFrame contendo os dados, incluindo uma coluna 'timestamp'.
     2 # Retorna: pandas.DataFrame: DataFrame com as médias mensais das colunas especificadas.
     3
     4 def calculate_monthly_averages(df):
     5
     6     # Converter 'timestamp' para datetime e extrair o mês
     7     df['timestamp'] = pd.to_datetime(df['timestamp'])
     8     df['month'] = df['timestamp'].dt.month
     9
    10     # Mapear números dos meses para nomes por extenso
    11     df = map_month_names(df)
    12
    13     # Criar uma categoria para ordenar os meses corretamente
    14     df['month_name'] = pd.Categorical(df['month_name'], categories=month_names.values(), ordered=True)
    15
    16     # Calcular a média mensal das colunas de interesse
    17     average_monthly_consumption = df.groupby('month_name')[['cpu_usage', 'memory_usage', 'power_consumption']].mean()
    18
    19     return average_monthly_consumption
    20
    21 # Uso da função
    22 avg_consumption = calculate_monthly_averages(df)
    23 print(avg_consumption)
    24

```

month_name	cpu_usage	memory_usage	power_consumption
January	50.031754	49.732502	247.928426
February	49.937779	50.078338	249.686502
March	49.878674	50.414807	249.992171
April	49.852901	49.771154	251.384672
May	50.456985	49.731022	249.451562
June	49.937449	49.767584	251.274920
July	49.933157	50.070940	251.211799

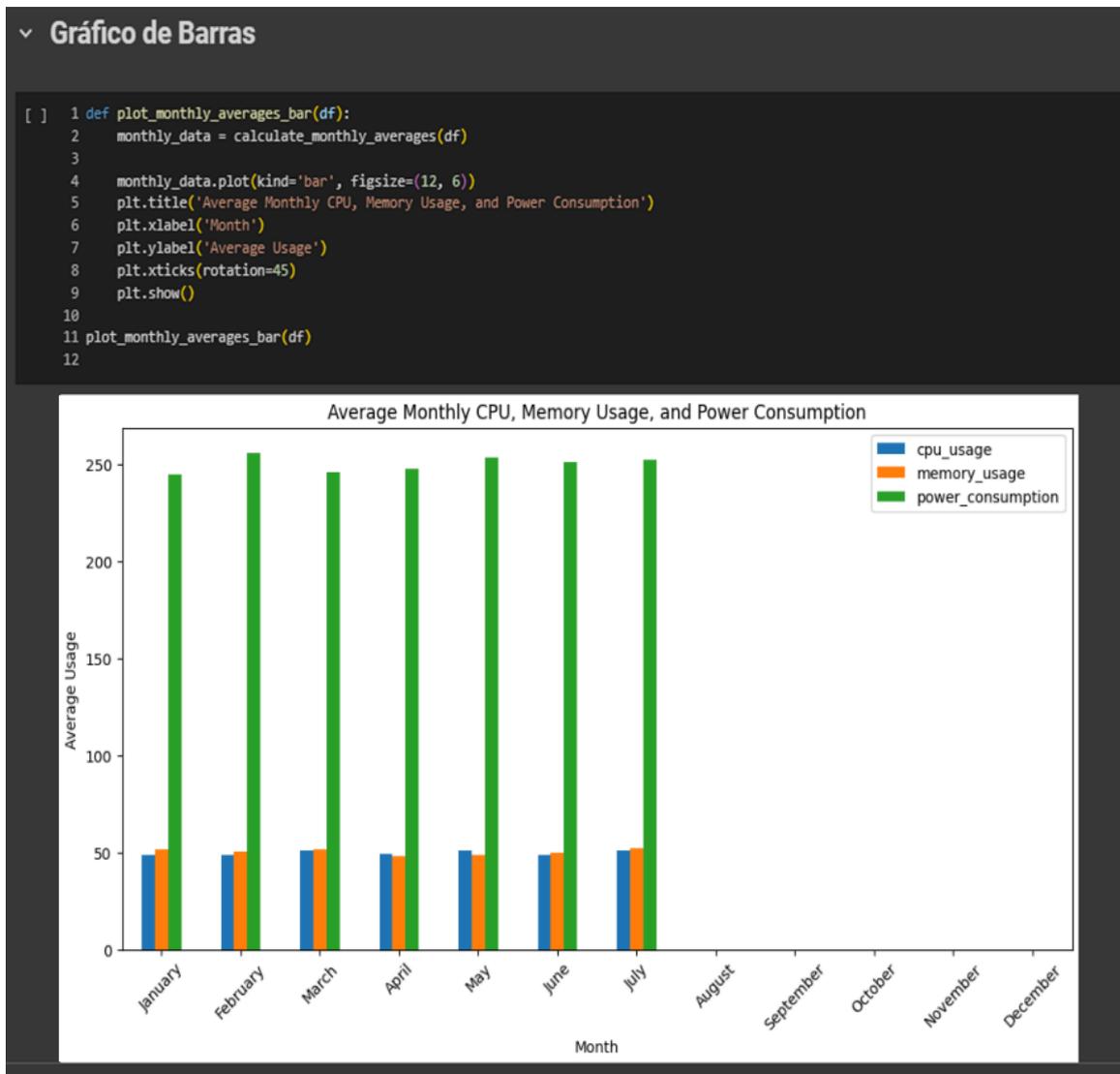
Fonte: Elaboração própria.

Figura 25 - Código que define a exibição do gráfico de consumo ou utilização de recursos de computação em nuvem por mês



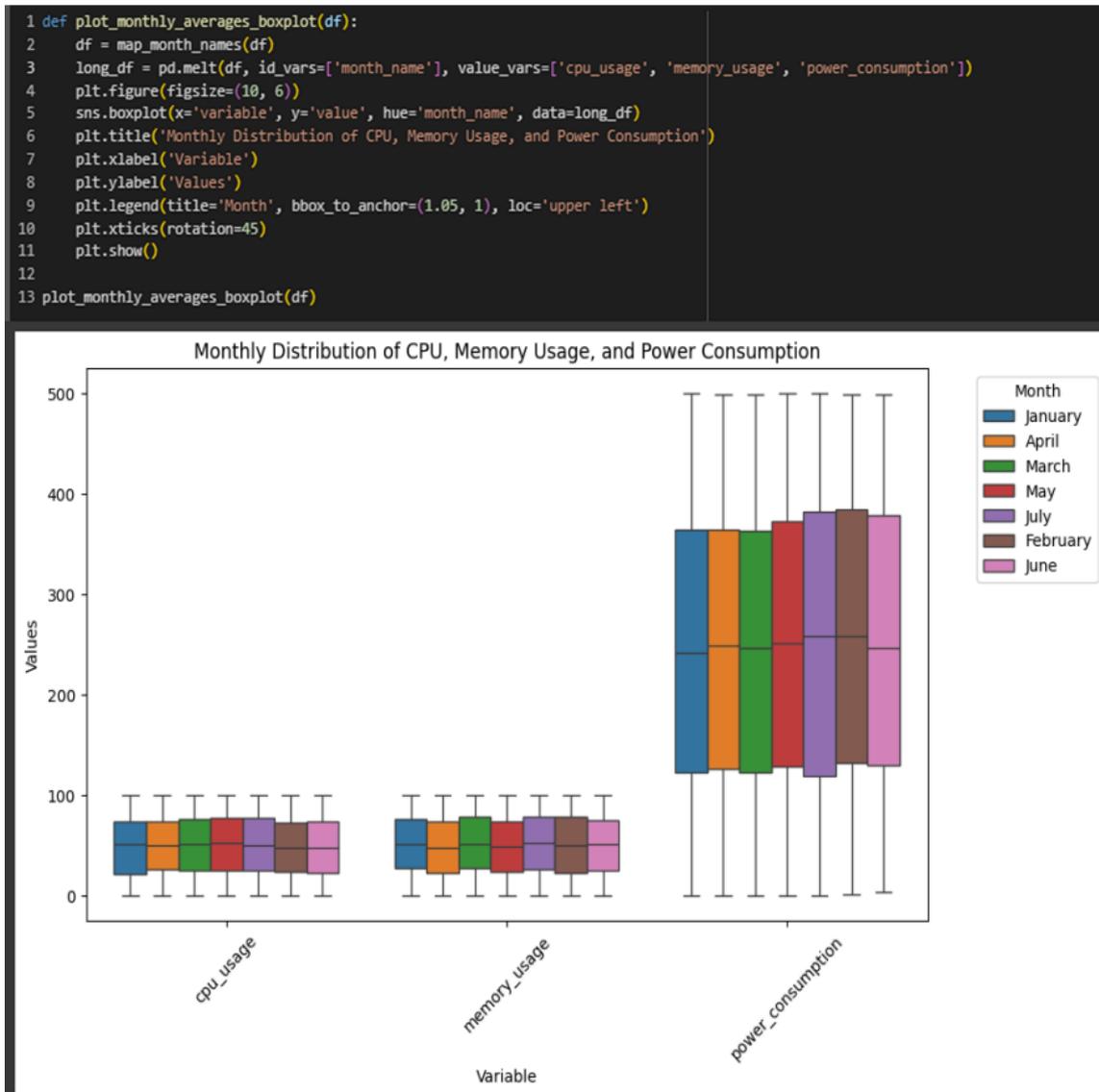
Fonte: Elaboração própria.

Figura 26 - Código que define a exibição do gráfico de barras de consumo ou utilização de recursos de computação em nuvem por mês



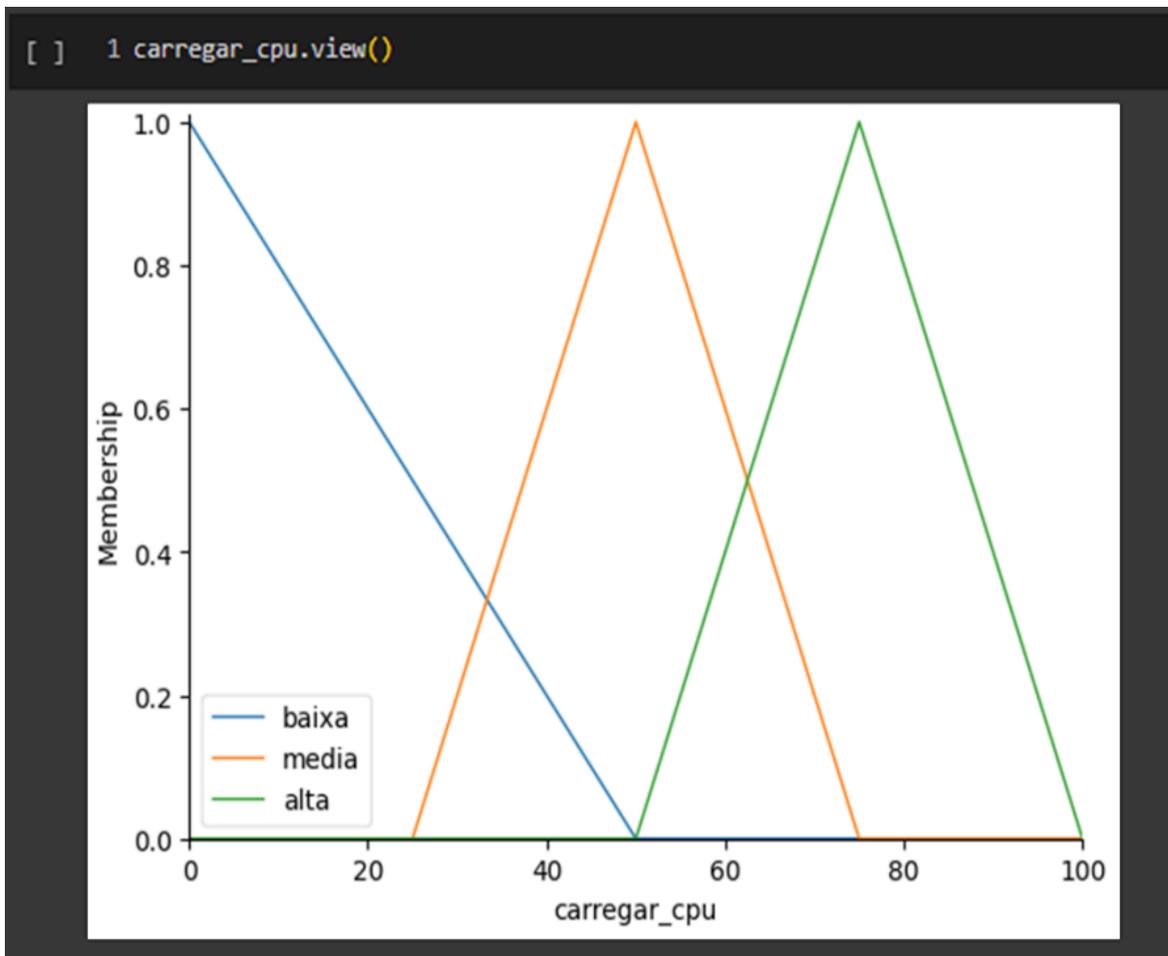
Fonte: Elaboração própria.

Figura 27 - Código que define a exibição do gráfico de distribuição de consumo ou utilização de recursos de computação em nuvem por mês



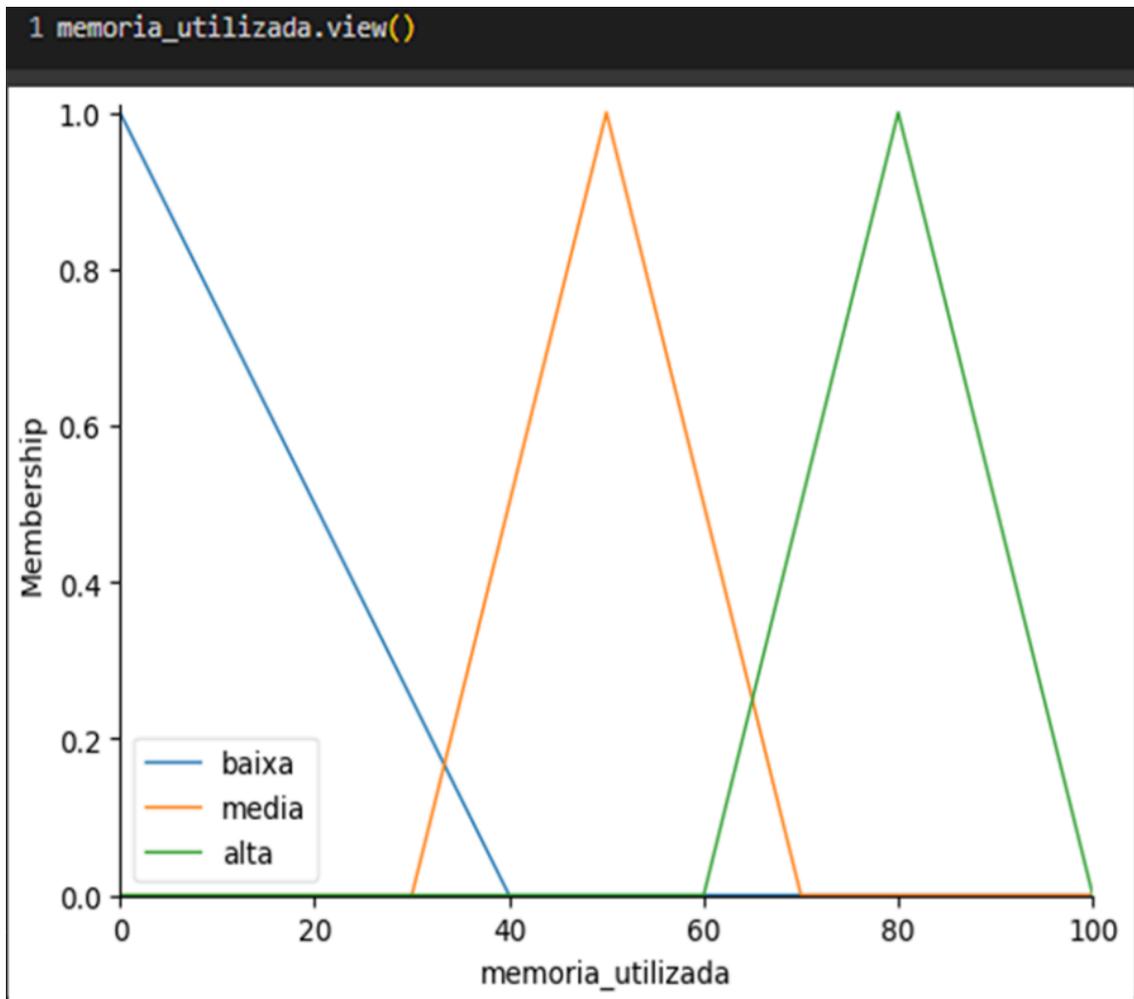
Fonte: Elaboração própria.

Figura 28 - Código que define a exibição do gráfico triangular de utilização da CPU em nuvem



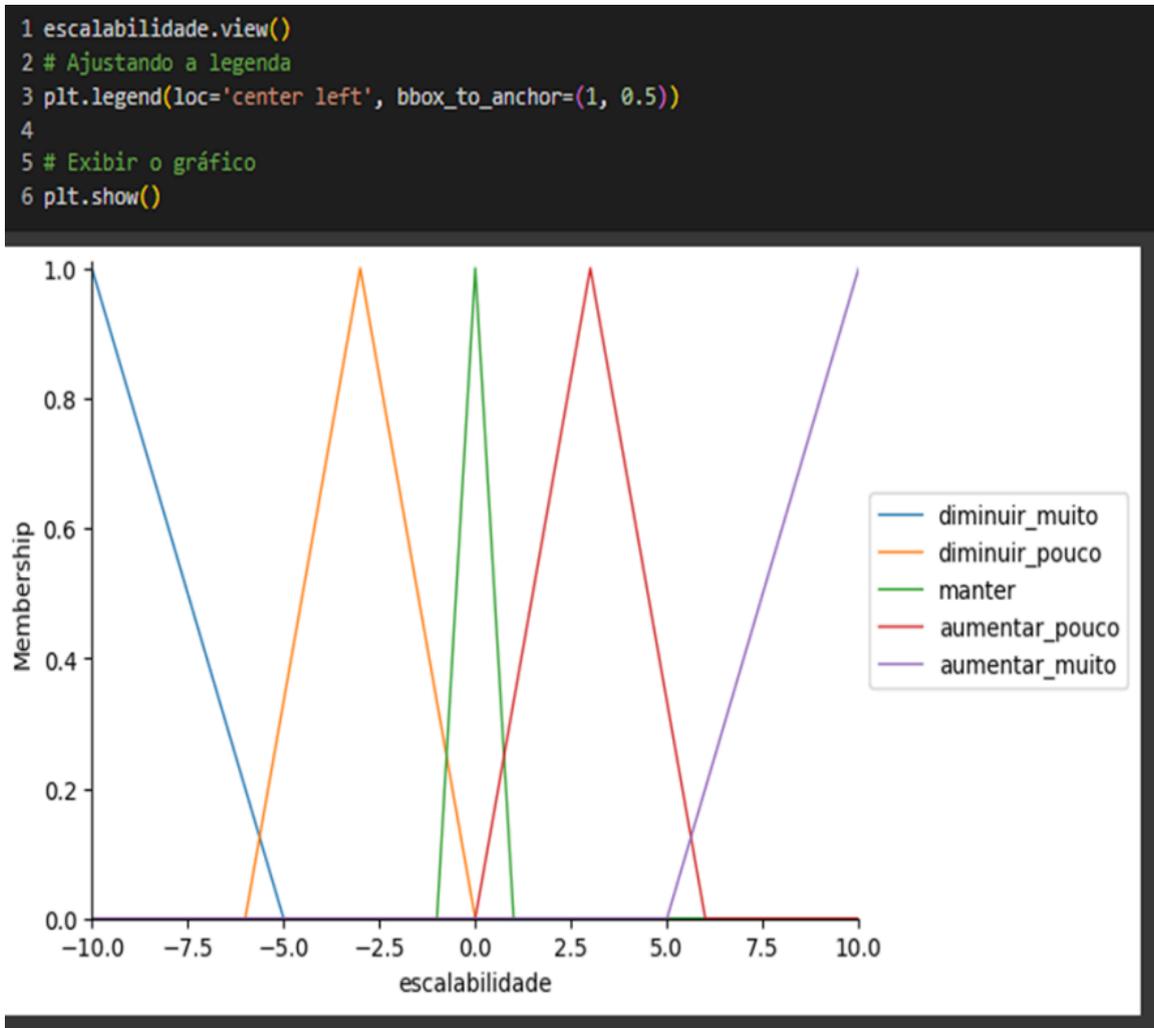
Fonte: Elaboração própria.

Figura 29 - Código que define a exibição do gráfico triangular de utilização da memória em nuvem



Fonte: Elaboração própria.

Figura 30 - Código que define a exibição do gráfico triangular de escalonamento de recursos



Fonte: Elaboração própria.

A seguir, no Capítulo 6 são encontradas as considerações finais e os trabalhos futuros do presente trabalho.

## 6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Nesta dissertação foi proposto um modelo *fuzzy* para caracterização de aplicações em ambiente de computação em nuvem que fosse capaz de analisar possibilidades alternativas para que os provedores da nuvem sejam capazes de controlar, configurar e escalar os ambientes computacionais conforme as necessidades de cada tipo de cliente ou em conformidade com as necessidades de negócio que o provedor da nuvem possa vir a demandar.

Para tanto, a abordagem *fuzzy* permitiu tratar, de forma matematicamente sólida, medidas subjetivas sujeitas a incertezas e imprecisões, obtidas através de dados discretos advindos a partir de critérios de avaliação e de inferência das máquinas virtuais utilizadas em ambientes de computação em nuvem, acerca de quatro diferentes óticas: perspectiva do consumo, configuração ideal de máquina virtual transacional, configuração ideal de máquina virtual distribuídas e configuração ideal de máquina virtual paralelas. Também, foi considerada a agregação *fuzzy* sob as opiniões de sete especialistas em computação em nuvem, considerando seu respectivo peso, calculado e avaliado, com base nos seguintes critérios: grau de conhecimento sobre computação em nuvem, o currículo dos especialistas, a experiência dos especialistas em computação em nuvem, o grau de instrução, a participação em congressos, a participação em projetos de computação em nuvem e a comprovação de prática efetiva em computação em nuvem.

Neste trabalho, depois da definição de variáveis, funções de pertinências, termos linguísticos e inferência *fuzzy*, tendo como base dados obtidos a partir do *Kaggle*, foi calculada a média de consumo mensal ou utilização de recursos de computação em nuvem, necessária para a criação do gráfico de linha de consumo ou utilização de recursos de computação em nuvem, gráfico de barras de consumo ou utilização de recursos de computação em nuvem e do gráfico de distribuição de consumo ou utilização de recursos de computação em nuvem.

Neste contexto, também se obteve o gráfico triangular *fuzzy* de utilização da CPU em computação em nuvem, o gráfico triangular *fuzzy* de utilização da memória em computação em nuvem e o gráfico triangular *fuzzy* de escalonamento de recursos utilizados em ambientes de computação em nuvem, que serão utilizados como base para

análise do processo de tomada de decisão, a fim de otimizar e melhorar o uso efetivo de CPU, memória e escalonamento em ambiente de computação em nuvem.

No decorrer dos experimentos foi observado algumas dificuldades, tais como a utilização gratuita dos recursos em nuvem da Microsoft Azure, que é limitada em sua versão gratuita, o que para experimentos dessa natureza poderia ter influenciado na representação final dos dados e em seu comportamento, esse problema foi resolvido a partir do momento que a plataforma *Kaggle* disponibilizou dados de máquinas virtuais para pesquisa da comunidade científica, bem como seus recursos que a compõe, tais como espaço em disco, poder de processamento, consumo de energia, tempo de processamento, data e hora e por entenderem que o trabalho desses profissionais em múltiplas áreas do conhecimento, ajudam na descoberta de inovações e a tornar os recursos melhores.

Apesar dos índices alcançados para obtenção dos gráficos analíticos que balizarão os processos de tomada de decisão por parte do especialista em computação em nuvem, serem considerados bons, foram identificadas lacunas, em relação ao uso de CPU, memória e escalonamento de recursos, que estão afetando diretamente a qualidade dos serviços de computação em nuvem prestados aos clientes que os solicitam.

Com base nessas informações os especialistas em computação em nuvem podem sugerir melhorias de otimização quanto ao uso da CPU, memória e escalabilidade de recursos em ambientes de computação em nuvem, alcançando melhor a eficácia e a eficiência em serviços em ambientes de computação em nuvem.

Pesquisas futuras poderão replicar o modelo *fuzzy* aqui discorrido, como por exemplo, nas áreas de robótica e automação industrial, em qualquer tipo de indústria sob a perspectiva da quarta revolução industrial ou superior.

Neste viés, poder-se-á acrescentar, a este modelo *fuzzy*, cálculos que nos permitam aferir a qualidade dos serviços prestados em ambiente de computação em nuvem (IaaS, PaaS e SaaS), assim como a satisfação de seus usuários, hipoteticamente, permitindo aprimorar o processo de tomada de decisão desses especialistas.

Ademais, pretende-se transformar esta dissertação em artigos a serem submetidos para publicação em revistas conceituada nas áreas da Engenharia de Produção, Ciência da Computação, Engenharia de Software e Sistemas de Informação, como também, desenvolver outros trabalhos relacionados às áreas mencionadas

anteriormente, contribuindo para uma extensão dos estudos em pós-graduação em nível de doutoramento.

## REFERÊNCIAS

- ALMURISI, N., TADISSETY, S. (2022). **Cloud-based virtualization environment for IoT-based WSN: solutions, approaches and challenges**. *J Ambient Intell Human Comput* 13, 4681–4703 (2022). Disponível em <<https://doi.org/10.1007/s12652-021-03515-z>>. Acessado em: 3 dez. 2022.
- AMAZON WEB SERVICES, (2023). **A. O que é SLA?** Explicação sobre acordos de nível de serviço - AWS. Disponível em: <<https://aws.amazon.com/pt/what-is/service-level-agreement/>>. Acesso em: 18 jan. 2023.
- AZEVEDO, L.J.M. et al. (2018). **Optimized Service Level Agreement Establishment in Cloud Computing**. *The Computer Journal*, v. 61, n. 10, p. 1429–1442, 1 out. 2018.
- BASET, A.; SCHULZRINNE, H.; SHIM, E. (2006). **A Common Protocol for Implementing Various DHT Algorithms**. Disponível em: <<https://docslib.org/doc/4957125/a-common-protocol-for-implementing-various-dht-algorithms>>. Acesso em: 18 jun. 2023.
- BEZERRA, E. (2015). **Princípios de Análise e Projeto de Sistemas com UML**. 3 ed. Rio de Janeiro: Elsevier.
- BOENTE, A.N.P. (2013). **Proposição de um modelo fuzzy para tomada de decisão acerca da avaliação da qualidade do produto de software AVA MOODLE utilizado no Curso de Pós-Graduação em Tecnologias Educacionais do IST-Rio e da satisfação de seus usuários**. Tese de Doutorado submetida ao programa de pós-graduação de Engenharia de Produção da Universidade Federal do Rio de Janeiro, COPPE/UFRJ.
- BOENTE, A.N.P. e DÓRIA, F.A.M.A. (2012). **Proposição de um Modelo Fuzzy para Tomada de Decisão acerca de Ambiente Virtual de Aprendizagem: AVA Moodle**. In: IX SEGeT, Simpósio de Excelência e Gestão em Tecnologia. Resende, Rio de Janeiro, Out. 2012.
- BOJADZIEV, G.; BOJADZIEV, M. (2007). **Fuzzy Logic For Business, Finance, And Management** (2<sup>nd</sup> Edition). [s.l.] World Scientific.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. (2005). **The Unified Modeling Language User Guide**. [s.l.] Addison-Wesley.
- BORGES, L.E. (2010). **Python para Desenvolvedores**. 2. ed. Rio de Janeiro: Edição do Autor, 2010. ISBN 978-85-909451-1-6.
- CÂNDIDO, A.C.; ARAÚJO JÚNIOR, R.H. DE. (2022). **Potencialidades do desenvolvimento de cloud computing no âmbito da gestão da informação**. *Perspectivas em Ciência da Informação*, v. 27, p. 57–80, 6 maio 2022.
- COSENZA, H.J.S.R. et al. (2006). **Aplicação de Um Modelo de Hierarquização como Instrumento para Tomada de Decisão: Caso de uma Multinacional**. In: XXVI Encontro Nacional de Engenharia de Produção, ENEGEP, 2006, Fortaleza.
- FERREIRA, V.M.S. (2015). **Modelo de Apoio à Avaliação Institucional de IES com base no Método Delphi e Lógica Fuzzy para tomada de decisão multicritério: Estudo de caso da avaliação institucional da DESUP**. Dissertação de Mestrado

submetida ao programa de pós-graduação de Engenharia de Produção da Universidade Federal do Rio de Janeiro, COPPE/UFRJ.

\_\_\_\_\_. (2022). **Lógica Fuzzy Aplicada à Análise de Conflito de ideias em Redes Sociais**. Tese de Doutorado submetida ao programa de pós-graduação em História das Ciências e das Técnicas e Epistemologia da Universidade Federal do Rio de Janeiro, HCTE/UFRJ.

GOVERNO DO BRASIL. (2022). **Modelo de contratação de software e Serviços de Computação em Nuvem**. Disponível em <<https://www.gov.br/participamaisbrasil/modelo-de-contratacao-de-software-e-nuvem>>. Acesso em: 05 jan. 2024.

GOLIGHTLY, L. et al. (2022). **Adoption of cloud computing as innovation in the organization**. International Journal of Engineering Business Management, v. 14, p. 18479790221093992, 1 nov. 2022.

GRANVILLE, L.Z. (2021). **Gerenciamento integrado de QoS em redes de computadores**. Tese de Doutorado submetida ao programa de pós-graduação em computação da Universidade Federal do Rio Grande do Sul, UFRS.

HASSANZADEH-NAZARABADI, Y.; TAHERI-BOSHROOYEH, S.; ÖZKASAP, Ö. (2022). **DHT-based Edge and Fog Computing Systems: Infrastructures and Applications**. IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Anais...2 maio 2022. Disponível em: <<http://arxiv.org/abs/2211.05851>>. Acesso em: 18 jan. 2023.

HAYAT, B; KIM, K.H. (2018). **A study on fuzzy logic based cloud computing**. Cluster Comput, SPRINGER, ano 2018, p. 589-603, 3 jun.

INTEL, Integrated Electronics. (2023). **Uma Introdução ao Projeto de Arquitetura em Nuvem**. Intel. Disponível em <<https://www.intel.com.br/content/www/br/pt/cloud-computing/cloud-architecture.html>>. Acessado em: 03 dez. 2023.

IBM, International Business Machines Corporation. (2023). **Tipos de virtualização**. IBM. Disponível em <<https://www.ibm.com/br-pt/topics/virtualization>>. Acessado em: 3 dez. 2023.

KAUR, J. (2023). **Azure Automation: A Comprehensive Recap of 2022 and What's Coming**. Disponível em: <<https://techcommunity.microsoft.com/t5/azure-governance-and-management/azure-automation-a-comprehensive-recap-of-2022-and-what-s-coming/ba-p/3810885>>. Acesso em: 18 nov. 2023.

KUMAR, N. et al. (2014). **Achieving Quality of Service (QoS) Using Resource Allocation and Adaptive Scheduling in Cloud Computing with Grid Support**. The Computer Journal, v. 57, n. 2, p. 281–290, 1 fev. 2014.

LOPES, N.; BAQUERO, C. (2023). **Análise da disponibilização de um índice invertido em p2p**. In: 6<sup>th</sup> Conferência sobre Redes de Computadores: Protocolos, tecnologias e aplicações para ambientes móveis, Bragança, Portugal.

MICROSOFT. (2023). **O que é o Microsoft Defender para Nuvem?** Microsoft Defender for Cloud. Disponível em:

<<https://learn.microsoft.com/pt-br/azure/defender-for-cloud/defender-for-cloud-introduction>>. Acesso em: 10 dez. 2023.

MICROSOFT AZURE. (2023). **Generally Available**: Azure Monitor VM Insights using Azure Monitor Agent | Azure updates | Microsoft Azure. Disponível em: <<https://azure.microsoft.com/en-us/updates/generally-available-azure-monitor-vm-insights-using-azure-monitor-agent/>>. Acesso em: 5 out. 2023.

MORÉ, J.D. (2004). **Aplicação da lógica Fuzzy na avaliação da confiabilidade humana nos ensaios não destrutivos por ultra-som**. Tese de Doutorado submetida ao programa de pós-graduação de Engenharia Metalúrgica e dos Materiais da Universidade Federal do Rio de Janeiro, COPPE/UFRJ.

NETO, M.V.S. (2015). **Computação em Nuvem**: Nova Arquitetura de TI. Rio de Janeiro: Brasport.

OGURA, D.R. (2011). **Uma metodologia para caracterização de aplicações em ambientes de computação nas nuvens**. Dissertação de Mestrado submetida ao programa de pós-graduação em Sistemas Digitais da Universidade de São Paulo, USP.

PEREIRA, D.L. (2014). **Comparação do Impacto do Canal de Comunicação no Desempenho de Grids**. p. 69, 14 nov. 2014.

PRIETO, G. A.; PENTEADO, R. D. (2000). **Utilização de Padrões de Projeto de Software na Reengenharia de Sistemas**. Anais Estendidos do Simpósio Brasileiro de Engenharia de Software (SBES). Anais. Em: ANAIS ESTENDIDOS DO XIV SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE. SBC, 4 out. 2000. Disponível em: <[https://sol.sbc.org.br/index.php/sbes\\_estendido/article/view/24207](https://sol.sbc.org.br/index.php/sbes_estendido/article/view/24207)>. Acesso em: 18 jan. 2023.

RONG, C.; ZHAO, Z. (2021). **Welcome to the new Journal of Cloud Computing by Springer**. Journal of Cloud Computing, v. 10, n. 1, p. 49, 10 set. 2021.

ROSSUM, G. VAN. (2015). **25 Years of Python at CWI**. Disponível em: <<https://www.cwi.nl/en/news/25-years-of-Python-at-cwi/>>. Acesso em: 18 nov. 2023.

SHARMA, M.; GUPTA, R.; ACHARYA, P. (2020). **Analysing the adoption of cloud computing service**: a systematic literature review. Global Knowledge, Memory and Communication, v. 70, n. 1/2, p. 114–153, 1 jan. 2020.

STEPHENSON, M. (2023). **Azure Application Insights vs Log Analytics**: Which one to choose? - Resultados da busca Yahoo Search. Disponível em: <<https://www.serverless360.com/blog/azure-application-insights-vs-log-analytics>>. Acesso em: 26 nov. 2023.

THOMSON, L.; WELLING, L. (2005). **Php E Mysql Desenvolvimento Web**. 3 ed. Rio de Janeiro: Campus.

TWIDI. (2021). **Step-by-Step Cloud Evolution with Microsoft Azure**. Disponível em: <[https://www.rackspace.com/sites/default/files/2022-11/Step-by-step\\_cloud\\_evolution\\_with\\_Microsoft\\_Azure\\_E-book-EMEA.PDF](https://www.rackspace.com/sites/default/files/2022-11/Step-by-step_cloud_evolution_with_Microsoft_Azure_E-book-EMEA.PDF)>. Acesso em: 1 dez. 2023.

VACCARO, G. L. R. (1997). **Modelagem e análise da confiabilidade de sistemas**. Dissertação de Mestrado submetida ao programa de pós-graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul, UFRS.



## APÊNDICE

# **QUESTIONÁRIO ESTRUTURADO**

**PESQUISA DE OPINIÃO**

**IDENTIFICAÇÃO DOS PESOS DOS  
ESPECIALISTAS EM COMPUTAÇÃO EM NUVEM**

## QUESTIONÁRIO - PESQUISA DE OPINIÃO - Identificação do Peso dos Especialistas em Computação em Nuvem

Solicitamos sua colaboração, respondendo as questões apresentadas a seguir, conforme as seguintes orientações:

- 1) Não é preciso se identificar nem assinar esta pesquisa.
- 2) Responda marcando com um X dentro dos parênteses que correspondam a sua resposta.
- 3) Responda a todas as perguntas, sem acrescentar observações ou alterar o formato do formulário.

1. Qual o seu conhecimento sobre os serviços disponibilizados em ambiente de computação em nuvem (SaaS, PaaS, IaaS)?  
( ) Nenhum ( ) Pouco ( ) Razoável ( ) Muito ( ) Excepcional
2. Seu currículo na plataforma lattes apresenta evidências factíveis sobre seu domínio com aplicações em cloud computing?  
( ) Nenhuma ( ) Pouca ( ) Razoável ( ) Muita ( ) Excepcional
3. A quanto tempo você trabalha com computação em nuvem (em anos)?  
( ) 0 ( ) Entre 1 e 2 ( ) Entre 3 e 5 ( ) Entre 6 e 9 ( ) 10 ou mais
4. Marque a alternativa que remete assertivamente o seu grau de instrução?  
( ) Graduação ( ) Especialização ( ) Mestrado ( ) Doutorado ( ) Pós-Doutorado
5. Já participou de eventos ou congressos que tivessem mostrado aplicações em ambiente de computação em nuvem? De quantos?  
( ) Nenhum ( ) Entre 1 e 2 ( ) Entre 3 e 4 ( ) Entre 5 e 6 ( ) 7 ou mais
6. Qual o número de Projetos em Ambientes de Computação em Nuvem você já participou?  
( ) Nenhum ( ) Apenas 1 ( ) Entre 2 e 5 ( ) Entre 6 e 9 ( ) 10 ou mais
7. A quantos anos trabalha com aplicações em ambiente de computação em nuvem?  
( ) 0 ( ) Apenas 1 ( ) Entre 2 e 5 ( ) Entre 6 e 9 ( ) 10 ou mais

