



**MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO AMAZONAS  
FACULDADE DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA ELÉTRICA**



**ALGORITMO DE SELEÇÃO DE POLÍTICAS DE MIGRAÇÃO DE MÁQUINAS  
VIRTUAIS EM FUNÇÃO DE DIFERENTES DEMANDAS DE RECURSOS**

**LUIZ CARLOS SILVA DE SALES**

**MANAUS**

**2023**

LUIZ CARLOS SILVA DE SALES

ALGORITMO DE SELEÇÃO DE POLÍTICAS DE MIGRAÇÃO DE MÁQUINAS  
VIRTUAIS EM FUNÇÃO DE DIFERENTES DEMANDAS DE RECURSOS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Orientador: Prof. D. Sc. Celso Barbosa Carvalho

MANAUS

2023

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S163a Sales, Luiz Carlos Silva de  
Algoritmo de seleção de políticas de migração de máquinas virtuais em função de diferentes demandas de recursos / Luiz Carlos Silva de Sales . 2023  
86 f.: il. color; 31 cm.

Orientador: Celso Barbosa Carvalho  
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Mobilidade usuário. 2. Computação em névoa. 3. Máquinas virtuais. 4. Algoritmo de seleção. I. Carvalho, Celso Barbosa. II. Universidade Federal do Amazonas III. Título



**Poder Executivo**  
**Ministério da Educação**  
**Universidade Federal do Amazonas**  
**Faculdade de Tecnologia**  
**Programa de Pós graduação em Engenharia Elétrica**


**LUIZ CARLOS SILVA DE SALES**


**ALGORITMO DE SELEÇÃO DE POLÍTICAS DE MIGRAÇÃO DE MÁQUINAS VIRTUAIS EM FUNÇÃO DE DIFERENTES DEMANDAS DE RECURSOS**


Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovada em 18 de dezembro de 2023.

**BANCA EXAMINADORA**

  
Prof. Dr. Celso Barbosa Carvalho  
Presidente  
Universidade Federal do Amazonas

  
Prof. Dr. Waldir Sabino da Silva Júnior, Membro  
Universidade Federal do Amazonas

  
Prof. Dr. Cleonor Crescencio das Neves, Membro  
Instituto Federal do Amazonas

## **AGRADECIMENTOS**

Minha esposa, Márcia Priscila Borges, tem sido uma fonte inesgotável de apoio. Mesmo em meio a tempos difíceis, ela esteve ao meu lado, incentivando-me a seguir em frente com este sonho. Sua dedicação à nossa família, aos seus estudos e a este projeto me deu a força necessária para enfrentar cada desafio com coragem e determinação.

Ao Professor Celso Carvalho, que depositou sua confiança neste trabalho e me orientou com paciência e cuidado, devo minha gratidão. Sua orientação foi essencial para o avanço desta jornada.

Agradeço aos meus filhos, Maria Laura, Maria Luiza e Pedro Lucas, por terem renunciado a tantos dias de lazer e por entenderem a importância do período pelo qual eu estava passando.

Meu colega de curso e amigo, Aduino Hanaoka, tem sido uma fonte constante de companheirismo e apoio no meio acadêmico. Seu incentivo foi uma inspiração, tornando esta jornada mais agradável e enriquecedora.

Finalmente, gostaria de expressar minha gratidão aos meus respeitados professores do Instituto Federal do Amazonas, em particular, Andrezza Carvalho, Daleno Rodrigues, Rubens Aguiar e Thiago Oliveira. Agradeço-lhes pelo apoio, pelas orientações valiosas, pelas palavras de incentivo e pelos momentos memoráveis que passamos juntos.

Resumo da Dissertação apresentada ao PPGEE/UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

## ALGORITMO DE SELEÇÃO DE POLÍTICAS DE MIGRAÇÃO DE MÁQUINAS VIRTUAIS EM FUNÇÃO DE DIFERENTES DEMANDAS DE RECURSOS

Luiz Carlos Silva de Sales

Orientador: Prof. PhD. Celso Barbosa Carvalho

Programa: Pós-Graduação em Engenharia Elétrica

Este estudo sugere a aplicação de um algoritmo para escolher: i) a abordagem de migração da máquina virtual, seja estática ou dinâmica; ii) a estratégia para selecionar o cloudlet de destino na máquina virtual, que pode ser baseada na menor latência entre o cloudlet de origem e um dos cloudlets de destino, a menor distância entre o usuário e um dos cloudlets de destino, ou o cloudlet conectado ao ponto de acesso mais próximo do usuário; e iii) a política de migração da máquina virtual, que pode ser pré-cópia ou pós-cópia, dependendo de parâmetros do sistema como qualidade do sinal, largura de banda, latência e tamanho da máquina virtual.

O algoritmo proposto realiza a migração da máquina virtual enquanto o usuário se move, mantendo a conexão com a internet ativa. Isso resulta na redução do tempo de migração da máquina virtual, do tempo de inatividade da máquina virtual e do volume de dados migrados da máquina virtual, por meio da seleção dinâmica da abordagem de migração, estratégia e política, de acordo com a qualidade dos parâmetros de entrada.

As principais métricas para avaliação são o tempo total de migração da máquina virtual, o tempo de inatividade da máquina virtual e o volume total de dados migrados da máquina virtual. Essas métricas são essenciais para manter a conexão com a internet do usuário durante seu movimento em um cenário de computação em névoa. Os resultados obtidos com este algoritmo superaram os de outros algoritmos, apresentando uma eficiência superior de mais de 6%.

Palavras-chave: Mobilidade Usuário, Computação em Névoa, Máquinas Virtuais, Algoritmo de Seleção.

Abstract of Dissertation presented to PPGEE/UFAM as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

"ALGORITHM FOR SELECTION OF VIRTUAL MACHINE MIGRATION POLICIES  
BASED ON DIFFERENT RESOURCE DEMANDS

Luiz Carlos Silva de Sales

Advisor: Professor PhD Celso Barbosa Carvalho

Program: Post-Graduation in Electrical Engineering

This study suggests the application of an algorithm to choose: i) the virtual machine migration approach, whether static or dynamic; ii) the strategy for selecting the destination cloudlet in the virtual machine, which can be based on the lowest latency between the source cloudlet and one of the destination cloudlets, the shortest distance between the user and one of the destination cloudlets, or the cloudlet connected to the nearest user access point; and iii) the virtual machine migration policy, which can be pre-copy or post-copy, depending on system parameters such as signal quality, bandwidth, latency, and virtual machine size.

The proposed algorithm performs the migration of the virtual machine while the user moves, keeping the internet connection active. This results in the reduction of the virtual machine migration time, the virtual machine downtime, and the volume of data migrated from the virtual machine, through the dynamic selection of the migration approach, strategy, and policy, according to the quality of the input parameters.

The main metrics for evaluation are the total migration time of the virtual machine, the downtime of the virtual machine, and the total volume of data migrated from the virtual machine. These metrics are essential to maintain the user's internet connection during their movement in a fog computing scenario. The results obtained with this algorithm surpassed those of other algorithms, presenting a superior efficiency of over 6%.

Keywords: User Mobility, Fog Computing, Virtual Machines, Selection Algorithm.

## LISTA DE FIGURAS

Figura 1 – Cenário de migração de máquinas virtuais com mobilidade do usuário em um ambiente de computação em névoa. Fonte: Própria. ....	22
Figura 2 - Deslocamento do usuário, <i>handoffs</i> e migração de VMs em computação em nevoa. Fonte: Própria.....	28
Figura 3 – Linha do tempo da mobilidade do usuário. Fonte: Própria.....	29
Figura 4 - Cenário de abordagem do ponto de migração estática. Fonte: Própria. ....	32
Figura 5 - Cenário de abordagem do ponto de migração dinâmica. Fonte: Própria.....	36
Figura 6 - Cenário estratégias de migrações E1. Fonte: Própria. ....	38
Figura 7 - Cenário estratégias de migrações E2. Fonte: Própria.....	39
Figura 8 - Cenário estratégias de migrações E3. Fonte: Própria. ....	41
Figura 9 - Linha do tempo da migração pré-cópia. Fonte: Adaptado de Clark (2020). ....	43
Figura 10 - Linha do tempo da migração pós-cópia. Fonte: Adaptado de Clark (2020).....	45
Figura 11 - Algoritmo de migração de VMs/função_ abordagem [Fluxograma]. ....	53
Figura 12 - Algoritmo de migração de VMs/função_ estratégia [Fluxograma]. ....	56
Figura 13 - Algoritmo de migração de VMs/função_ política [Fluxograma]. ....	59
Figura 14 - Cenário de mobilidade do usuário da cidade de Manhattan. Fonte: Adaptado de Google. (2022). Manhattan. [s.l.]: Google Maps. ....	67
Figura 15 - Cenário Manhattan vetorizado com ERBs, <i>cloudlets</i> e rede de fibra ótica. Fonte: Própria.....	68
Figura 16 - Resultado gráfico de <i>handoffs</i> e migrações de VMs. Fonte: Própria. ....	71
Figura 17 – Avaliação do tempo total de migração. Fonte: Própria.....	72
Figura 18 – Média do tempo total de migração da VM com algoritmo e matriz de seleção. Fonte: Própria. ....	73
Figura 19 - Avaliação do tempo de inatividade da VM. Fonte: Própria. ....	74
Figura 20 – Avaliação do tempo em que a VM está indisponível (inatividade). E1, E2 e E3. Fonte: Própria.....	75
Figura 21 - Avaliação do tempo total de migração. Fonte: Própria. ....	76
Figura 22 - Média do tempo total de migração da VM com algoritmo e matriz de seleção. Fonte: Própria.....	77
Figura 23 - Avaliação do tempo de inatividade da VM. Fonte: Própria. ....	78
Figura 24 – Avaliação do tempo em que a VM está indisponível (inatividade). Fonte: Própria. ....	79
Figura 25 - Avaliação da soma do tamanho das VMs migradas. Fonte: Própria. ....	80
Figura 26 - Avaliação da soma do tamanho das VMs migradas. Fonte: Própria. ....	81



## LISTA DE TABELAS

Tabela 1 - Resumo dos artigos relacionados.....	20
Tabela 2 - descrição dos parâmetros de entrada da função selecionar_abordagem .....	50
Tabela 3 - Etapas da função selecionar_abordagem .....	50
Tabela 4 - Condições para abordagem estática e dinâmica.....	51
Tabela 5 - Etapas da função selecionar_estratégia.....	54
Tabela 6 - Condições para estratégias E1, E1 e E3 .....	55
Tabela 7 - Etapas da função selecionar_política .....	57
Tabela 8 - Matriz de necessidades usuário x abordagem, estratégia e política .....	65
Tabela 9 - Ferramentas e recursos utilizados na criação do algoritmo.....	66

## LISTA DE ABREVIATURAS

b	bits
B	Bytes
Mbps	Mega bits por segundo
COIN	Computação em Rede Integrada
.csv	Arquivo Separado por Vírgula
EB	Estação Base
ERB	Estação Rádio Base
IoT	Internet das Coisas
m	metro
MDP	Processo de Decisão Markov
ms	milissegundos
NFV	Virtualização de Função de Rede
PA	Ponto de Acesso
PLC	Comunicação de linha de energia
QoE	Qualidade de Experiência
QoS	Qualidade de Serviço
rad	Radianos
RAN	Rede de Acesso por Rádio
s	Segundos
TI	Tecnologia da Informação
VM	Máquina Virtual

## SUMÁRIO

CAPITULO 1 .....	9
1. INTRODUÇÃO .....	9
1.1 OBJETIVO GERAL .....	11
1.2 OBJETIVOS ESPECÍFICOS .....	12
1.3 JUSTIFICATIVA .....	12
1.4 DESENVOLVIMENTO DO TRABALHO .....	13
CAPITULO 2 .....	15
2. TRABALHOS RELACIONADOS .....	15
CAPITULO 3 .....	21
3. MIGRAÇÃO DE MÁQUINAS VIRTUAIS .....	21
3.1 Arquitetura .....	21
3.2 Virtualização .....	23
3.3 Computação em névoa .....	23
3.4 Mobilidade do usuário .....	24
3.5 Qualidade do serviço (QoS) .....	25
3.6 Arquitetura em camadas .....	25
3.7 Cenário com migração de VM e <i>handoff</i> .....	27
3.8 Abordagens, estratégias e políticas de migração de máquinas virtuais .....	31
3.8.1 Abordagem do ponto de migração estática .....	31
3.8.2 Abordagem do ponto de migração dinâmica .....	35
3.8.3 Estratégia menor latência .....	37
3.8.4 Estratégia menor distância entre usuário e um dos <i>cloudlets</i> de destino .....	39
3.8.5 Estratégia <i>cloudlet</i> conectado ao ponto de acesso mais próximo do usuário no momento da abordagem. ....	40
3.8.6 Política de migração complete container .....	41
3.8.7 Política de migração pós-cópia .....	44
CAPITULO 4 .....	47
4. METODOLOGIA .....	47
4.1 Função do algoritmo de migração de VM .....	47
4.1.1 Critérios de tratamento dos parâmetros de entrada .....	48
4.1.2 Normalização dos parâmetros de entrada .....	49
4.1.3 Descrição da função <i>selecionar_abordagem</i> .....	50

4.1.4 Descrição da função <i>selecionar_estratégia</i> .....	54
4.1.5 Descrição da função <i>selecionar_política</i> .....	56
4.2 Parâmetros Avaliados.....	59
4.2.1 Estimativa do tempo de migração de VM usando a técnica de pré-cópia.....	60
4.2.2 Estimativa do tempo de migração de VM usando a técnica de pós-cópia .....	61
4.2.3 Estimativa de inatividade na política de pós-cópia .....	62
4.2.4 Tamanho das VMs enviadas na política de pré-cópia .....	63
4.3 Matriz de seleção para abordagem, estratégia e política .....	64
4.4 Ambiente de teste.....	65
4.4.1 Ferramentas e recursos utilizadas na criação e validação do algoritmo .....	65
4.4.2 Cenário Manhattan .....	66
4.5 Procedimentos na validação do algoritmo.....	68
CAPITULO 5.....	70
5. RESULTADOS.....	70
5.1 RESULTADOS E DESCRIÇÃO.....	70
5.1.1 Avaliação do tempo total de migração da VM para alta demanda de largura de banda.....	71
5.1.2 Avaliação do tempo em que a VM está indisponível (inatividade) para alta demanda de largura de banda.....	73
5.1.3 Avaliação do tempo total de migração da VM para movimento rápido.....	75
5.1.4 Avaliação do tempo em que a VM está indisponível (inatividade) para movimento rápido .....	77
5.1.5 Avaliação da soma do tamanho da VM migrada para alta demanda de largura de banda .....	79
5.2 ANÁLISE DOS RESULTADOS.....	81
CAPITULO 6.....	83
6. CONCLUSÕES FINAIS.....	83
REFERÊNCIAS.....	84

## 1. INTRODUÇÃO

Em 2023, o 5G<sup>1</sup> ultrapassou 1 bilhão de usuários de *smartphones* globalmente e, na América Latina, chegou a 724 milhões, com uma estimativa de chegar a 759 milhões em 2025, segundo Ericsson (2021). O tráfego de dados da rede móvel cresceu 40% entre o quarto trimestre de 2021 e o quarto trimestre de 2022, atingindo 118 estações base (EB) instaladas por mês e dobrou em apenas dois anos. Isso foi principalmente devido ao crescimento de usuários de *smartphones* e ao aumento da visualização de conteúdo de vídeo, o que resultou em um aumento significativo de usuários e dispositivos móveis e, conseqüentemente, aumentou a quantidade de dados processados na nuvem e ocasionou maior latência da rede.

Isso mostra como o crescimento do uso de dispositivos móveis e do consumo de dados está afetando a infraestrutura de rede e criando desafios para gerenciar a latência da rede. A computação em névoa pode ser uma solução para lidar com esses desafios, pois permite o processamento na borda da rede, mais perto do usuário final, para melhorar a latência e fornecer suporte à mobilidade do usuário. Isso pode ser especialmente útil em aplicações sensíveis à latência que exigem respostas rápidas.

A migração de máquinas virtuais é uma técnica que permite mover uma máquina virtual de um servidor físico para outro sem interromper o serviço. Isso pode ser útil em um ambiente de computação em névoa para atender à mobilidade do usuário e diminuir a latência da rede de computação em névoa. Com o crescimento do uso de dispositivos móveis e do consumo de dados, como mencionado anteriormente, a infraestrutura de rede está enfrentando desafios para gerenciar a latência da rede. A computação em névoa pode ser uma solução para lidar com esses desafios, pois permite o processamento na borda da rede, mais perto do usuário final, para melhorar a latência e fornecer suporte à mobilidade.

A computação em nuvem é uma tecnologia que permite o acesso a recursos de computação sob demanda pela Internet com preço conforme o uso. Ela consiste em grandes *data centers* que podem ser localizados em todo o mundo e é mais poderosa do que a névoa em

---

<sup>1</sup> 5G é a quinta geração de tecnologia padrão para redes celulares, que as empresas de telefonia celular começaram a implantar em todo o mundo em 2019, e é o sucessor da tecnologia 4G que fornece conectividade para a maioria dos telefones móveis atuais.

relação aos recursos de processamento e capacidade de armazenamento. No entanto, possui grande dependência de conexões com a internet (ORACLE, 2023).

A computação em névoa, por outro lado, é um paradigma que visa trazer algumas capacidades da computação em nuvem para a borda da rede, mais perto do usuário final, para melhorar a latência, aumentar a conscientização sobre a localização e fornecer suporte à mobilidade. A névoa basicamente estende a computação em nuvem e seus serviços até a borda da rede, aproximando suas vantagens e benefícios. Ela tem uma maior proximidade com os usuários finais e maior distribuição geográfica. Além disso, possui outras vantagens como a baixa latência e menor dependência de conexões de internet na transmissão de dados.

Em resumo, enquanto a computação em nuvem é centralizada e consiste em grandes *data centers*, a computação em névoa é descentralizada e distribuída na borda da rede. Ambas fornecem armazenamento e processamento de dados, entre outras soluções, mas têm diferenças importantes em termos de localização, distribuição geográfica e dependência de conexões com a internet.

No entanto, devido às limitações de recursos, heterogeneidade de recursos, natureza dinâmica e imprevisibilidade do ambiente de névoa, o gerenciamento de recursos é um dos problemas desafiadores a serem considerados num ambiente de computação em névoa (SHAKARAMI *et al.*, 2022). O gerenciamento de recursos pode ser classificado em seis subáreas: provisionamento de recursos, colocação de aplicativos, agendamento, alocação de recursos e balanceamento de carga. Existem várias abordagens para melhorar o gerenciamento de recursos na computação em névoa. Por exemplo, Fahimullah, Ahvar e Trocan (2022) fornece uma revisão das abordagens de gerenciamento de recursos baseadas em aprendizado de máquina em um ambiente de névoa. Essas abordagens podem ajudar a prever o comportamento dinâmico da névoa e gerenciar os recursos adequadamente.

Além disso, a computação em névoa é necessária para operar autonomamente para garantir serviços ininterruptos mesmo quando há conexões não confiáveis com a nuvem, mas também é necessário integrar-se totalmente com a nuvem quando a conectividade total dos recursos é restaurada. Isso exige o desenvolvimento de mecanismos eficazes de gerenciamento e orquestração de recursos para garantir o desempenho aceitável das aplicações e serviços enquanto aproveita as capacidades da nuvem (DLAMINI; VENTURA, 2019).

Um dos mecanismos que pode ser usado para lidar com os desafios da computação em névoa é a migração de máquinas virtuais, que consiste em transferir uma máquina virtual de um servidor para outro sem interromper o serviço. A migração de máquinas virtuais pode

ser usada em conjunto com a computação em névoa para mover máquinas virtuais para dispositivos de névoa mais próximos do usuário, reduzindo a latência e melhorando o desempenho. No entanto, a migração de máquinas virtuais também envolve custos e riscos, como o tempo de migração, o consumo de recursos e a perda de dados.

Por exemplo, Goncalves (2018) apresentou uma política de migração proativa de máquinas virtuais em ambientes de névoa a fim de melhorar o gerenciamento dos recursos computacionais utilizados pelos veículos conectados a essa infraestrutura. As simulações sugerem que a utilização de previsões sobre o trajeto futuro do veículo pode melhorar o gerenciamento de recursos da névoa, mantendo a máquina virtual do usuário em dispositivos de névoa tão próximos quanto possível do trajeto do automóvel. A solução apresentada reduz o número de migrações realizadas durante o trajeto do usuário sem prejudicar o tempo de resposta da máquina virtual alocada na névoa (GONCALVES *et al.*, 2018).

Considerando que a migração de máquinas virtuais (VM) apresenta-se como uma solução para dar suporte à mobilidade do usuário, discute-se as políticas de migração pré-cópia e pós-cópia, as abordagens do ponto de migração estática e dinâmica e as estratégias para seleção do *cloudlet* de destino baseadas na menor latência entre o *cloudlet* de origem e os *cloudlets* de destino, menor distância entre o usuário e um dos *cloudlets* de destino e para o *cloudlet* conectado ao ponto de acesso (PA) mais próximo do usuário do momento da abordagem, considerando diferentes disponibilidades de recursos de rede.

Finalmente, após investigar-se os impactos da mobilidade do usuário na qualidade do serviço (QoS), com o uso do simulador MobFogSim<sup>2</sup> e outras referências bibliográficas, propõe-se um algoritmo de seleção dinâmica de políticas de migração de máquinas virtuais baseado na mobilidade do usuário, na necessidade do usuário e nos recursos disponíveis com o intuito reduzir o tempo total de migração, o tempo de inatividade da VM e o tamanho dos dados migrados.

## 1.1 OBJETIVO GERAL

Desenvolver um algoritmo de seleção de políticas de migração máquinas virtuais em função de diferentes demandas de recursos.

---

<sup>2</sup> O MobFogSim é um simulador que estende o iFogSim para modelar aspectos relacionados à mobilidade de dispositivos e migração de VM/container em computação em nuvem e névoa.

## 1.2 OBJETIVOS ESPECÍFICOS

- 1) Revisar as principais bibliografias existentes;
- 2) Apresentar os fundamentos básicos sobre migração de máquinas virtuais;
- 3) Desenvolver um algoritmo para selecionar as políticas de migração de máquinas virtuais;
- 4) Simular a migração de máquinas virtuais em um ambiente de computação em névoa considerando a mobilidade do usuário;
- 5) Analisar os resultados;

## 1.3 JUSTIFICATIVA

A migração de máquinas virtuais (VMs) é um tópico de grande importância teórica e prática na computação em névoa. Do ponto de vista do usuário, a migração de VMs oferece disponibilidade contínua, permitindo que os serviços continuem funcionando sem interrupções ou atrasos. Isso é especialmente crucial em ambientes onde a continuidade do serviço é uma necessidade, como em operações de negócios críticas ou infraestruturas de missão crítica. Além disso, a migração de VMs pode levar a um desempenho melhorado, com latências reduzidas e maior velocidade nas operações.

Do ponto de vista do sistema, a migração de VMs permite uma utilização mais eficiente dos recursos de computação e de rede. Isso pode resultar em economia de energia, o que é uma consideração importante em *data centers* de grande escala e em ambientes de computação em nuvem e névoa. A migração de VMs também pode facilitar o balanceamento de carga, ajudando a garantir que nenhum recurso seja sobrecarregado enquanto outros estão ociosos.

Existem vários cenários em que a migração de VMs pode ser particularmente útil em um ambiente urbano. Por exemplo, na automação industrial, a migração de VMs pode permitir que as operações continuem sem interrupções, mesmo quando ocorrem falhas de *hardware* ou são necessárias manutenções de rotina. Isso pode resultar em maior eficiência e menor tempo de inatividade.

Na mobilidade urbana, a migração de VMs pode permitir que os usuários acessem serviços de forma contínua enquanto se movem pela cidade. Isso pode ser particularmente útil para serviços como navegação por GPS, *streaming* de mídia ou jogos online.



Nos serviços públicos, a migração de VMs pode permitir a continuidade dos serviços mesmo em face de interrupções ou picos de demanda. Isso pode ser crucial para serviços como energia, água e transporte público.

Finalmente, no teletrabalho e na educação à distância, a migração de VMs pode permitir que os usuários acessem seus ambientes de trabalho ou de aprendizagem de qualquer lugar, sem interrupções ou atrasos. Isso pode ser especialmente útil em situações onde os usuários precisam se deslocar frequentemente ou onde a conectividade à internet pode ser instável.

O exemplo prático de aplicação da migração de máquinas virtuais (VMs) no contexto da mobilidade urbana é descrito a seguir:

- O cenário descrito é o de um usuário utilizando um serviço de *streaming* de música em seu *smartphone* enquanto se desloca pela cidade. O problema que surge nesse cenário é a latência na comunicação, que pode resultar em interrupções ou atrasos na transmissão da música.
- A solução proposta é a migração de VMs. Ao migrar a VM que está fornecendo o serviço de *streaming* de um servidor para outro mais próximo do usuário, é possível reduzir a latência e melhorar a qualidade do serviço.
- A resiliência do sistema também é aprimorada através da migração de VMs. Em caso de falha de um servidor, a VM pode ser rapidamente migrada para outro servidor, garantindo a continuidade do serviço de *streaming*.
- Portanto, a migração de VMs oferece benefícios significativos tanto do ponto de vista do usuário, proporcionando uma melhor experiência de *streaming*, quanto do ponto de vista do sistema, permitindo uma utilização mais eficiente dos recursos e melhorando a resiliência do sistema.

Em resumo, a migração de VMs é uma ferramenta poderosa que pode melhorar a disponibilidade e o desempenho do usuário, ao mesmo tempo em que permite uma utilização mais eficiente dos recursos do sistema. Seu uso em ambientes urbanos pode ter um impacto significativo em uma variedade de áreas, desde a automação industrial até a mobilidade urbana, os serviços públicos e o teletrabalho e a educação à distância.

## 1.4 DESENVOLVIMENTO DO TRABALHO

Este trabalho está organizado nos seguintes capítulos:

- Capítulo I: Introdução;
- Capítulo II: Trabalhos relacionados;
- Capítulo III: Migração de máquinas virtuais;
- Capítulo IV: Metodologia;
- Capítulo V: Resultados;
- Capítulo VI: Conclusões finais.

O Capítulo I apresenta um contexto para o trabalho, situando o mesmo frente a outras pesquisas da área e delineando as diferenças e semelhanças. São apresentados também os objetivos gerais, específicos e justificativa.

O Capítulo II apresenta os principais trabalhos que foram publicados na literatura sobre o tema abordado nessa dissertação.

O Capítulo III apresenta os principais fundamentos teóricos utilizados no desenvolvimento deste trabalho.

No Capítulo IV são descritas as funções de abordagem, estratégias e políticas de migração de VM. Também, é descrito o ambiente de teste e como foi realizada a validação do algoritmo;

No Capítulo V são apresentados os resultados da pesquisa através de gráficos e Tabelas e realizada a análise destes;

Por fim, no Capítulo VI é apresentado as conclusões finais da pesquisa e também é apresentado sugestões para trabalhos futuros.

---

## 2. TRABALHOS RELACIONADOS

Diante do cenário definido, há necessidade de avaliar as políticas de migração de máquinas virtuais em ambiente de computação em névoa, considerando a mobilidade do usuário, incluindo abordagens do ponto de migração e estratégias de migração para melhorar o uso dos recursos de computação em névoa e a experiência do usuário (QoS).

A metodologia utilizada incluiu a revisão de literatura relevante sobre o assunto, a configuração de um cenário de simulação para avaliar o impacto da mobilidade do usuário na latência durante a migração de máquinas virtuais, a análise dos dados coletados e a apresentação de uma solução baseada nos resultados obtidos. Essa abordagem permitiu uma compreensão mais profunda do problema e a proposição de soluções eficazes para minimizar o impacto da mobilidade do usuário na latência durante a migração de máquinas virtuais em ambientes de computação em névoa.

Vários estudos têm explorado a migração de máquinas virtuais em ambientes de computação em névoa. Este parágrafo apresenta os principais objetivos e técnicas dos trabalhos relacionados à migração de máquinas virtuais. Os autores desses tópicos buscam demonstrar conceitos, vantagens e possibilidades da computação em névoa. A computação em névoa é um paradigma que estende os serviços de nuvem para a borda da rede numa escala amplamente distribuída. Ela é vista como uma solução para atender requisitos como mobilidade, localidade e baixa latência em aplicações de Internet das Coisas (IoT).

Bittencourt et al. (2015), apresenta a computação de névoa como uma tecnologia emergente no campo dos serviços de rede que permite a transferência de dados entre dispositivos para realizar atividades. A computação de névoa é um conceito estendido da computação em nuvem que funciona entre a Internet das Coisas (IoT) e os *data centers* em nuvem para reduzir as lacunas de comunicação. O autor destaca os benefícios da computação de névoa, como a diminuição da latência e o baixo congestionamento da rede.

Complementando essa ideia, Lopes et al. (2017) discute a importância da baixa latência em aplicativos de TI (tecnologia da informação) e apresenta o MyiFogSim<sup>3</sup>, uma extensão do

---

<sup>3</sup> O MyiFogSim é uma ferramenta de simulação por eventos discretos desenvolvida especificamente para cenários de computação em névoa, com atenção especial a cenários com mobilidade de dispositivos.

iFogSim que suporta a mobilidade por meio da migração de máquinas virtuais entre *cloudlets* para manter as latências baixas em face da mobilidade dos usuários.

Velasquez et al. (2018), descreve o funcionamento da computação em névoa na borda da rede em torno de um nó de névoa e a rede de dispositivos do usuário conectados a ele. O autor apresenta o OpenFog RA (OpenFog Reference Architecture), uma arquitetura leve e confiável que permite a construção de uma rede em névoa mais inteligente para descarregamento de computação.

Goncalves *et al.* (2018) apresenta uma política de migração proativa de máquinas virtuais para aplicações móveis em ambientes de computação em névoa. A solução utiliza previsões sobre o trajeto futuro do veículo para manter a máquina virtual do usuário em dispositivos de névoa tão próximos quanto possível do trajeto do veículo.

Khaleel e Zhu. (2021) propõe um algoritmo multi-etapas para alocação de máquinas virtuais chamado “E-PRWA (relação desempenho-potência aprimorada para aplicações de fluxo de trabalho)” em ambientes de computação em nuvem/névoa. O algoritmo busca alcançar um equilíbrio entre o desempenho do nó e o consumo de energia.

Rosário et al. (2018) apresenta uma abordagem para migração de serviços de nuvem para nós de névoa multicamadas para disseminação de multimídia com suporte a Qualidade de Experiência (QoE). A principal contribuição do trabalho é a descrição dos impactos e benefícios operacionais associados à migração de serviços da nuvem para a computação em névoa multicamadas para distribuição de vídeo com suporte a QoE.

Puliafito *et al.* (2020) apresenta um modelo inovador para o posicionamento e migração de máquinas virtuais em ambientes de computação em névoa. O modelo é baseado na previsão de mobilidade dos usuários móveis conectados à infraestrutura *Fog* e utiliza um simulador chamado MobFogSim para avaliar as características da rede, dos dispositivos e dos aplicativos.

Tinini *et al.* (2020) apresenta um simulador baseado em SimPy<sup>4</sup> para avaliar o desempenho de arquiteturas de rede de acesso por rádio (RAN) híbridas em nuvem e névoa para redes 5G. Os resultados mostram que a utilização de uma arquitetura híbrida pode melhorar significativamente o desempenho da rede em comparação com uma abordagem puramente em nuvem ou puramente em névoa.

---

<sup>4</sup> SimPy é uma biblioteca de comandos que conferem ao Python o poder de construir modelos de eventos discretos.

Goncalves et al. (2020) propõe uma abordagem para o fatiamento dinâmico de rede em ambientes de computação em névoa para usuários móveis. A abordagem usa o simulador MobFogSim para modelar o comportamento dos usuários, dos recursos e dos aplicativos na borda da rede.

He (2021) apresenta uma abordagem para o gerenciamento de migração de máquinas virtuais e containers em ambientes de computação na borda e na nuvem com redes. A abordagem usa técnicas de otimização e aprendizado de máquina para decidir quando, onde e como migrar as cargas de trabalho.

Venâncio et al. (2022) propõe um novo paradigma de computação em rede chamado NFV-COIN, que combina as tecnologias de virtualização de funções de rede (NFV) e computação em rede integrada (COIN). O NFV-COIN permite implantar e executar funções de rede virtualizadas (VNFs) em dispositivos de rede, aproveitando os recursos computacionais disponíveis na rede. O artigo mostra que o NFV-COIN pode melhorar a eficiência, a escalabilidade e a flexibilidade da computação em rede, bem como reduzir o custo e a complexidade da infraestrutura.

Fraser e Fitchett (2022) apresenta um estudo abrangente sobre a migração ao vivo de máquinas virtuais, que é o processo de mover uma máquina virtual em execução de um hospedeiro para outro sem interromper o serviço. O artigo explica os conceitos e os benefícios da migração ao vivo, bem como os desafios e as limitações envolvidas.

Tang *et al.* (2019) propõe algoritmos e arquitetura de migração de containers para suportar tarefas de mobilidade com vários requisitos de aplicação em computação em névoa. Os algoritmos são realizados a partir de três aspectos: 1) considera-se que as tarefas de aplicação móvel podem ser hospedadas em um contêiner de um nó de neblina correspondente que pode ser migrado, levando em consideração o atraso na comunicação e o consumo de energia computacional; 2) modela-se essa estratégia de migração de containeres como espaços do Processo de Decisão Markov (MDP) multidimensionais e 3) implementa-se o modelo e os algoritmos como um sistema protótipo de migração de containeres e testa-se sua viabilidade e desempenho.

Gonçalves, Bittencourt e Madeira. (2019) apresenta uma análise do impacto de uma baixa acurácia na predição de mobilidade do usuário para otimizar o processo de migração de aplicações em um ambiente de computação em névoa. Resultados de simulações indicaram que o uso de predição de mobilidade pode reduzir até 45% o número de migrações, mas um erro de

cálculo da posição do usuário pode aumentar a latência média experimentada por ele em até 30%.

Moraes *et al.* (2020) propõe um mecanismo de migração de serviços ciente da localização da nuvem e/ou névoa, em que os serviços são encapsulados em containers e máquinas virtuais e enviados através de redes ópticas elásticas com multiplexação espacial.

Torre *et al.* (2020) apresenta uma heurística evolutiva dinâmica para o posicionamento de máquinas virtuais (VMs) em *data centers* em nuvem, considerando múltiplos objetivos, como minimização do consumo de energia, maximização da utilização de recursos e minimização do tempo de resposta.

Matos e Maziero (2021) apresenta uma solução para a gerência da migração de máquinas virtuais em ambientes de computação em nuvem, destacando um modelo de arquitetura que possibilita atingir tal propósito. Simulações possibilitaram a avaliação da arquitetura, destacando que a adoção das ações de gerência da arquitetura torna os resultados de uma migração de VMs mais perenes, diminuindo novas instabilidades que, usualmente, indicariam a necessidade de novas migrações.

Chen *et al.*, (2021) discute a migração de máquinas virtuais (VMs) e como a relevância do serviço pode afetar a eficiência da migração. Ele sugere que a definição de fatores de relevância do serviço pode ajudar a quantificar o grau de relevância do serviço, construir grupos de migração de VMs e reduzir efetivamente a sobrecarga de comunicação entre servidores durante a migração.

Belgacem; Mahmoudi; e Ferrag (2023) abordam a questão da migração de VMs aplicando um modelo de aprendizado de máquina para reduzir o número de migrações de VMs e o consumo de energia. O algoritmo proposto é baseado na melhoria do processo de migração e seleção de VMs.

Solomon e Alley, (2014) discutem o desempenho do balanceamento de carga do DRS Score. Eles mencionam que a pontuação é um valor determinante para o sistema DRS e explica como o DRS calcula a pontuação de cada VM em um *cloudlet* específico.

A maioria dos trabalhos apresentados estão focados em migração de máquinas virtuais e containers em ambientes de computação em nuvem e névoa, com o objetivo de otimizar o

desempenho em termos de latência, consumo de energia e custo de migração. Eles propõem diferentes abordagens, algoritmos e arquiteturas para alcançar esse objetivo.

Nesse trabalho propõe-se a criação de um algoritmo de seleção da abordagem, e estratégia e política a ser adotada com base em parâmetros do sistema, tais como, qualidade do sinal, largura de banda, latência e tamanho da VM.

O grande diferencial desse trabalho (Sales, 2023) é apresentar um algoritmo que realiza a migração da VM com base em parâmetros de entrada com o objetivo de reduzir o tempo total de migração da VM, o tempo de inatividade da VM e o tamanho da VM migrada em um ambiente de computação em névoa e o usuário em movimento.

Um algoritmo para realizar o processo de migração de máquinas virtuais em ambientes de computação em nuvem e névoa pode adaptar-se dinamicamente às mudanças nas condições do ambiente de várias maneiras. Por exemplo, ele pode monitorar continuamente as condições do ambiente, como a carga de trabalho, a disponibilidade de recursos e a mobilidade dos usuários, e ajustar suas decisões de migração com base nessas informações. Além disso, o algoritmo pode usar técnicas de aprendizado de máquina para aprender com os dados históricos e prever as condições futuras do ambiente, permitindo que ele tome decisões proativas para otimizar o desempenho. Finalmente, o algoritmo pode ser projetado para ser flexível e capaz de lidar com diferentes cenários e requisitos, permitindo que ele se adapte às mudanças nas condições do ambiente sem a necessidade de intervenção humana.

Na Tabela 1 abaixo é apresentado o resumo dos artigos relacionados.

Tabela 1 - Resumo dos artigos relacionados.

	Ano / Autor/Título	Cenário	Avaliou a disponibilidade recursos?	Considerou a mobilidade?	Qual abordagem avaliou?	Qual estratégia avaliou?	Qual política avaliou?
1	Lopes et al. (2017). “MyifogSim: A simulator for virtual machine migration in fog computing.”	Computação em névoa	Avaliou	Considerou	Estática	Não avaliou	Pós-cópia
2	Rosário et al. (2018). “Service migration from cloud to multitier fog nodes for multimedia dissemination with QoE support.”	Migração dos serviços de nuvem para névoa	Avaliou	Considerou	Estática	Não avaliou	Não
3	Bittencourt et al. (2015). “Towards Virtual Machine Migration in Fog Computing”.	Computação em névoa	Avaliou	Considerou	Estática	Não avaliou	Não
4	Puliafito et al., (2020). “MobFogSim: Simulation of mobility and migration for fog computing.”	Computação em névoa	Avaliou	Considerou	Estática dinâmica e	Menor latência entre <i>cloudlets</i>	Pós-cópia
5	Tinini et al., (2020). “5GPy: A SimPy-based simulator for performance evaluations in 5G hybrid Cloud-Fog RAN architectures”.	Computação em nuvem e névoa	Avaliou	Não considerou	Estática	Não avaliou	Não
6	Fraser e Fitchett (2022). “Live Migration of Virtual Machines.”	Computação em nuvem	Avaliou	Não considerou	Não avaliou	Menor latência entre <i>cloudlets</i>	Não
7	Goncalves et al, (2020). “Dynamic network slicing in fog computing for mobile users in MobFogSim.”	Computação em nuvem e névoa	Avaliou	Considerou	Não avaliou	Não avaliou	Não
8	Sales, (2023) “Algoritmo de seleção de políticas de migração de máquinas virtuais em função de diferentes demandas de recursos”.	Névoa	Avaliou	Considerou	Estática dinâmica e	Menor latência entre <i>cloudlets</i> , menor distância entre usuário e <i>cloudlet</i> e PA	Pré-cópia e pós-cópia



### 3. MIGRAÇÃO DE MÁQUINAS VIRTUAIS

Este capítulo apresenta informações sobre as ferramentas utilizadas para conduzir a pesquisa, incluindo a descrição da arquitetura e os algoritmos propostos nesta dissertação.

#### 3.1 Arquitetura

A Figura 1, cujo algoritmo em MatLab<sup>5</sup> e apresentado no Apêndice A, ilustra uma estrutura de computação em névoa em ambiente urbano vetorizada composta por *cloudlets*, ERBs (pontos de acesso sem fio) e rede de fibra ótica que fornecem serviços descentralizados para usuários e dispositivos em movimento.

O gráfico abaixo apresenta as latitudes (eixo x) e longitudes (eixo y) do centro da cidade de Manhattan, 67 ERBs (triângulos) divididas em triângulos equidistantes (700m), com raio de cobertura de 500 metros (circunferências). 15 dessas ERBs também desempenham a função de computação em névoa (*cloudlets*) distribuídas entre as demais ERBs. Todas as *cloudlets* estão interconectadas com uma rede de fibra ótica e cada ERB está ligada ao *cloudlet* mais próximo com fibra ótica. Cada ERB tem uma área de cobertura de rede sem fio de 500 metros de raio.

---

<sup>5</sup> MATLAB trata-se de um *software* interativo de alta performance voltado para o cálculo numérico. Integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil de usar.

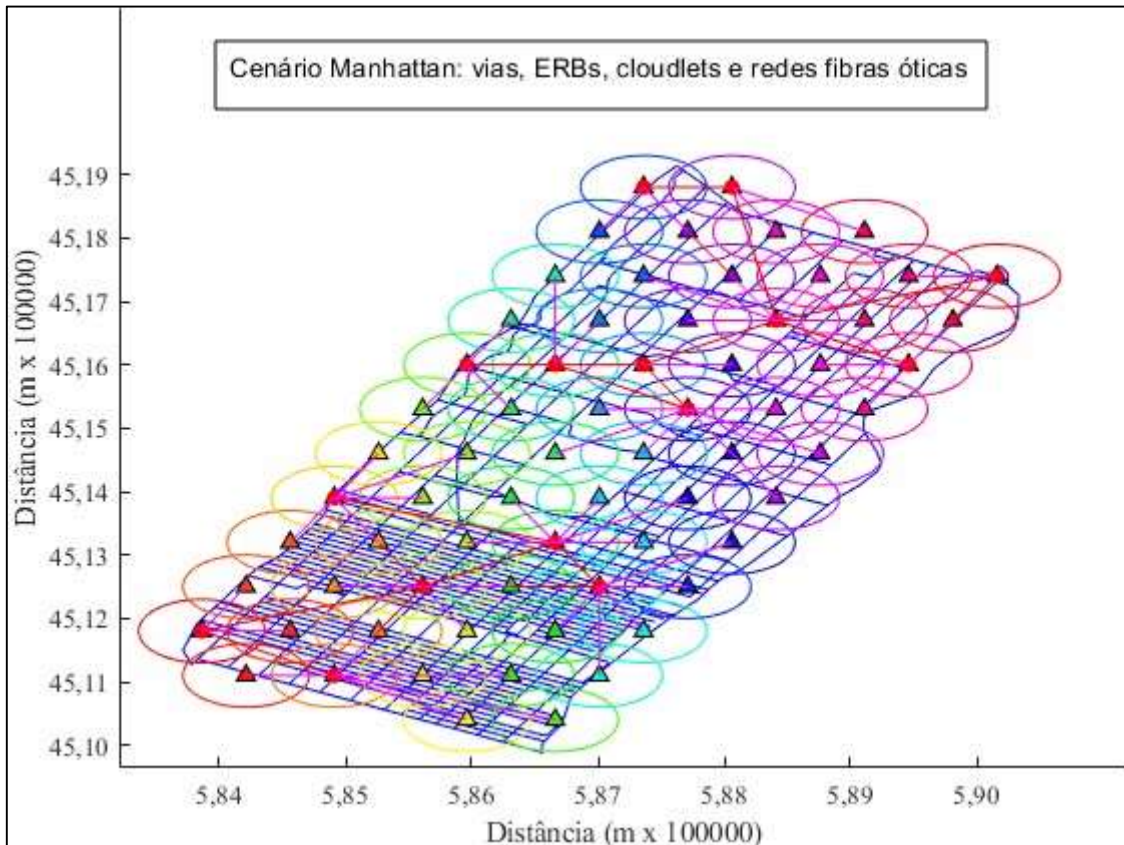


Figura 1 – Cenário de migração de máquinas virtuais com mobilidade do usuário em um ambiente de computação em névoa. Fonte: Própria.

De acordo com Venâncio et al. (2022), a virtualização de redes permite a virtualização de serviços em redes de computadores, facilitando a implementação de novos serviços pelos provedores. Isso é complementado por Ahmad et al. (2017), que afirmam que a computação em nuvem fornece recursos de *hardware* e *software* para fornecer serviços de computação em nuvem. Além disso, Chin et al. (2019) definem a internet das coisas (IoT) como uma rede de dispositivos que permitem a troca e análise de dados e é frequentemente usada para monitorar e controlar dispositivos remotamente. Para lidar com os desafios relacionados ao suporte à mobilidade e orquestração de recursos/serviços, Velasquez et al. (2018) afirmam que a computação em névoa traz funções de computação, armazenamento, controle e rede para a borda da rede, diminuindo a latência da rede e descongestionando o tráfego de dados para a nuvem.

A internet das coisas (IoT) é usada para monitorar e controlar dispositivos remotamente, aumentar a produtividade e reduzir custos. As aplicações da IoT incluem monitoramento de tráfego, temperatura, equipamentos e qualidade do ar. A computação em nuvem permite o armazenamento e análise segura dos dados coletados e aplicativos específicos podem ser usados para gerenciar e controlar dispositivos remotamente. No entanto, o aumento das demandas da

IoT aumentou a quantidade, o tráfego, o armazenamento e processamento de dados, aumentando a latência da rede. Para lidar com esses desafios, a computação em névoa foi criada para trazer serviços críticos para a borda da rede de forma descentralizada. A computação em névoa compreende funções de computação, armazenamento, controle e rede e tem vantagens como diminuição da latência da rede e descongestionamento do tráfego de dados para a nuvem.

### 3.2 Virtualização

A virtualização é a tecnologia que permite criar representações virtuais de servidores, armazenamento, redes e outras máquinas físicas. O *software* virtual imita as funções do *hardware* físico para executar várias máquinas virtuais simultaneamente em uma única máquina física. Isso permite que as empresas usem seus recursos de *hardware* com eficiência e obtenham maiores retornos de seus investimentos.

Em um ambiente de computação em névoa, a virtualização pode ser usada para estender os recursos computacionais de forma integrada com elementos de rede como *switches*, roteadores e *gateways*. O processamento ocorre em um dispositivo inteligente gerenciando processos e serviços de maneira local, assim reduzindo o fluxo de dados enviados para nuvem

### 3.3 Computação em névoa

A computação em névoa é uma infraestrutura de computação descentralizada que traz os benefícios da nuvem para mais perto da fonte de dados. Embora seja frequentemente associada à Cisco, a computação em névoa é um conceito aberto à comunidade em geral. Ela complementa a computação em nuvem, permitindo análises de curto e longo prazo em diferentes níveis e enfrentando desafios como privacidade, segurança e conformidade. A computação em névoa é usada em aplicações como redes inteligentes, cidades inteligentes e redes definidas por *software*. A principal diferença entre a computação em nuvem e a computação de borda é onde a inteligência e o poder de computação estão localizados. Enquanto a computação em nuvem é mais escalável e oferece uma melhor visão geral da rede, a computação de borda reduz os pontos de falha ao permitir que cada dispositivo opere de forma independente.

Um exemplo comum é o controle de tráfego, que usa sensores conectados a redes celulares e recursos de computação próximos às torres de celular para analisar os dados de

tráfego em tempo real e ajustar os sinais de trânsito. Esse conceito também pode ser aplicado aos veículos autônomos, que são dispositivos de ponta com grande poder de computação a bordo. Eles podem enviar dados em tempo real para o sistema de controle de tráfego, usando a conectividade quando disponível, e receber orientações para dirigir com segurança. A computação em névoa tem vantagens como economizar largura de banda, melhorar o tempo de resposta e ser independente da rede, mas também tem desvantagens como depender da localização física e estar sujeita a problemas de segurança.

A computação em névoa tem seus desafios, como a dependência da localização física, os riscos de segurança, os custos de inicialização e a falta de uma definição clara. Ela é usada para aplicações da Internet das Coisas (IoT) que não podem usar a computação em nuvem por causa da latência e do custo. Ela consiste em uma rede de nós que processam dados de dispositivos IoT em tempo real e enviam resumos analíticos para a nuvem. Ela precisa de uma conexão rápida entre os dispositivos e os nós, o que pode ser feito com fio ou sem fio. O 5G é uma boa opção para recursos móveis ou remotos que precisam de conectividade de alta velocidade.

### 3.4 Mobilidade do usuário

A mobilidade do usuário é um conceito que se refere à capacidade de um indivíduo de se mover livremente e com facilidade em um ambiente, seja ele físico ou virtual. Existem vários modelos de mobilidade que tentam prever e entender os padrões de movimento dos usuários, levando em consideração fatores como a infraestrutura de transporte, a disponibilidade de serviços e a localização geográfica. Alguns dos fatores que influenciam a mobilidade do usuário incluem o planejamento urbano, o acesso a serviços de transporte, a acessibilidade e a qualidade dos serviços, bem como fatores socioeconômicos e culturais.

A rede 5G é a mais recente tecnologia de comunicação móvel, que promete revolucionar a conectividade e oferecer uma variedade de novas possibilidades. Uma das principais características do 5G é sua baixa latência, ou seja, o tempo de resposta entre o envio e o recebimento de dados é extremamente baixo, permitindo uma comunicação praticamente em tempo real. Isso é especialmente importante para aplicações que exigem alta velocidade e confiabilidade, como carros autônomos e cirurgias remotas. Além disso, a arquitetura da rede 5G foi projetada para tornar a transição mais suave quando os usuários mudam de célula para

célula, melhorando a experiência geral do usuário e limitando interrupções na transferência de dados e perda de sinal.

### 3.5 Qualidade do serviço (QoS)

Qualidade do Serviço (QoS) é um conceito que se refere à capacidade de uma rede de fornecer um serviço com um nível de desempenho garantido. Isso inclui métricas como largura de banda, atraso, *jitter*<sup>6</sup> e perda de pacotes. Os parâmetros de QoS são usados para definir os requisitos de desempenho para diferentes tipos de tráfego, como voz, vídeo e dados. As técnicas de avaliação de QoS incluem monitoramento em tempo real, testes de desempenho e análise estatística. O objetivo é garantir que a rede possa atender às expectativas dos usuários e fornecer um serviço confiável e de alta qualidade.

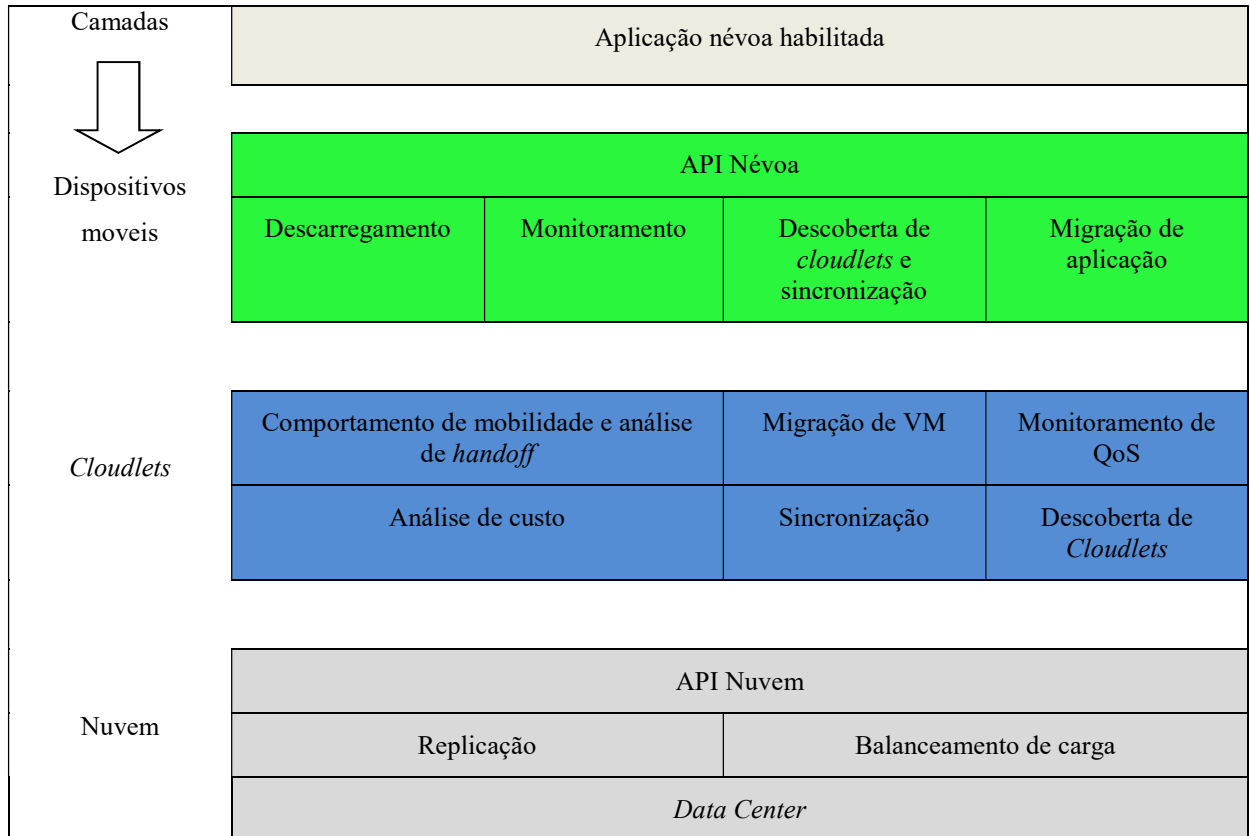
### 3.6 Arquitetura em camadas

A arquitetura em camadas de computação em nuvem, computação em névoa, IoT e usuários, demonstrado na Tabela 2, envolve a organização desses elementos em diferentes níveis de uma estrutura hierárquica. Na camada inferior, temos a computação em nuvem, que fornece serviços de processamento e armazenamento de dados por meio de *data centers* acessados pela internet. Na camada intermediária, temos a computação em névoa, que traz funções de computação, armazenamento, controle e rede para a borda da rede, diminuindo a latência da rede e descongestionando o tráfego de dados para a nuvem. Na camada superior, temos a IoT e os usuários finais. A IoT é uma rede de dispositivos que permitem a troca e análise de dados e é frequentemente usada para monitorar e controlar dispositivos remotamente. Os usuários finais são aqueles que utilizam os serviços fornecidos pelas camadas superiores.

---

<sup>6</sup> *jitter* se refere à variação com atraso na entrega de dados em uma rede. Isso significa que há uma interrupção na sequência normal de troca de pacotes de dados durante a conexão.

Tabela 2 - Arquitetura em camadas para migração de máquinas virtuais



Fonte: Adaptado de Bittencourt *et al.*, (2015).

Bittencourt *et al.*, (2015) descreve a arquitetura de um aplicativo habilitado para névoa que roda em dispositivos móveis e se comunica com camadas inferiores, como *cloudlets* e a nuvem. Os componentes dessa arquitetura são descritos em termos gerais e de alto nível. A camada do dispositivo móvel inclui a API *Fog*, que consiste em várias funcionalidades:

- a) Descarregamento: determina onde os dados devem ser organizados, armazenados e transferidos.
- b) Monitoramento: rastreia as ações dos componentes relacionados às migrações.
- c) Descoberta de *cloudlet* e sincronização: seleciona *cloudlets* com base em parâmetros estabelecidos e oferece funcionalidade de sincronização para replicação, criação e/ou exclusão de dados durante a migração.
- d) Migração de aplicativos: move dados de aplicativos de máquinas virtuais para o *cloudlet* selecionado durante a fase de descoberta do *cloudlet*.

A API da névoa permite o deslocamento de dados e computação e o controle de migração para o desenvolvedor de aplicativos. A camada de *cloudlet* oferece controle sobre os dados do usuário, ou seja, a máquina virtual (ou contêiner) que hospeda os dados. As funcionalidades oferecidas pela camada de *cloudlet* são:

a) Comportamento de mobilidade e análise de *handoff*, que é responsável pela detecção, monitoramento e análise do deslocamento dos usuários;

b) Migração de Máquinas Virtuais, que é o principal módulo que forma a central de migração e é responsável pelas decisões de mobilidade entre as *cloudlets* aptas a receber as máquinas virtuais;

c) Monitoramento de QoS, que é responsável por verificar se a comunicação está ou não dentro dos parâmetros estabelecidos para aplicações em tempo real;

d) Análise de custo, que é baseada em um modelo de negócios com características para o oferecimento de infraestrutura de computação em névoa e de *cloudlets* por parte de empresas privadas;

e) Sincronização, que tem um comportamento similar ao módulo sincronização na camada dos dispositivos móveis;

f) Descoberta de *cloudlet*, que é responsável pela montagem da topologia em que as *cloudlets* estão organizadas para tornar mais eficiente o processo de migração.

Para limitar o escopo desta dissertação, foi necessário escolher somente o módulo migração de máquinas virtuais.

### 3.7 Cenário com migração de VM e *handoff*

Na prática, durante a locomoção dos usuários, podem ocorrer vários cenários em um ambiente com migração de máquinas virtuais para usuários móveis em uma névoa e processos de *handoff* de redes sem fios ativos. Na Figura 2, é possível observar um cenário com deslocamento do usuário (linha vermelha), os *handoffs* realizados (círculos amarelos) e a migração de VM realizada (círculos verdes).

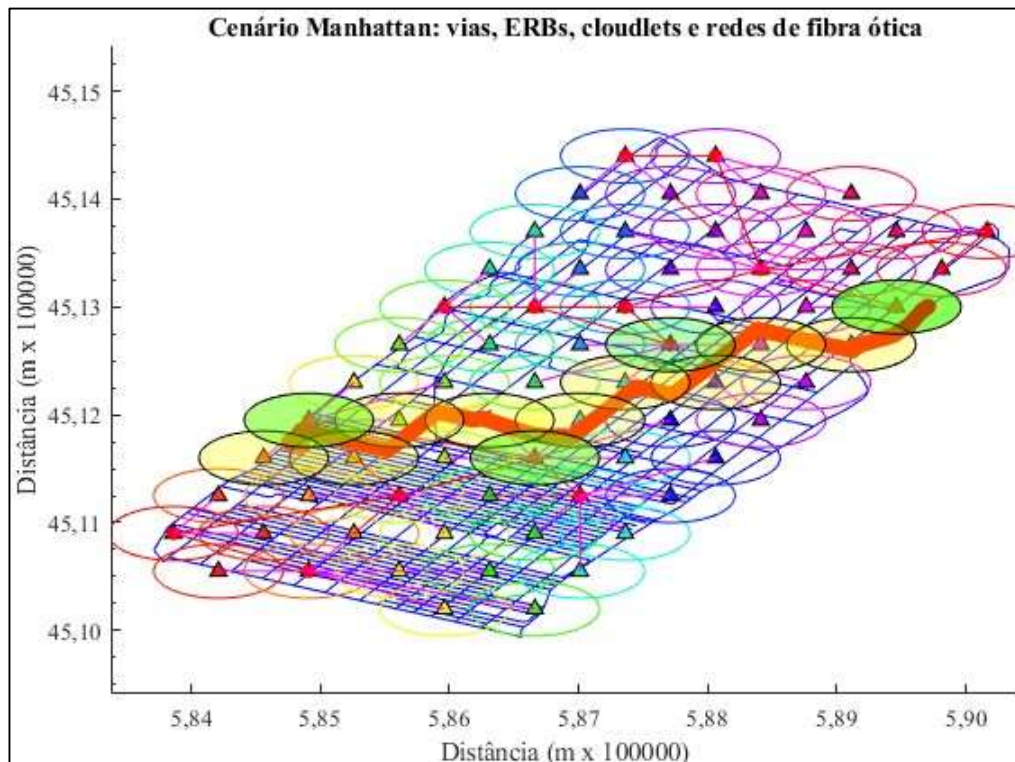


Figura 2 - Deslocamento do usuário, *handoffs* e migração de VMs em computação em névoa. Fonte: Própria

A Figura 3 ilustra um cenário simples que pode ocorrer durante a movimentação de um usuário. Quando ele está prestes a realizar o *handoff*, ou seja, mudando de uma célula de rede para outra, o processo de migração pode ser iniciado para aproveitar o *downtime*<sup>7</sup> causado pelo *handoff* e minimizar o tempo de inatividade total. A lógica é que, como o *handoff* já causará uma interrupção na aplicação do usuário, o processo de migração pode ser iniciado neste momento para “aproveitar” esse período e minimizar o impacto no usuário. Esta abordagem pode ser eficaz para otimizar o processo de migração e reduzir o tempo de inatividade para o usuário. Em uma linha do tempo, várias possibilidades de configurações podem ocorrer envolvendo:

i) *Cloudlets*: dispositivos que fornecem serviços de computação em névoa para usuários móveis.

ii) *Rede de fibra ótica* que conecta os *cloudlets* dentro de uma região de cobertura de computação em névoa;

<sup>7</sup> “Downtime”, que em português significa “inatividade”, é um termo utilizado para descrever o tempo que uma máquina, equipamento ou sistema fica temporariamente inativo.



iii) Rede de fibra ótica que conecta os pontos de acesso (ERBs) a uma *cloudlets* mais próxima;

iv) Uma conexão sem fio que abrange uma área a partir de um ponto de acesso (ERB) que conecta usuários móveis.

v) Eventos em uma linha do tempo que representam ocorrências que acontecem ao longo do tempo, como a migração de máquinas virtuais e os processos de transferência de redes sem fio.

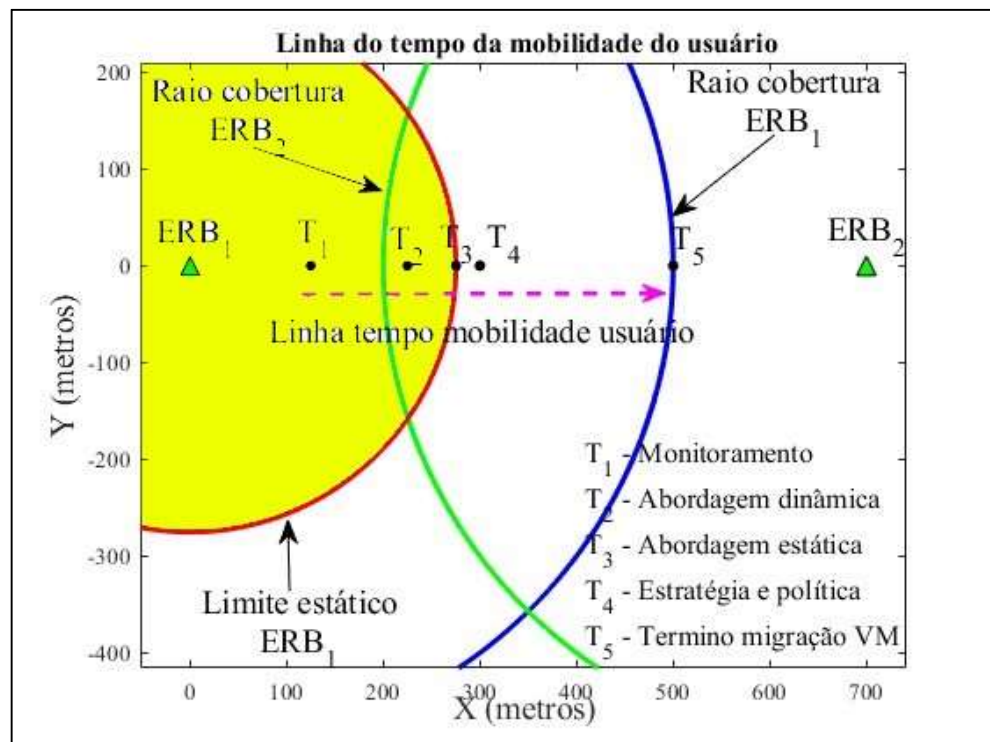


Figura 3 – Linha do tempo da mobilidade do usuário. Fonte: Própria

Antes da migração e do *handoff*, a máquina virtual do usuário é hospedada e operada pela *cloudlet* de origem, que está conectada à ERB do usuário por meio de uma rede de fibra ótica. Após a migração e o *handoff*, a *cloudlet* de destino assume a hospedagem e operação da máquina virtual do usuário. Idealmente, assim que o usuário atravessa uma fronteira geográfica para uma nova região, a máquina virtual já deve estar pronta no destino para atender às novas solicitações do usuário.

Na linha do tempo da mobilidade, existem vários instantes em que ocorrem eventos relacionados a esta arquitetura. Esses instantes são os  $T_x$ <sup>8</sup> e serão descritos a seguir:

- $T_1$  - Monitoramento da mobilidade: É o intervalo de tempo em que o algoritmo monitora a posição, a direção e velocidade dos usuários, além de outros parâmetros pré-estabelecidos;
- $T_2$  - Seleção da migração dinâmica: É o instante em que o algoritmo de migração decide migrar de acordo com as métricas velocidade e direção, por exemplo, ou outra métrica estabelecida no mecanismo;
- $T_3$  - Seleção da migração estática: É o momento em que o algoritmo de migração toma a decisão de migrar com base na métrica de limite máximo de cobertura do sinal de acesso sem fio ou outra métrica estabelecida pelo mecanismo;
- $T_4$  - Seleção da estratégia de migração: É o instante em que o algoritmo de migração decide migrar de acordo com a estratégia estabelecida no mecanismo, por exemplo, menor distância entre o dispositivo do usuário e o ponto de acesso, menor distância entre o dispositivo do usuário e a *cloudlet* ou menor latência entre os *cloudlets*;
- $T_4$  - Seleção da política de migração: É o instante em que o algoritmo de migração decide migrar a VM de acordo com a política estabelecida no mecanismo;
- $T_5$  - Fim do processo de migração: Neste momento, o processo de migração da máquina virtual estará completo. A máquina virtual deverá estar pronta para receber as novas requisições do usuário.

Várias permutações podem ocorrer entre os  $T_x$  quando se leva em consideração aspectos reais. No pior cenário, atrasos em ambos os processos (migração e *handoff*) podem ocorrer e o início e o fim do processo de *handoff* acontecem antes do início do processo de migração. Isso pode ocasionar a degradação na QoE do usuário. No melhor cenário, o

---

<sup>8</sup> Instantes de mobilidade do usuário em horas.

mecanismo de *handoff* de rede e o processo de migração iniciam e terminam juntos. Também pode haver congestionamento na rede entre *cloudlets* ou atraso no início do processo de migração. Diferentes permutações podem ocorrer entre os finais do *handoff* e da migração na linha do tempo.

### 3.8 Abordagens, estratégias e políticas de migração de máquinas virtuais

A migração de máquinas virtuais é um processo importante em ambientes de computação em nuvem e névoa. Para otimizar esse processo, é importante considerar o ponto de migração, as estratégias de migração e as políticas de migração. O ponto de migração refere-se ao momento (quando) em que o processo de migração é iniciado. As estratégias de migração determinam para onde os dados são migrados. As políticas de migração definem as regras e diretrizes para o processo de migração (como). Neste tópico iremos considerar esses fatores, para realizar o processo de migração e minimizar o impacto da mobilidade do usuário.

#### 3.8.1 Abordagem do ponto de migração estática

O ponto de migração estática é um conceito importante no contexto de redes sem fio. Ele se refere ao momento em que um usuário cruza o limite de uma rede sem fio e precisa iniciar os procedimentos para migrar para outra rede. Em nosso estudo, propomos um cenário, ilustrado na Figura 3, no qual a migração é iniciada quando o usuário atinge uma distância pré-definida do ponto de acesso.

A Figura 4 mostra um diagrama representando um cenário de abordagem do ponto de migração estática. O diagrama é um gráfico 2D com os eixos X e Y (em metros) representando longitude e latitude, respectivamente. No gráfico, o círculo verde rotulado como limite máximo de cobertura da ERB<sub>2</sub>, o azul rotulado como limite máximo de cobertura da ERB<sub>1</sub> e um vermelho tracejado rotulado como ponto de abordagem estática da ERB<sub>1</sub>. O usuário se desloca em direção à ERB<sub>2</sub> e atingir 75% do limite máximo de cobertura, ponto vermelho, o algoritmo

decide realizar o *handoff* para a ERB mais próxima e inicia o processo de escolha da *cloudlet* para migração da VM.

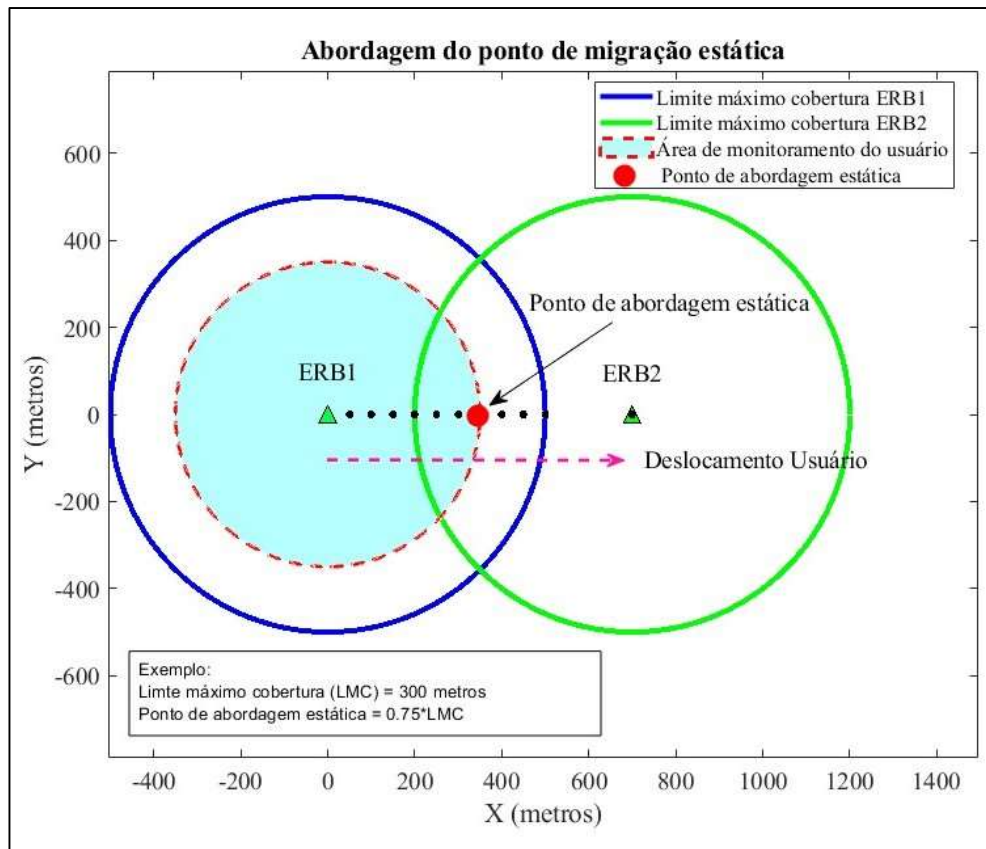


Figura 4 - Cenário de abordagem do ponto de migração estática. Fonte: Própria.

É possível mapear todas as ERBs em tempo real e disponibilizar essas informações para os gerenciadores de migrações de VMs nos *cloudlets*, considerando que as ERBs são localizadas por coordenadas de latitude e longitude e que o usuário se desloca no mapa. Também é possível identificar as áreas de intersecção entre ERBs vizinhas, o raio máximo de alcance das ERBs e definir a localização exata do ponto de migração estática. Além disso, é possível monitorar a localização inicial do usuário, sua velocidade, direção e posição futura para comparar com o ponto de migração da VM e iniciar o processo. Os modelos matemáticos necessários para mapear a rede de ERBs e monitorar o deslocamento do usuário são apresentados a seguir.

Para encontrar os pontos de intersecção de dois círculos em uma esfera com centros dados em latitude e longitude, você pode usar as seguintes etapas:

Seja  $r$  o raio da esfera. Deixe o primeiro círculo ter centro na latitude  $\varphi_1$  e longitude  $\lambda_1$ , e ter um raio angular de  $r_1$ . Deixe o segundo círculo ter centro na latitude  $\varphi_2$  e longitude  $\lambda_2$ , e ter um raio angular de  $r_2$ .

1. Converta as coordenadas de latitude e longitude dos centros dos círculos em coordenadas esféricas ( $\theta, \varphi$ ):

$$\theta_1 = 90^\circ - \varphi_1$$

$$\varphi_1 = \lambda_1$$

$$\theta_2 = 90^\circ - \varphi_2$$

$$\varphi_2 = \lambda_2$$

1. Use a lei esférica dos cossenos para encontrar a distância angular  $d$  entre os centros dos dois círculos:

$$\cos(d) = \cos(\theta_1) \cos(\theta_2) + \sin(\theta_1) \sin(\theta_2) \cos(\varphi_1 - \varphi_2) \quad (1)$$

3. Use a lei esférica dos cossenos para encontrar os ângulos  $a$  e  $b$  entre os pontos de interseção e os centros dos círculos:

$$\cos(a) = \frac{\cos(r_1) - \cos(r_2) \cos(d)}{\sin(r_2) \sin(d)} \quad (2)$$

$$\cos(b) = \frac{\cos(r_2) - \cos(r_1) \cos(d)}{\sin(r_1) \sin(d)} \quad (3)$$

4. Use a lei esférica dos senos para encontrar o ângulo  $c$  entre os pontos de interseção:

$$\sin(c) = \sin(a) \sin(b) \sin(d) \quad (4)$$

5. Use a trigonometria esférica para encontrar as coordenadas esféricas ( $\theta_3, \varphi_3$ ) e ( $\theta_4, \varphi_4$ ) dos pontos de interseção.

6. Converta as coordenadas esféricas dos pontos de interseção em coordenadas de latitude e longitude.

A fórmula de Haversine<sup>9</sup> é usada para calcular a distância do grande círculo entre dois pontos em uma esfera, dadas suas longitudes e latitudes. Aqui está a fórmula:

$$d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (5)$$

onde:

$d$  é a distância entre os dois pontos

$r$  é o raio da esfera

$\varphi_1, \lambda_1$  são a latitude e a longitude do ponto 1 em radianos

$\varphi_2, \lambda_2$  são a latitude e a longitude do ponto 2 em radianos

Aqui está um modelo genérico para encontrar os ângulos entre os centros dos círculos e os pontos de interseção de dois círculos em uma esfera com centros dados em graus decimais de latitude e longitude:

Seja  $r$  o raio da esfera. Deixe o primeiro círculo ter centro na latitude  $\varphi_1$  e longitude  $\lambda_1$ , e ter um raio angular de  $r_1$ . Deixe o segundo círculo ter centro na latitude  $\varphi_2$  e longitude  $\lambda_2$ , e ter um raio angular de  $r_2$ .

1. Converta as coordenadas de latitude e longitude dos centros dos círculos em coordenadas esféricas  $(\theta, \phi)$ :

$$\theta_1 = 90^\circ - \varphi_1$$

$$\phi_1 = \lambda_1$$

$$\theta_2 = 90^\circ - \varphi_2$$

$$\phi_2 = \lambda_2$$

---

<sup>9</sup> O nome haversine foi criado em 1835 pelo matemático e astrônomo James Inman. Estes nomes se devem ao fato de que são escritos nos termos da função haversine, dado por  $\text{hav}(\theta) = \sin^2(\theta/2)$ .

2. Use a lei esférica dos cossenos para encontrar a distância angular **d** entre os centros dos dois círculos:

$$\cos(d) = \cos(\theta_1) \cos(\theta_2) + \sin(\theta_1) \sin(\theta_2) \cos(\varphi_2 - \varphi_1) \quad (6)$$

3. Use a lei esférica dos cossenos para encontrar os ângulos **a** e **b** entre os pontos de interseção e os centros dos círculos:

$$\cos(a) = \frac{\cos(r_1) - \cos(r_2) \cos(d)}{\sin(r_2) \sin(d)} \quad (7)$$

$$\cos(b) = \frac{\cos(r_2) - \cos(r_1) \cos(d)}{\sin(r_1) \sin(d)} \quad (8)$$

Estes são os ângulos entre os pontos de interseção e os centros dos círculos.

Este é um modelo genérico para encontrar os ângulos entre os centros dos círculos e os pontos de interseção de dois círculos em uma esfera com centros dados em graus decimais de latitude e longitude.

### 3.8.2 Abordagem do ponto de migração dinâmica

O ponto de migração dinâmica ocorre quando o usuário atende a um conjunto de requisitos pré-estabelecidos, como velocidade e direção do usuário, tamanho da VM, carga do dispositivo, qualidade do sinal, capacidade da rede, entre outros. Após atender a esses requisitos, os procedimentos para migração devem ser iniciados. No cenário proposto na Figura 5, a migração começará quando o usuário atingir uma velocidade e direção pré-definidas, suficientes para realizar todo o processo da migração da VM antes de atingir o limite máximo de cobertura da ERB atual. Nesta abordagem se faz necessário comparar o tempo de migração da VM de acordo com a política adotada com o tempo para atingir o limite máximo de cobertura e iniciar o processo de migração no limiar deste tempo.

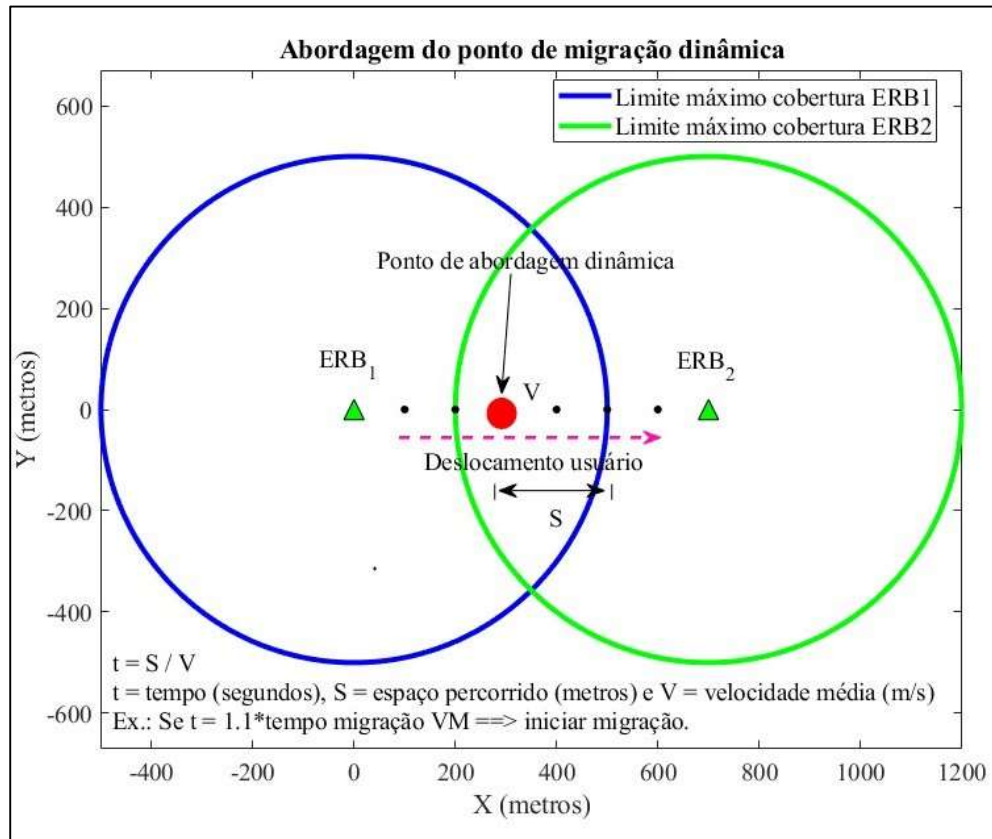


Figura 5 - Cenário de abordagem do ponto de migração dinâmica. Fonte: Própria

Aqui está um modelo matemático simples para calcular a velocidade de um usuário entre dois pontos com latitude e longitude:

Defina o problema: Calcule a velocidade de um usuário entre dois pontos com latitude e longitude.

Defina as variáveis: Digamos que os dois pontos sejam A e B, com coordenadas (latA, lonA) e (latB, lonB), respectivamente. Seja t o tempo que o usuário leva para ir do ponto A ao ponto B.

Calcule uma solução: Primeiro, precisamos calcular a distância entre os dois pontos. Podemos usar a fórmula de Haversine para calcular a distância do grande círculo entre dois pontos em uma esfera a partir de suas longitudes e latitudes. A fórmula é a seguinte:

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{latA}) * \cos(\text{latB}) * \sin^2\left(\frac{\Delta\text{lon}}{2}\right) \quad (9)$$

$$c = 2 * \text{atan2}(\text{sqrt}(a), \text{sqrt}(1 - a)) \quad (10)$$

$$d = R * c \quad (11)$$

onde:

$$\Delta\text{lat} = \text{latB} - \text{latA}$$



$$\Delta \text{lon} = \text{lonB} - \text{lonA}$$

R é o raio da Terra (raio médio = 6.371 km)

d é a distância entre os dois pontos

Uma vez calculada a distância, podemos usar a fórmula velocidade = distância / tempo para calcular a velocidade do usuário entre os dois pontos.

$$t = \frac{S}{V} \text{ (segundos)}$$

Onde:

t = tempo (segundos)

S = espaço percorrido (metros)

V = velocidade média (metros/segundos)

### 3.8.3 Estratégia menor latência

A estratégia de migração de máquina virtual envolve a tomada de decisão sobre o destino de migração da máquina virtual. Essa decisão pode ser baseada em vários fatores, como a latência entre o *cloudlet* de origem e os *cloudlets* de destino, a distância entre o usuário (dispositivo) e os *cloudlets* de destino, ou a proximidade do usuário ao ponto de acesso de destino. Assim, a máquina virtual pode ser migrada para o *cloudlet* com menor latência em relação ao *cloudlet* de origem, para o *cloudlet* mais próximo do usuário (dispositivo), ou para o *cloudlet* cujo ponto de acesso esteja mais próximo do usuário. A Figura 6 apresenta um cenário que ilustra essa estratégia de migração, mostrando como a escolha do destino de migração é feita com base na menor latência entre o *cloudlet* de origem e os *cloudlets* de destino. Neste

exemplo, a latência entre o *cloudlet*<sub>1</sub> e *cloudlet*<sub>2</sub> é menor que a latência entre o *cloudlet*<sub>1</sub> e *cloudlet*<sub>3</sub>, portanto inicia-se o processo de migração para *cloudlet*<sub>1</sub>.

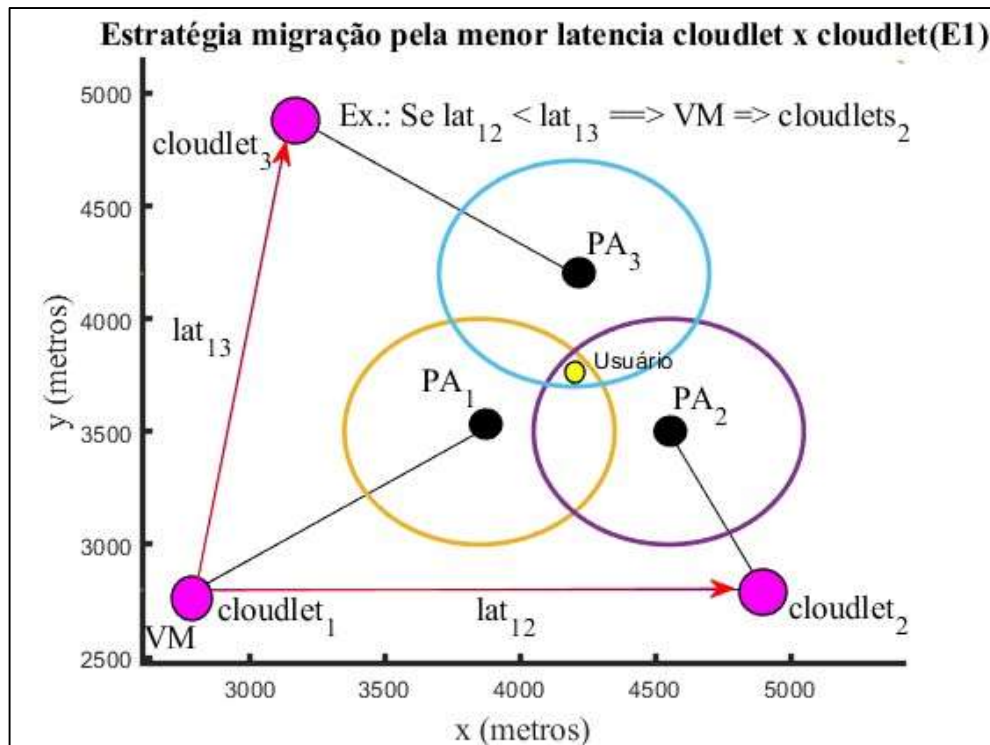


Figura 6 - Cenário estratégias de migrações E1. Fonte: Própria.

A estratégia de migração de máquina virtual que visa minimizar a latência entre *cloudlets* busca garantir que a máquina virtual seja migrada para um *cloudlet* que esteja o mais próximo possível e conectado à mesma rede com a menor latência possível. Isso é feito para minimizar o tempo de transmissão de dados entre os dois *cloudlets* e garantir uma comunicação eficiente da máquina virtual. Na prática, a latência entre dois *cloudlets* pode ser obtida através de testes de *ping*<sup>10</sup> ou *traceroute*<sup>11</sup>, que medem o tempo que os pacotes de dados levam para viajar entre os dois pontos. Essas informações podem ser usadas para determinar o *cloudlet* com

<sup>10</sup> *Ping* é uma ferramenta que envia pacotes de dados do tipo “*echo request*” para um destino específico e mede o tempo que leva para os pacotes serem enviados e recebidos de volta. Isso permite medir a latência, ou seja, o tempo de ida e volta dos pacotes, entre o ponto de origem e o destino.

<sup>11</sup> *Traceroute* é uma ferramenta que rastreia a rota que os pacotes de dados seguem ao serem enviados de um ponto de origem para um destino. Ele envia pacotes com valores crescentes de TTL (Time To Live) e registra os endereços IP dos roteadores pelos quais os pacotes passam. Isso permite visualizar a rota completa que os pacotes seguem e identificar possíveis pontos de congestionamento ou falhas na rede.

a menor latência em relação ao *cloudlet* de origem e, assim, escolher o destino ideal para a migração da máquina virtual.

### 3.8.4 Estratégia menor distância entre usuário e um dos *cloudlets* de destino

Esta estratégia de migração de máquina virtual busca migrar a máquina virtual para um *cloudlet* que esteja o mais próximo possível do usuário. Isso visa garantir que o usuário possa acessar a máquina virtual em um curto espaço de tempo e também minimizar a latência da comunicação entre a máquina virtual e o usuário. Na Figura 7, é apresentado um cenário com estratégia de menor distância entre o usuário e um dos *cloudlets* de destino. É possível observar que o *handoff* deve ser realizado para um ponto de acesso conectado ao *cloudlet* selecionado.

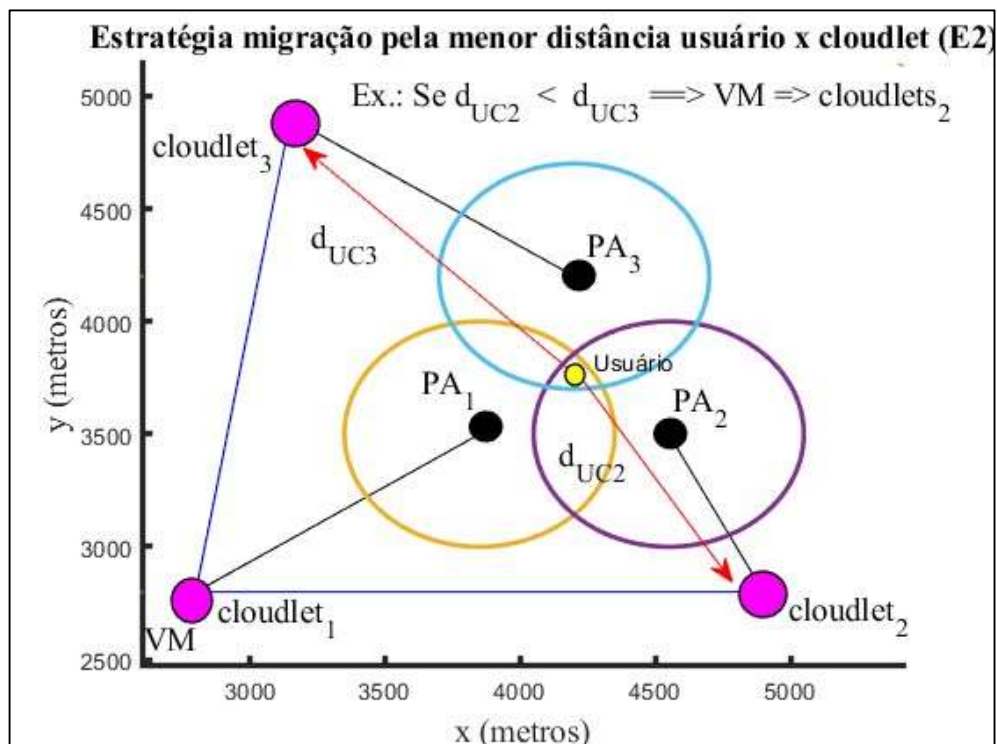


Figura 7 - Cenário estratégias de migrações E2. Fonte: Própria

Aqui está um modelo matemático simples para encontrar a menor distância entre um ponto comum e dois pontos distintos:

Defina o problema: encontre a menor distância entre um ponto comum e dois pontos distintos.

Defina as variáveis: digamos que o ponto comum é C, e os dois pontos distintos são A e B, com coordenadas (latC, lonC), (latA, lonA) e (latB, lonB), respectivamente.

Calcule uma solução: Primeiro, precisamos calcular a distância entre o ponto comum e cada um dos dois pontos distintos. Podemos usar a fórmula de Haversine para calcular a distância do grande círculo entre dois pontos em uma esfera a partir de suas longitudes e latitudes. A fórmula é a seguinte:

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat1}) * \cos(\text{lat2}) * \sin^2\left(\frac{\Delta\text{lon}}{2}\right) \quad (12)$$

$$c = 2 * \text{atan2}(\text{sqrt}(a), \text{sqrt}(1 - a)) \quad (13)$$

$$d = R \cdot c \quad (14)$$

onde:

$$\Delta\text{lat} = \text{lat2} - \text{lat1}$$

$$\Delta\text{lon} = \text{lon2} - \text{lon1}$$

R é o raio da Terra (raio médio = 6.371 km)

d é a distância entre os dois pontos

Podemos usar esta fórmula para calcular a distância entre o ponto C e o ponto A (dCA) e entre o ponto C e o ponto B (dCB). Então, podemos comparar essas duas distâncias para descobrir qual é a mais curta.

### 3.8.5 Estratégia *cloudlet* conectado ao ponto de acesso mais próximo do usuário no momento da abordagem.

*Cloudlet* conectada ao ponto de acesso (PA) mais próximo do usuário no momento da abordagem: Esta estratégia de migração de máquina virtual busca migrar a máquina virtual para um *cloudlet* que esteja conectado ao ponto de acesso do usuário mais próximo do usuário no momento da abordagem. Isso visa garantir que o usuário possa acessar a máquina virtual de forma eficiente, minimizando a latência da comunicação entre a máquina virtual e o ponto de acesso. Na Figura 8 abaixo, é apresentado um cenário com estratégia menor distância entre usuário e ponto de acesso de destino.

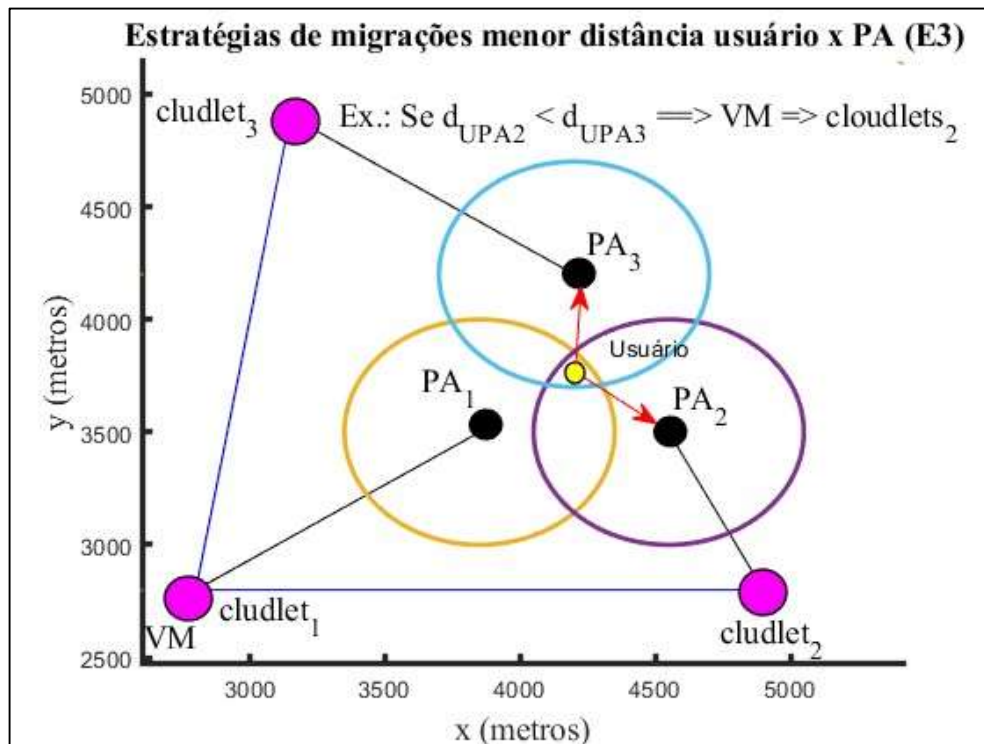


Figura 8 - Cenário estratégias de migrações E3. Fonte: Própria.

### 3.8.6 Política de migração complete container

A política de migração se refere ao modo (técnica) como as páginas de memória máquina virtual deve ser transferida para o *cludlet* de destino. Neste trabalho iremos aplicar as políticas de migração pré-cópia e pós-cópia.

A migração de máquinas virtuais (VM) está se tornando cada vez mais importante na era de computação em névoa. A migração de VM permite a transferência de uma máquina virtual para outro local físico, permitindo que as organizações aproveitem as vantagens de usar ambientes computacionais mais avançados, como computação em nuvem, computação distribuída e computação de alto desempenho. Uma política de migração de VM é essencial para garantir que os ambientes computacionais sejam seguros, eficientes e escalonáveis. Na prática as máquinas virtuais podem ser usadas para hospedar servidores de aplicações, desenvolver e testar aplicativos, hospedar sites, fornecer infraestrutura de computação em nuvem e permitir a virtualização de *desktops*.

Em um ambiente de computação em névoa, a migração de máquinas virtuais permite que os usuários mantenham seus aplicativos em execução em tempo real enquanto se movem

entre áreas de cobertura. Isso é possível transferindo a máquina virtual para uma *cloudlet* dentro da nova área de cobertura.

A técnica de migração pré-cópia tem como objetivo reduzir o tempo de inatividade da máquina virtual, copiando e transferindo todas as páginas da máquina para o destino em várias iterações até que o número de páginas modificadas seja muito pequeno.

A migração pré-cópia permite o reenvio da máquina virtual da origem nos casos de ocorrências de falha severas no destino e, essa capacidade de recuperação torna essa técnica robusta em termos de confiabilidade, sendo, de acordo com Zhang et al. (2018), a preferida pelos provedores de serviços de migração de máquinas virtuais.

Na computação em névoa, o conceito de página de memória é fundamental para a gestão eficiente da memória. Uma página de memória é uma sequência contígua de endereços de memória. Em um sistema operacional que utiliza a paginação, a memória física é dividida em blocos de tamanho fixo chamados páginas. Quando um programa é executado, suas diferentes partes (ou páginas) são carregadas na memória conforme necessário, o que permite que o programa use mais memória do que a disponível fisicamente. Por exemplo, uma máquina virtual (VM) com 1280 MB de dados pode ser dividida em 32 páginas de 40 MB cada, permitindo que a VM acesse apenas as páginas de que precisa, em vez de ter que acessar todos os 1280 MB de dados de uma vez.

Por outro lado, uma página modificada (ou página suja) refere-se a uma página de memória que foi alterada durante a cópia em função da execução de um programa em uso pelo usuário. Isso significa que a página é “suja” e precisa ser copiada para a memória física antes que possa ser usada novamente. Por exemplo, se um programa altera uma página de memória de 4 MB em uma VM, essa página se torna uma “página suja”. Antes que essa página possa ser usada novamente (por exemplo, antes que a VM possa ser migrada), a página suja deve ser copiada de volta para a memória física. Isso garante que todas as alterações feitas na página sejam preservadas e enviadas para a *cloudlet* de destino.

A Figura 9 representa a linha do tempo da técnica de migração pré-cópia pode ser implementada seguindo os seguintes passos:

1. A máquina virtual mantém sua execução na origem.
2. Todas as páginas da máquina são copiadas e transferidas para o destino.

3. Se houver alguma alteração nas páginas recém-copiadas durante a primeira interação, elas serão copiadas na interação seguinte e enviadas para o destino.
4. Esse processo se repete até que o número de páginas modificadas na interação anterior seja muito pequeno.
5. Depois de determinado número de interação, a máquina é interrompida e todas as páginas restantes são enviadas para a VM de destino.

Alterações ocorridas durante o momento de inatividade serão enviadas também neste período.

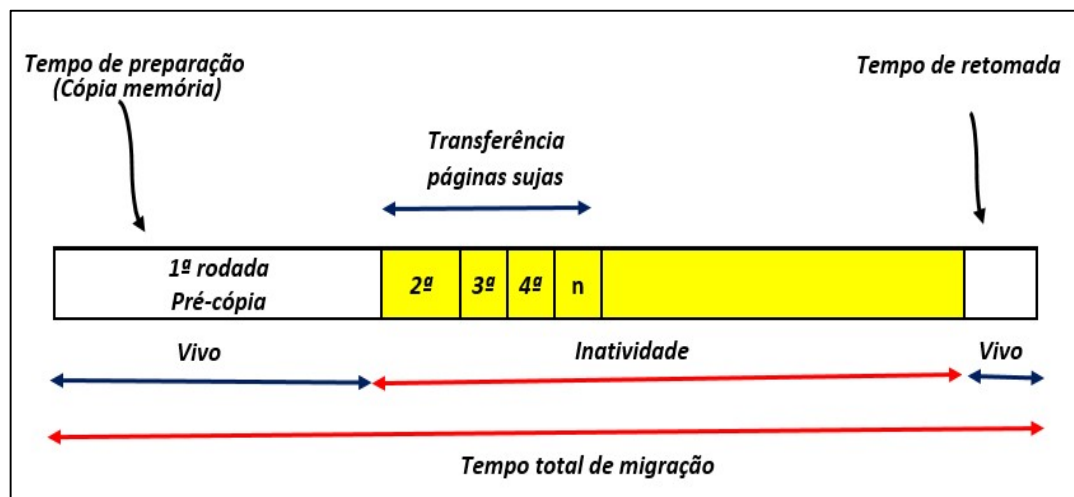


Figura 9 - Linha do tempo da migração pré-cópia. Fonte: Adaptado de Clark (2020).

A migração de uma máquina virtual (VM) é um processo que envolve várias etapas e pode ser dividido em várias rodadas e um exemplo prático é descrito a seguir:

- Na primeira rodada, a VM, que possui 32 páginas de 40 MB cada, é transferida. Isso significa que um total de 1,28 GB de dados (32 páginas x 40 MB por página) é movido de um servidor para outro.
- Nas rodadas subsequentes, menos dados são transferidos. Especificamente, há três rodadas subsequentes nas quais são transferidas 4, 3 e 2 páginas, respectivamente. Cada página tem 4 MB, portanto, um total de 36 MB (9 páginas x 4 MB por página) é transferido nessas rodadas.

- Durante a transferência de dados, a VM fica indisponível, o que é conhecido como tempo de inatividade. Isso pode afetar a experiência do usuário, pois os serviços fornecidos pela VM não estão disponíveis durante esse período.
- Após a transferência de todas as páginas, a VM é retomada. Isso é conhecido como retomada e ocorre após o tempo de inatividade. Durante a retomada, a VM volta a ficar disponível e os serviços que ela fornece podem ser acessados novamente pelos usuários.

O tempo total de migração é a soma do tempo gasto para copiar as páginas, o tempo de inatividade e o tempo de retomada. Este é um indicador importante da eficiência do processo de migração e pode variar dependendo de vários fatores, incluindo a quantidade de dados a serem transferidos, a velocidade da rede e a capacidade do sistema.

### 3.8.7 Política de migração pós-cópia

A migração de pós-cópia é um processo em que uma máquina virtual é pausada antes de iniciar a migração. O estado da CPU, I/O e memória mínima são copiados e enviados para a máquina virtual de destino. Quando a máquina virtual de destino é inicializada, a memória da origem é solicitada usando uma abordagem por demanda. As páginas são enviadas proativamente para garantir que sejam transferidas apenas uma vez e o tempo de migração seja determinístico. Embora essa política tenha a vantagem de ter um tempo de migração previsível, também tem uma desvantagem significativa: se a migração falhar, a máquina virtual é perdida.

A Figura 10 abaixo, representa a linha do tempo da técnica de migração pós-cópia é um processo que envolve várias etapas e pode ser implementada seguindo os seguintes passos:

1. Pausar a máquina virtual: Antes de iniciar o processo de migração, a máquina virtual é pausada para garantir que nenhum dado seja alterado durante o processo.
2. Copiar o estado da CPU, I/O e memória mínima: O estado da CPU, I/O e memória mínima são copiados e enviados para a máquina virtual de destino.
3. Inicializar a máquina virtual de destino: Quando a máquina virtual de destino é inicializada, a memória da origem é solicitada usando uma abordagem por demanda.
4. Transferir páginas de memória: As páginas de memória são enviadas proativamente para garantir que sejam transferidas apenas uma vez e o tempo de migração seja determinístico.



5. Retomar a execução da máquina virtual: Depois que todas as páginas de memória são transferidas, a máquina virtual é retomada na máquina de destino.

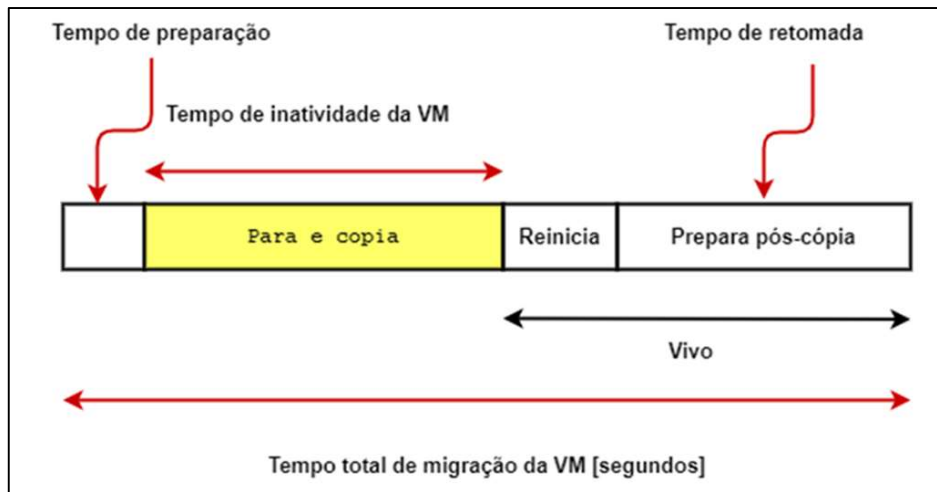


Figura 10 - Linha do tempo da migração pós-cópia. Fonte: Adaptado de Clark (2020).

Neste exemplo, a VM tem 32 páginas de memória, cada uma com 40 MB. Isso significa que a VM tem um total de 1280 MB de memória (32 páginas x 40 MB por página).

O processo de migração começa com a transferência de páginas. Neste caso, 20% das páginas são transferidas primeiro. Isso corresponde a 7 páginas, ou 280 MB de memória (7 páginas x 40 MB por página). Durante essa fase, a VM fica indisponível, o que é conhecido como tempo de inatividade.

Após a transferência inicial, o restante das páginas é transferido. Isso é conhecido como pós-cópia. Neste exemplo, 80% das páginas restantes são transferidas, o que corresponde a 26 páginas, ou 1040 MB de memória (26 páginas x 40 MB por página).

Portanto, o processo de migração envolve a transferência de um total de 32 páginas de memória, ou 1280 MB de dados. Isso é feito em duas fases: a transferência inicial de 20% das páginas durante o tempo de inatividade, seguida pela pós-cópia das 80% páginas restantes. A migração é concluída quando todas as páginas foram transferidas e a VM é retomada.

A retomada da execução da máquina virtual se refere ao momento em que a máquina virtual de origem é desligada e a máquina virtual de destino assume o controle da aplicação. Isso só acontece depois que todas as páginas de memória são transferidas, pois caso contrário, haveria inconsistência entre os dados da origem e do destino. A inicialização da máquina virtual de destino não significa que ela já esteja executando a aplicação, mas apenas que ela está pronta

para receber as páginas de memória que faltam. Portanto, a retomada da execução da máquina virtual é o último passo do processo de migração, que garante a continuidade da aplicação sem interrupções perceptíveis para o usuário.

É importante notar que, embora essa política tenha a vantagem de ter um tempo de migração previsível, também tem uma desvantagem significativa: se a migração falhar, a máquina virtual é perdida.

De acordo com He (2021) e Fraser et al. (2022), há diversas maneiras para migrar o conteúdo da memória de uma VM de um *cloudlet* para outro, no entanto, quando uma VM está executando um serviço ao vivo, é importante que essa transferência ocorra de maneira que equilibre os requisitos de minimizar o tempo de inatividade e o tempo total de migração.

Neste capítulo, apresentamos os conceitos básicos sobre as políticas de migração de máquinas virtuais, que são fundamentais para entender o funcionamento e os desafios dessas técnicas. No próximo capítulo, iremos apresentar a nossa proposta de algoritmo de seleção de abordagem, estratégia e política de migração.

## 4. METODOLOGIA

O objetivo desta pesquisa é desenvolver um algoritmo de seleção de abordagem, estratégia e política de migração de máquinas virtuais em função dos parâmetros de entrada do sistema, tais como, qualidade do sinal, largura de banda, latência e tamanho da VM com propósito de permitir que o usuário em mobilidade acesse aplicações e dados de qualquer lugar e a qualquer momento mantendo a conexão com a internet, afim de minimizar o tempo total de migração da VM, o tempo de inatividade da VM, o total de migrações e o tamanho das VMs migrações. O sistema monitora a localização do usuário na rede de computação em névoa e os parâmetros do sistema e quando apropriado, executa o *handoff* e a migração das Máquinas Virtuais (VMs). A seguir, explora-se alguns dos desafios específicos associados à migração de VMs para usuários em movimento.

### 4.1 Função do algoritmo de migração de VM

O algoritmo descrito a seguir tem como principal função a migração de Máquinas Virtuais (VMs) durante o deslocamento do usuário. O objetivo é minimizar o tempo total de migração e o tempo de inatividade da VM. Para isso, o algoritmo leva em consideração parâmetros como a qualidade do sinal, a latência da rede, a largura de banda e o tamanho da VM.

A redução do tempo total de migração e inatividade é alcançada através da seleção da melhor abordagem, estratégia e política de migração. Essa seleção é baseada nos parâmetros de entrada e nos critérios adotados para a respectiva seleção de abordagem, estratégia e política.

Os critérios de seleção são adotados a partir de estudos acadêmicos citados na bibliografia. Esses estudos relacionam os parâmetros de entrada a uma abordagem, estratégia e política mais adequadas para a migração da VM.

Durante o deslocamento do usuário, o algoritmo decide quando, para onde e como migrar a VM. Essa seleção é tomada com base nos parâmetros de entrada e nos critérios de seleção.

O problema que se pretende resolver com este algoritmo é a redução do tempo total de migração e do tempo de inatividade da VM. Isso é alcançado através da seleção adequada de abordagem, estratégia e política de migração, que são adaptadas às necessidades do usuário. Dessa forma, o algoritmo contribui para uma melhor experiência do usuário, minimizando interrupções e garantindo uma transição suave durante a migração da VM.

#### 4.1.1 Critérios de tratamento dos parâmetros de entrada

Para classificar um sinal como estável ou instável em relação ao limite de cobertura, latência da rede, largura de banda e tamanho da VM, você pode considerar os seguintes critérios:

- Limite de cobertura: Se a posição do usuário estiver dentro do limite de cobertura da ERB (Estação Rádio Base) durante um período de tempo considerável, o sinal pode ser classificado como estável. Se a posição do usuário estiver se movendo para fora do limite de cobertura, o sinal pode ser classificado como instável.
- Latência da rede: Se a latência da rede estiver dentro de um limite aceitável e não variar muito durante um período de tempo, o sinal pode ser classificado como estável. Se a latência da rede estiver variando muito ou exceder um limite aceitável, o sinal pode ser classificado como instável.
- Largura de banda: Se a largura de banda estiver acima de um certo limite e não variar muito durante um período de tempo, o sinal pode ser classificado como estável. Se a largura de banda estiver abaixo desse limite ou variar muito, o sinal pode ser classificado como instável.
- Tamanho da VM: Se o tamanho da VM for pequeno o suficiente para ser acomodado pela largura de banda disponível e a latência da rede, o sinal pode ser classificado como estável. Se o tamanho da VM for muito grande para a largura de banda disponível ou a latência da rede, o sinal pode ser classificado como instável.

Você pode implementar esses critérios em seu algoritmo monitorando continuamente esses parâmetros e classificando o sinal como estável ou instável com base em seus valores.

A função MATLAB abaixo é usada para gerar valores aleatórios para quatro parâmetros diferentes: qualidade do sinal, latência da rede, largura de banda e tamanho da Máquina Virtual (VM).

```
qualidade_do_sinal = rand(1, n);
```

```
latencia_da_rede = rand(1, n);
```

```
largura_de_banda = rand(1, n);
```

```
tamanho_da_VM = rand(1, n);
```

Aqui estão os detalhes:

- `qualidade_do_sinal = rand(1, n);` - Esta linha gera um *array* de *n* números aleatórios entre 0 e 1 para representar a qualidade do sinal.
- `latencia_da_rede = rand(1, n);` - Esta linha gera um *array* de *n* números aleatórios entre 0 e 1 para representar a latência da rede.
- `largura_de_banda = rand(1, n);` - Esta linha gera um *array* de *n* números aleatórios entre 0 e 1 para representar a largura de banda.
- `tamanho_da_VM = rand(1, n);` - Esta linha gera um *array* de *n* números aleatórios entre 0 e 1 para representar o tamanho da Máquina Virtual (VM).

A finalidade desta função é gerar valores aleatórios para esses parâmetros, que podem ser usados para simular diferentes condições de rede e recursos de VM em um ambiente de teste. Isso pode ser útil para testar o desempenho de um sistema sob diferentes condições.

#### 4.1.2 Normalização dos parâmetros de entrada

A normalização dos parâmetros é uma etapa crucial em muitos algoritmos de aprendizado de máquina e otimização. O objetivo da normalização é ajustar os valores medidos em diferentes escalas para uma escala comum.

Nas funções a serem apresentadas, a normalização é usada para garantir que todos os parâmetros (qualidade do sinal, latência da rede, largura de banda e tamanho da VM) contribuam igualmente para a pontuação final, independentemente de suas unidades ou magnitude.

Por exemplo, a latência da rede pode ser medida em milissegundos e pode variar de 0 a 100, enquanto a qualidade do sinal pode ser uma porcentagem variando de 0 a 1. Se esses parâmetros não forem normalizados, a latência da rede, devido à sua maior magnitude, dominará a pontuação final, o que pode levar a uma política de migração sub ótima.

Ao normalizar esses parâmetros para a mesma escala (neste caso, dividindo cada parâmetro pelo seu valor máximo), garantimos que cada parâmetro tenha a mesma chance de influenciar a política de migração, levando a uma seleção mais equilibrada e potencialmente mais otimizada.

#### 4.1.3 Descrição da função *selecionar\_abordagem*

A função *selecionar\_abordagem* opera com base em um conceito conhecido como decisão de limiar (JALALI MOGHADDAM *et al.*, 2020). Nesse contexto, cada parâmetro, descritos na Tabela 2 (qualidade do sinal, latência da rede, largura de banda e tamanho da VM) é comparado a um limite predefinido. Se o parâmetro estiver acima do limite (para qualidade do sinal e largura de banda) ou abaixo do limite (para latência da rede e tamanho da VM), a contagem para a abordagem estática é incrementada. Caso contrário, a contagem para a abordagem dinâmica é incrementada.

Parâmetros de Entrada	Descrição
qualidade_do_sinal	Um vetor que representa a qualidade do sinal
latencia_da_rede	Um vetor que representa a latência da rede
largura_de_banda	Um vetor que representa a largura de banda
tamanho_da_VM	Um vetor que representa o tamanho da Máquina Virtual (VM)

Tabela 2 - descrição dos parâmetros de entrada da função *selecionar\_abordagem*

A função *selecionar\_abordagem* é descrita na Tabela 3 abaixo retorna a abordagem selecionada (Estática ou Dinâmica). Nota-se que a escolha dos limites e a lógica para decidir a abordagem podem variar dependendo do contexto específico e dos objetivos do problema que se pretende resolver.

Operações	Descrição
Inicialização	A variável abordagem é inicializada como uma <i>string</i> vazia
Definição de Limites	Limites são definidos para cada parâmetro. Neste caso, todos os limites são definidos como 0.5
Seleção da Abordagem	A abordagem é decidida com base nos parâmetros. Se a qualidade do sinal for maior que o limite e a latência da rede for menor que o limite, a abordagem será 'Estática'. Caso contrário, a abordagem será 'Dinâmica'

Tabela 3 - Etapas da função *selecionar\_abordagem*

Na Tabela 4 abaixo apresenta-se as condições para seleção da abordagem estática ou dinâmica estão os parâmetros mencionados e suas possíveis interpretações:

**Qualidade do sinal:** Este parâmetro pode se referir à qualidade da conexão de rede. Na abordagem estática, se a qualidade do sinal for maior que um limite definido, a condição é satisfeita. Na abordagem dinâmica, a condição é satisfeita se a qualidade do sinal for menor ou igual ao limite.

**Latência de rede:** A latência de rede é o tempo que leva para os dados serem transferidos de um ponto para outro na rede. As condições para abordagens estática e dinâmica podem variar com base na latência da rede.

**Largura de banda:** A largura de banda é a capacidade máxima de transferência de dados de uma rede. As condições para as abordagens estática e dinâmica podem depender da largura de banda disponível.

**Tamanho da VM:** Este parâmetro pode se referir ao tamanho da Máquina Virtual (VM) em termos de recursos alocados, como CPU, memória, etc. As condições para as abordagens estática e dinâmica podem variar com base no tamanho da VM.

**Tamanho do arquivo:** Este parâmetro pode se referir ao tamanho do arquivo que está sendo transferido ou processado. As condições para as abordagens estática e dinâmica podem variar com base no tamanho do arquivo.

Parâmetro	Condição para Abordagem Estática	Condição para Abordagem Dinâmica
<b>Qualidade do sinal</b>	qualidade_do_sinal > limite qualidade do sinal	qualidade_do_sinal <= limite qualidade do sinal
<b>Latência da rede</b>	latência_da_rede < limite latência da rede	latência_da_rede >= limite latência da rede
<b>Largura de banda</b>	largura_de_banda > limite largura de banda	largura_de_banda <= limite largura de banda
<b>Tamanho da VM</b>	tamanho_da_VM < limite tamanho da VM	tamanho_da_VM >= limite tamanho da VM

Tabela 4 - Condições para abordagem estática e dinâmica

A função *escolher\_abordagem* é apresentada a seguir:

```
% Definir o tamanho dos arrays
n = 1;

% Criar arrays aleatórios para qualidade do sinal, latência da rede, largura de banda e tamanho da VM
qualidade_do_sinal = rand(1, n);
```

```

latencia_da_rede = rand(1, n);
largura_de_banda = rand(1, n);
tamanho_da_VM = rand(1, n);

% Definir os limites para cada parâmetro
limite_qualidade_do_sinal = .5;
limite_latencia_da_rede = .5;
limite_largura_de_banda = .5;
limite_tamanho_da_VM = .5;

% Decidir a abordagem a ser adotada
for i = 1:n
    abordagem = escolher_abordagem(qualidade_do_sinal(i), latencia_da_rede(i), largura_de_banda(i), tamanho_da_VM(i),
limite_qualidade_do_sinal, limite_latencia_da_rede, limite_largura_de_banda, limite_tamanho_da_VM);
    disp(['A abordagem a ser adotada para o usuário ' num2str(i) ' é ' abordagem]);
end

% Função para decidir a abordagem a ser adotada
function abordagem = escolher_abordagem(qualidade_do_sinal, latencia_da_rede, largura_de_banda, tamanho_da_VM,
limite_qualidade_do_sinal, limite_latencia_da_rede, limite_largura_de_banda, limite_tamanho_da_VM)
    % Inicializar contadores para abordagem estática e dinâmica
    estatica = 0;
    dinamica = 0;

    % Verificar cada parâmetro e incrementar o contador apropriado
    if qualidade_do_sinal > limite_qualidade_do_sinal
        estatica = estatica + 1;
    else
        dinamica = dinamica + 1;
    end

    if latencia_da_rede < limite_latencia_da_rede
        estatica = estatica + 1;
    else
        dinamica = dinamica + 1;
    end

    if largura_de_banda > limite_largura_de_banda
        estatica = estatica + 1;
    else
        dinamica = dinamica + 1;
    end

    if tamanho_da_VM < limite_tamanho_da_VM
        estatica = estatica + 1;
    else
        dinamica = dinamica + 1;
    end
end

```



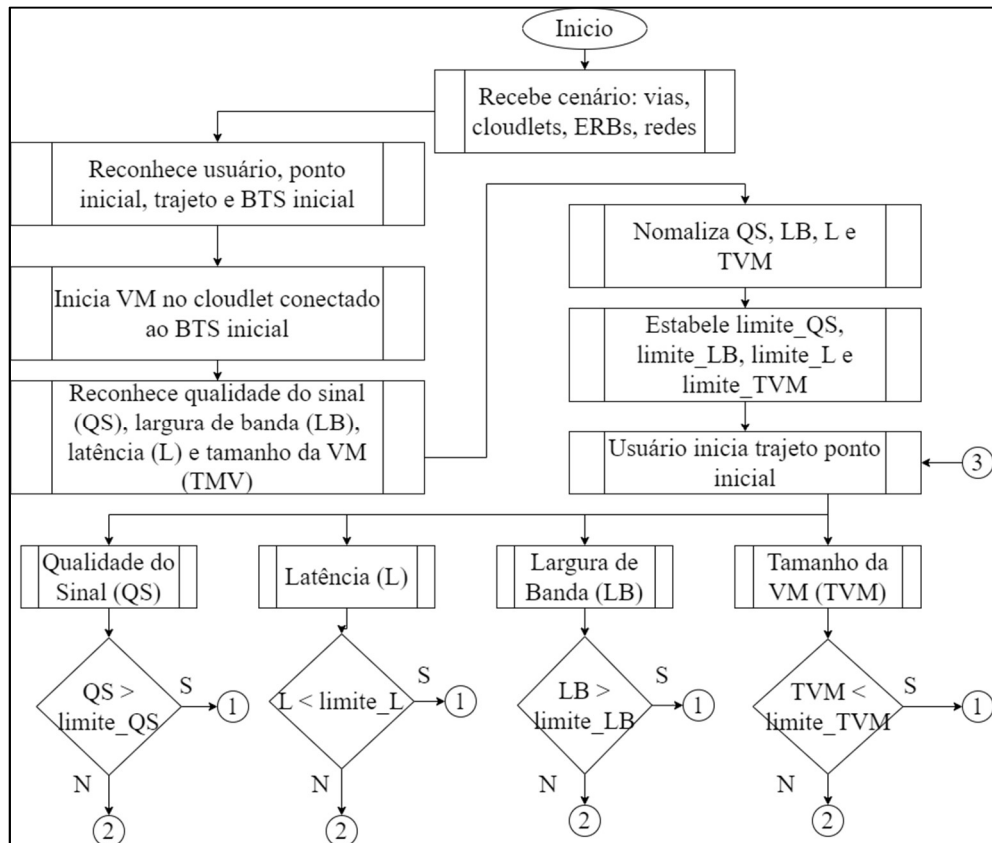
```

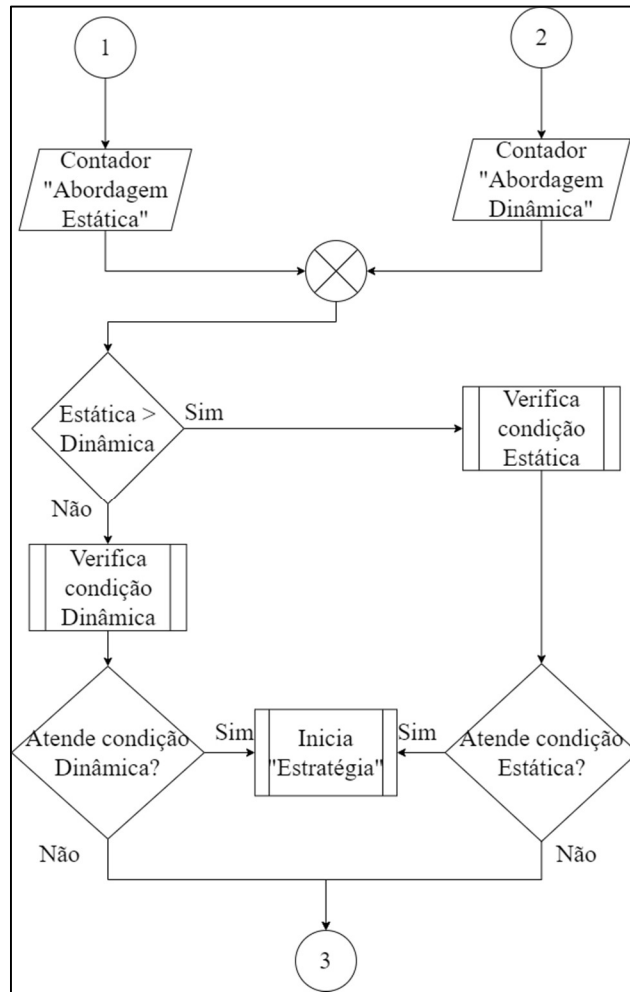
% Decidir a abordagem a ser adotada com base na maioria
if estatica > dinamica
    abordagem = 'estatica';
else
    abordagem = 'dinamica';
end
end
end

```

No fluxograma abaixo apresenta-se o processo de seleção da abordagem de migração de VM a ser adotada pelo algoritmo de seleção. Observa-se que o algoritmo recebe os parâmetros de entrada, compara individualmente com os limites pré-estabelecidos, verifica qual a estratégia que mais se adequa e finalmente retorna a abordagem mais adequada aos parâmetros de entrada.

Figura 11 - Algoritmo de migração de VMs/função\_abordagem [Fluxograma]. Fonte: Própria





#### 4.1.4 Descrição da função *selecionar\_estratégia*

A função *selecionar\_estratégia* opera com base em fatores de relevância do serviço que calcula média ponderada levando em consideração pesos e parâmetros de entrada (CHEN *et al.*, 2021). Na Tabela 5 abaixo é descrito as operações realizadas para retornar a estratégia a ser adotada para seleção do *cloudlet* de destino.

Operações	Descrição
Inicialização	A variável estratégia é inicializada como uma <i>string</i> vazia
Definição de Pesos	Pesos iguais são definidos para todos os parâmetros
Cálculo da Pontuação	A pontuação para três estratégias diferentes (E1, E2, E3) é calculada com base nos parâmetros fornecidos
Seleção da Estratégia	A estratégia com a maior pontuação é selecionada

Tabela 5 - Etapas da função *selecionar\_estratégia*

Na Tabela 6 abaixo apresenta-se as condições para seleção da estratégia E1, E2 e E3 com os parâmetros mencionados e suas possíveis interpretações:

Pontuações	Descrição
pontuacao_E1	É a média ponderada da qualidade do sinal e do inverso da latência da rede
pontuacao_E2	É a média ponderada da largura de banda e do inverso do tamanho da VM
pontuacao_E3	É a média ponderada da qualidade do sinal e da largura de banda

Tabela 6 - Condições para estratégias E1, E2 e E3

A função *escolher\_abordagem* é apresentada a seguir:

```
function estrategia = seleccionar_estrategia(qualidade_do_sinal, latencia_da_rede, largura_de_banda, tamanho_da_VM)
% Inicializar a variável estrategia
estrategia = "";

% Definir os pesos para os diferentes parâmetros
peso_qualidade_do_sinal = 1;
peso_latencia_da_rede = 1;
peso_largura_de_banda = 1;
peso_tamanho_da_VM = 1;

% Calcule a pontuação para cada estratégia com base nos parâmetros fornecidos
pontuacao_E1 = peso_qualidade_do_sinal x mean(qualidade_do_sinal) + peso_latencia_da_rede x (1 - mean(latencia_da_rede));
pontuacao_E2 = peso_largura_de_banda x mean(largura_de_banda) + peso_tamanho_da_VM x (1 - mean(tamanho_da_VM));
pontuacao_E3 = peso_qualidade_do_sinal x mean(qualidade_do_sinal) + peso_largura_de_banda x mean(largura_de_banda);

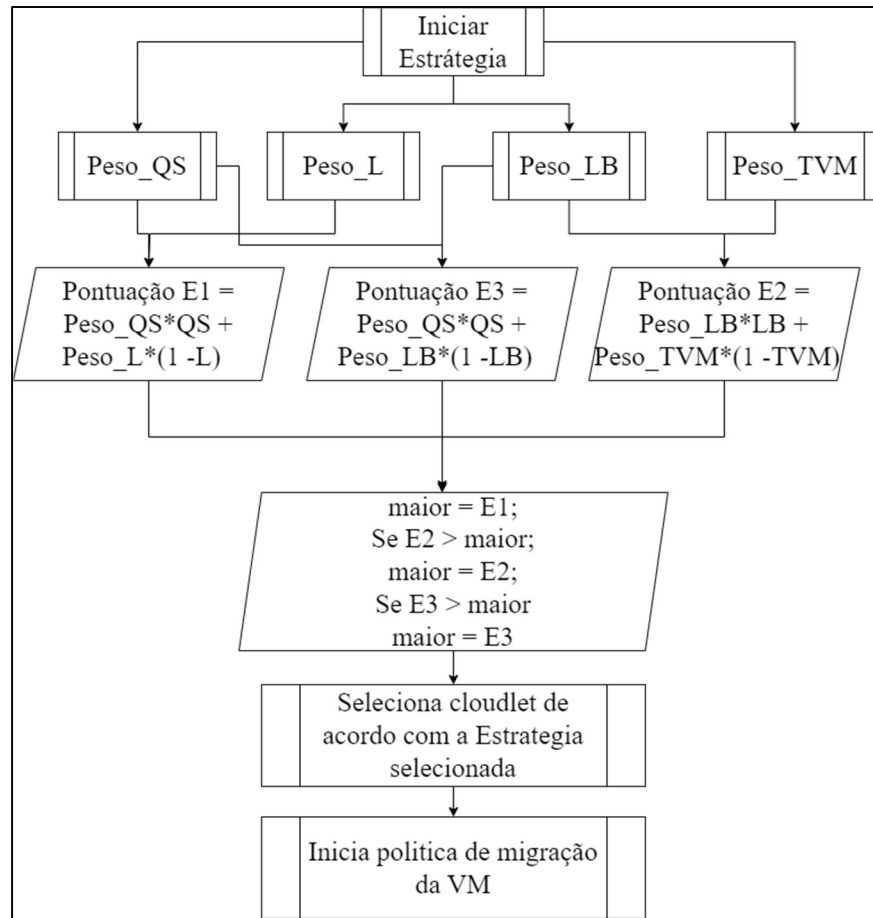
% Selecione a estratégia com a maior pontuação
[~, indice] = max([pontuacao_E1, pontuacao_E2, pontuacao_E3]);

% Atribua a estratégia selecionada à variável estrategia
if indice == 1
    estrategia = 'E1';
elseif indice == 2
    estrategia = 'E2';
else
    estrategia = 'E3';
end

return;
end
```

Nos fluxogramas abaixo apresenta-se o processo de seleção da estratégia de seleção da VM a ser adotada pelo algoritmo de seleção. Observa-se que o algoritmo recebe os parâmetros de entrada, calcula a pontuação para cada estratégia a dos parâmetros de entrada e finalmente seleciona a *cloudlet* de acordo com a estratégia que receber maior pontuação.

Figura 12 - Algoritmo de migração de VMs/função\_estratégia [Fluxograma]. Fonte: Própria



#### 4.1.5 Descrição da função *selecionar\_política*

A função *selecionar\_política* é descrita na Tabela 7 abaixo e retorna a política a ser adotada para realizar a migração da VM

Passo	Descrição
1	Inicialização de variáveis: Define o número de <i>cloudlets</i> como 10 e inicializa <i>arrays</i> aleatórios para qualidade do sinal, latência da rede, largura de banda e tamanho da VM. A variável política é inicializada como uma <i>string</i> vazia.
2	Definição de pesos: Define os pesos para os diferentes parâmetros como 1.
3	Normalização dos parâmetros: Normaliza os parâmetros de entrada para a mesma escala dividindo cada um pelo seu valor máximo. Isso garante que todos os parâmetros contribuam igualmente para a pontuação final, independentemente de suas unidades ou magnitude.
4	Cálculo da pontuação: Calcula a pontuação para cada <i>cloudlet</i> com base nos parâmetros fornecidos. A qualidade do sinal e a largura de banda contribuem positivamente para a pontuação, enquanto a latência da rede e o tamanho da VM contribuem negativamente.

Passo	Descrição
5	Seleção da política de migração: Calcula a média da pontuação e, se for maior que 0, escolhe a política de pré-cópia. Caso contrário, escolhe a política de pós-cópia.

Tabela 7 - Etapas da função selecionar\_política

A pontuação calculada na função *migrar\_VM* é baseada em uma abordagem teórica usada em sistemas de computação em nuvem, conhecida como "VM DRS Score" (SOLOMON; ALLEY, 2014). A ideia básica é calcular uma pontuação para cada VM com base em vários parâmetros que afetam o desempenho da VM. No caso da função *migrar\_VM*, os parâmetros considerados são a qualidade do sinal, a latência da rede, a largura de banda e o tamanho da VM. A pontuação é calculada como a soma ponderada desses parâmetros, onde cada parâmetro é normalizado para a mesma escala (dividindo pelo valor máximo) e então multiplicado por um peso correspondente. A qualidade do sinal e a largura de banda contribuem positivamente para a pontuação (pois valores mais altos desses parâmetros são desejáveis), enquanto a latência da rede e o tamanho da VM contribuem negativamente para a pontuação (pois valores mais baixos desses parâmetros são desejáveis). A pontuação resultante é usada para decidir a política de migração da VM. Se a média da pontuação for maior que 0, a política de pré-cópia é escolhida. Caso contrário, a política de pós-cópia é escolhida.

Resumo o cálculo da pontuação: A pontuação é calculada como a soma ponderada dos parâmetros normalizados. Isso significa que cada parâmetro (qualidade do sinal, latência da rede, largura de banda e tamanho da VM) é primeiro normalizado (ou seja, ajustado para uma escala comum, geralmente 0 a 1), e então esses valores normalizados são combinados para calcular a pontuação:

- Qualidade do sinal e largura de banda: Esses dois parâmetros contribuem positivamente para a pontuação. Isso significa que, quando a qualidade do sinal ou a largura de banda aumentam, a pontuação também aumenta. Isso faz sentido intuitivamente, pois uma maior qualidade do sinal e uma maior largura de banda geralmente levam a um melhor desempenho da VM.
- Latência da rede e tamanho da VM: Por outro lado, esses dois parâmetros contribuem negativamente para a pontuação. Isso significa que, quando a latência da rede ou o tamanho da VM aumentam, a pontuação diminui. Isso também faz sentido intuitivamente, pois uma maior latência da rede pode levar

a um desempenho mais lento da VM, e uma VM maior pode exigir mais recursos, o que pode ser mais difícil de acomodar.

Portanto, a pontuação é uma medida do desempenho esperado da VM, levando em consideração vários parâmetros importantes. Uma pontuação mais alta indica um desempenho esperado melhor, enquanto uma pontuação mais baixa indica um desempenho esperado pior. Esta pontuação pode ser usada para tomar decisões sobre a alocação de recursos ou a migração de VMs.

A função *escolher\_politica*, em matlab, é apresentada a seguir:

```
function [politica] = escolher_politica(qualidade_do_sinal, latencia_da_rede, largura_de_banda, tamanho_da_VM)

n = 10; % Número de cloudlets
% Criar arrays aleatórios para qualidade do sinal, latência da rede, largura de banda e tamanho da VM
qualidade_do_sinal = rand(1, n);
latencia_da_rede = rand(1, n);
largura_de_banda = rand(1, n);
tamanho_da_VM = rand(1, n);

% Inicialize a variável política
politica = "";

% Defina os pesos para os diferentes parâmetros
peso_qualidade_do_sinal = 1;
peso_latencia_da_rede = 1;
peso_largura_de_banda = 1;
peso_tamanho_da_VM = 1;

% Normalize os parâmetros de entrada para a mesma escala
qualidade_do_sinal = qualidade_do_sinal / max(qualidade_do_sinal);
latencia_da_rede = latencia_da_rede / max(latencia_da_rede);
largura_de_banda = largura_de_banda / max(largura_de_banda);
tamanho_da_VM = tamanho_da_VM / max(tamanho_da_VM);

% Calcule a pontuação para cada cloudlet com base nos parâmetros fornecidos
pontuacao = peso_qualidade_do_sinal x qualidade_do_sinal - peso_latencia_da_rede x latencia_da_rede + peso_largura_de_banda x
largura_de_banda - peso_tamanho_da_VM x tamanho_da_VM;

% Calcule a média da pontuação
media_pontuacao = mean(pontuacao);

% Se a média da pontuação for maior que 0, use a política de pré-cópia
if media_pontuacao > 0
    politica = 'pre-copia';
% Caso contrário, use a política de pós-cópia
```

```

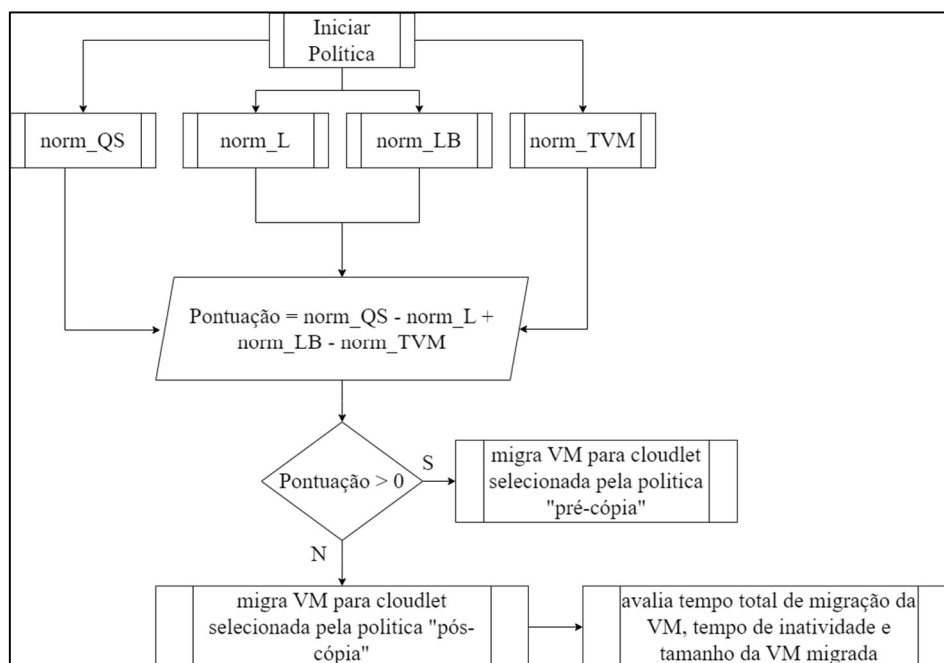
else
  politica = 'pos-copia';
end

return;
end

```

Nos fluxogramas abaixo apresenta-se o processo de seleção da política de migração da VM a ser adotada pelo algoritmo de decisão. Observa-se que o algoritmo recebe os parâmetros de entrada, calcula a pontuação para cada para o *cloudlet* de acordo com a fórmula e seleciona a política de acordo com a média calculada.

Figura 13 - Algoritmo de migração de VMs/função\_política [Fluxograma]. Fonte: Própria



## 4.2 Parâmetros Avaliados

A seguir, explora-se quatro tópicos principais relacionados à migração de Máquinas Virtuais (VMs) em cenários de computação em nuvem. Esses tópicos são fundamentais para entender a eficiência e eficácia de um algoritmo de migração de VMs.

- Tempo total de migração de VM: Este é o tempo total necessário para migrar uma máquina virtual (VM) de um servidor (ou *cloudlet*) para outro. Ele inclui o tempo necessário para preparar a VM para a migração, transferir os dados da VM, e reiniciar a VM no servidor de destino. A minimização do tempo total de migração é

crucial para garantir uma experiência suave para o usuário e evitar interrupções no serviço.

- Tempo de indisponibilidade da VM: Este é o tempo durante o qual a VM não está disponível para o usuário durante a migração. Durante este tempo, o usuário pode experimentar uma interrupção no serviço. A minimização do tempo de indisponibilidade é importante para garantir a continuidade do serviço e a satisfação do usuário.
- Tamanho das VMs migradas: Este é o tamanho total dos dados das VMs que foram migrados. O tamanho dos dados pode afetar tanto o tempo total de migração quanto o tempo de indisponibilidade, pois VMs maiores requerem mais tempo para serem transferidas.

Estes parâmetros são essenciais para avaliar a eficácia do algoritmo de migração e podem ser usados para otimizar o algoritmo e melhorar a qualidade do serviço e a seguir detalha-se como é tratado em cada política.

#### 4.2.1 Estimativa do tempo de migração de VM usando a técnica de pré-cópia

Neste tópico, aborda-se a estimativa do tempo de migração de VM usando a técnica de pré-cópia. A pré-cópia é uma técnica que envia os dados da VM para o servidor de destino antes da migração. Ela consiste em várias rodadas de cópia, onde cada rodada envia as páginas de memória que foram modificadas desde a última rodada. A pré-cópia visa reduzir o tempo de inatividade da VM durante a migração.

Para estimar o tempo de migração usando a pré-cópia, precisamos considerar alguns fatores, como o tamanho dos dados da VM, o número de páginas de memória, o número de rodadas de cópia, o número de páginas modificadas por rodada, a largura de banda da rede e a latência da rede. Além disso, precisamos considerar o tempo para suspender e retomar a VM no servidor de origem e destino, respectivamente.

Para converter 1280 MB (Megabytes) para Mbps (Megabits por segundo), precisa-se entender que 1 Byte é igual a 8 bits. Então, primeiro, converta Megabytes para Megabits.



Depois, para saber quantos Megabits podem ser transferidos por segundo, divide-se pelo tempo em segundos.

Aqui está a fórmula:

$$\text{Megabytes} \times 8 = \text{Megabits}$$

$$\text{Megabits} \div \text{Tempo em segundos} = \text{Megabits por segundo (Mbps)}$$

A seguir, apresentamos as fórmulas para calcular o tempo de migração usando a pré-cópia, bem como um exemplo numérico para ilustrar o cálculo.

Fórmulas:

- Tempo de transferência = Tamanho dos dados / Largura de banda
- Tempo de cópia = Número de rodadas x (Tempo de cópia por rodada + Latência da rede)
- Tempo total de migração = Tempo de cópia + Tempo de transferência + tempo suspender a VM + tempo retomada VM
- Tempo de inatividade = Tempo para suspender a VM + Tempo para retomar a VM.

#### 4.2.2 Estimativa do tempo de migração de VM usando a técnica de pós-cópia

Neste tópico, aborda-se a estimativa do tempo de migração de VM usando a técnica de pós-cópia. A pós-cópia é uma técnica que envia os dados da VM para o servidor de destino após a migração. Ela consiste em suspender a VM no servidor de origem, transferir o estado da VM para o servidor de destino, e retomar a VM no servidor de destino. A pós-cópia visa reduzir o tempo total de migração da VM.

Para estimar o tempo de migração usando a pós-cópia, precisamos considera-se alguns fatores, como o tamanho dos dados da VM, o número de páginas de memória, o número de rodadas de cópia, o número de páginas modificadas por rodada, a largura de banda da rede e a latência da rede. Além disso, precisa-se considerar o tempo para suspender e retomar a VM no servidor de origem e destino, respectivamente.

A seguir, apresenta-se as fórmulas para calcular o tempo de migração usando a pós-cópia, bem como um exemplo numérico para ilustrar o cálculo.

Fórmulas:

- Tempo de transferência = Tamanho dos dados / largura de banda
- Tempo de cópia = Número de rodadas x (Tempo de cópia por rodada + Latência da rede)
- Tempo total de migração = Tempo de cópia + Tempo de transferência + tempo para suspender a VM + tempo para retomar a VM
- Tempo de inatividade = Tempo para suspender a VM + Tempo para retomar a VM.

#### 4.2.3 Estimativa de inatividade na política de pós-cópia

Neste tópico, vamos aborda-se o tempo de inatividade na política de pós-cópia. O tempo de inatividade é o período durante o qual a VM está suspensa e não pode processar nenhuma solicitação. O tempo de inatividade afeta a disponibilidade e a qualidade do serviço das aplicações que usam a VM.

Para calcular o tempo de inatividade na política de pós-cópia, precisa-se considerar dois fatores: o tempo para suspender a VM no servidor de origem e o tempo para retomar a VM no servidor de destino. O tempo para suspender a VM depende do tamanho total da VM e da largura de banda da rede. O tempo para retomar a VM depende do tamanho total da VM, da velocidade de leitura do armazenamento e da velocidade de escrita da memória.

A seguir, apresenta-se as fórmulas para calcular o tempo de inatividade na política de pós-cópia, bem como um exemplo numérico para ilustrar o cálculo.

Fórmulas:

- Tempo de inatividade = Tempo para suspender a VM + Tempo para retomar a VM
- Tempo para suspender a VM = (Tamanho total da VM x 8 Mb / MB) / Largura de banda
- Tempo para retomar a VM = (Tamanho total da VM) / min(Velocidade de leitura, Velocidade de escrita) + Latência

#### 4.2.4 Tamanho das VMs enviadas na política de pré-cópia

Neste tópico, vamos abordar o tamanho das VMs enviadas na política de pré-cópia com 4 interações de cópias. A pré-cópia é uma técnica que envia os dados da VM para o servidor de destino antes da migração. Ela consiste em várias rodadas de cópia, onde cada rodada envia as páginas de memória que foram modificadas desde a última rodada.

Para calcular o tamanho dos dados enviados na política de pré-cópia, precisamos considerar alguns fatores, como o tamanho total da VM, o número de páginas de memória, o número de rodadas de cópia e o número de páginas modificadas por rodada.

A seguir, apresentamos as fórmulas para calcular o tamanho dos dados enviados na política de pré-cópia, bem como um exemplo numérico para ilustrar o cálculo.

Fórmulas:

- Tamanho total da VM = Número total de páginas x Tamanho de cada página
- Tamanho dos dados enviados = Tamanho total da VM + (Número de páginas modificadas por rodada x Número de rodadas x Tamanho de cada página)

#### 4.2.2.5 Tamanho das VMs enviadas na política de pós-cópia

Neste tópico, aborda-se o tamanho das VMs enviadas na política de pós-cópia. A pós-cópia é uma técnica que envia os dados da VM para o servidor de destino após a migração. Ela consiste em suspender a VM no servidor de origem, transferir o estado da VM para o servidor de destino, e retomar a VM no servidor de destino.

Para calcular o tamanho dos dados enviados na política de pós-cópia, precisa-se considerar alguns fatores, como o tamanho total da VM e a taxa de modificação das páginas. A taxa de modificação das páginas é a fração das páginas que são alteradas durante a cópia.

A seguir, apresentamos as fórmulas para calcular o tamanho dos dados enviados na política de pós-cópia, bem como um exemplo numérico para ilustrar o cálculo.

Fórmulas:

Tamanho total da VM = Número total de páginas x Tamanho de cada página

Tamanho dos dados enviados = Tamanho total da VM x (1 + Taxa de modificação das páginas)

### 4.3 Matriz de seleção para abordagem, estratégia e política

As funções para seleção de abordagem, estratégia e política de migração descritas no tópico anterior realizam estas tarefas a partir dos parâmetros qualidade do sinal, a latência da rede, a largura de banda e o tamanho da VM e para avaliação do tempo total de migração e tempo de inatividade busca-se uma matriz de seleção baseada em experimentos realizados por Liu *et al.*, (2020), Magalhães; Soares e gomes, (2011), Rohden, (2015) e Motaki, (2021) baseados em necessidades específicas em um ambiente de computação em nuvem e névoa.

Abaixo apresenta-se necessidades comuns encontradas em um ambiente de computação em nuvem e névoa:

- Baixa demanda de largura de banda: Algumas aplicações têm uma baixa demanda de largura de banda. Nestes casos, é essencial que o algoritmo de migração seja capaz de se adaptar a estas condições, garantindo que a migração ocorra sem interrupções e sem consumir recursos desnecessários.
- Conexão de rede instável: Em áreas onde a conexão de rede é instável, o algoritmo deve ser robusto o suficiente para lidar com interrupções e flutuações na conexão. Isso pode envolver a implementação de mecanismos de escolha de um momento mais estável para realizar a migração.
- Demanda variável de recursos: A demanda por recursos pode variar dependendo da aplicação em execução na VM. O algoritmo deve ser capaz de se adaptar a estas variações, escalando os recursos para cima ou para baixo conforme necessário.
- Movimento rápido: Em cenários onde o usuário está se movendo rapidamente, como em um trem ou carro, o algoritmo deve ser capaz de prever e responder a mudanças rápidas na localização do usuário.
- Movimento previsível: Se o movimento do usuário for previsível, como em uma rotina diária, o algoritmo pode utilizar esta previsibilidade para otimizar a migração das VMs.

- Restrições de energia: As restrições de energia devem ser consideradas ao migrar VMs, especialmente em dispositivos móveis onde a vida útil da bateria é uma preocupação.

A partir das necessidades do usuário é implementado a *função\_necessidades* com objetivo principal de selecionar a abordagem, estratégia e política pré-estabelecida na matriz de seleção.

Uma matriz reduzida das necessidades do usuário consolidada a partir de experiências constantes da literatura é apresentada na Tabela 8 abaixo:

Necessidade	Abordagem	Estratégia	Política
Baixa demanda de largura de banda	Estática	E2	P1
Conexão de rede instável	Estática	E3	P2
Demanda variável de recursos	Dinâmica	E1	P2
Movimento rápido	Dinâmica	E3	P2
Movimento previsível	Estática	E2	P1
Restrições de energia	Estática	E1	P2

Tabela 8 - Matriz de necessidades usuário x abordagem, estratégia e política

Para avaliar o algoritmo de seleção proposta utiliza-se as combinações de abordagem, estratégia e política apresentadas na matriz de necessidades da Tabela 8 acima.

O algoritmo completo é apresentado no GitHub: <https://github.com/luizsales2909>. A seguir detalha-se o ambiente de teste do algoritmo.

#### 4.4 Ambiente de teste

Neste tópico descreve-se as ferramentas, cenário e procedimentos para validação do algoritmo proposto.

##### 4.4.1 Ferramentas e recursos utilizados na criação e validação do algoritmo

Na tabela 9 abaixo apresenta-se as ferramentas utilizadas para criar o algoritmo de migração de VMs.

Ferramenta	Descrição
<b>MATLAB</b>	Um <i>software</i> computação numérica (interação C, C++, Java e Python). <a href="https://matlab.mathworks.com/">https://matlab.mathworks.com/</a>
<b>QGIS</b>	Um <i>software</i> livre e de código aberto para sistemas de informações geográficas (SIG). <a href="https://qgis.org/pt_BR/site/">https://qgis.org/pt_BR/site/</a>

<b>Manhattan</b>	Um cenário para pesquisa em computação em névoa, baseado no padrão de ruas e infraestrutura da cidade de Manhattan.
<b>Notebook Dell</b>	Processador Intel(R) Core(TM) i3-8145U CPU @ 2.10GHz, 8,00 GB RAM e Sistema operacional de 64 bits.
<b>GitHub</b>	É uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. <a href="https://github.com/luizsales2909">https://github.com/luizsales2909</a>

Tabela 9 - Ferramentas e recursos utilizados na criação do algoritmo

Para criar o algoritmo de migração de VMs, escolheu-se o MATLAB como a linguagem de programação. O MATLAB é uma ferramenta poderosa e popular para pesquisa e desenvolvimento em diversas áreas, incluindo computação em névoa.

Para virtualizar as vias da cidade de Manhattan e criar as rotas necessárias para o deslocamento do usuário utilizou-se a ferramenta QGIS, um *software* livre e gratuito de Sistemas de Informação Geográfica (SIG), que permite visualizar, editar, analisar e gerenciar dados espaciais.

Na pesquisa utilizou-se um notebook Dell com processador Intel® Core™ i3-8145U CPU @ 2.10GHz, 2304 Mhz, 2 núcleos e 4 processadores lógicos e sistema operacional Windows 64. Além disso, foi consultado a biblioteca virtual CAPES PERIÓDICOS<sup>12</sup> para revisar a literatura relacionada ao tema e usou-se o gerenciador de referências Mendeley<sup>13</sup> para organizar as citações. Também se utilizou o Lucidchart, um aplicativo de diagramação inteligente para construção do fluxograma do algoritmo.

Para controle de versões e publicação do algoritmo utilizou-se o GitHub.

#### 4.4.2 Cenário Manhattan

Manhattan, localizada em Nova York, EUA, é um cenário ideal para a validação do algoritmo de computação em névoa. O padrão de ruas em formato de retângulos, altamente simétrico e previsível, facilita a modelagem e a simulação de cenários de mobilidade urbana. A simetria das ruas permite fazer suposições simplificadoras sobre o movimento dos usuários e a distribuição das *cloudlets*, que são unidades de processamento distribuídas pela cidade. Na Figura 11 abaixo tem-se o mapa da Cidade de Manhattan com as ruas vetorizadas e alguns trajetos utilizados na validação do algoritmo.

<sup>12</sup> <https://www-periodicos-capes-gov-br.ezl.periodicos.capes.gov.br>

<sup>13</sup> <https://www.mendeley.com/reference-manager/library/all-references>



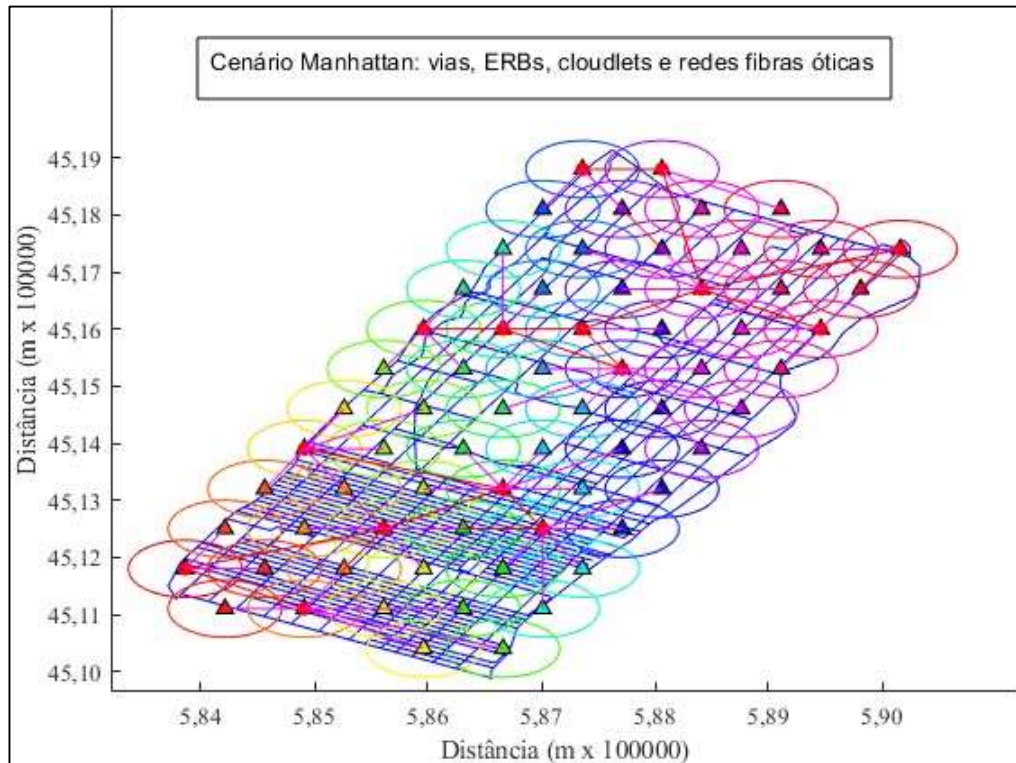


Figura 15 - Cenário Manhattan vetorizado com ERBs, *cloudlets* e rede de fibra ótica. Fonte: Própria.

Na seção seguinte, descreve-se os passos seguidos para realizar os testes no algoritmo.

#### 4.5 Procedimentos na validação do algoritmo

A validação do algoritmo é um processo crucial que garante a qualidade e eficiência do algoritmo, permitindo identificar e corrigir possíveis erros ou falhas. Ela avalia o desempenho do algoritmo em termos de confiabilidade, precisão e segurança, além de possibilitar a comparação com outros algoritmos existentes para aprimoramento contínuo. A validação também comprova a viabilidade e aplicabilidade do algoritmo em diferentes cenários ou contextos, assegurando sua eficácia e utilidade.

**Passo 1 - Preparação do Conjunto de Testes:** O primeiro passo é selecionar um conjunto de testes que represente diferentes cenários de mobilidade urbana. Para isso, escolhemos 30 trajetos distintos no cenário de Manhattan, com diferentes distâncias, sentidos e velocidades das vias. Esses trajetos serão usados para testar o desempenho do algoritmo em diferentes condições de tráfego, densidade populacional e demanda por serviços.

**Passo 2 - Carregamento do Algoritmo:** O segundo passo é carregar o algoritmo principal e as funções auxiliares no MATLAB, que é a ferramenta que usamos para



implementar e simular o algoritmo. Antes de executar o algoritmo, certifique-se de que todas as dependências estão corretamente instaladas e que o algoritmo está funcionando conforme esperado.

Passo 3 - Execução do Teste: O terceiro passo é executar o teste para cada trajeto selecionado. Para isso, rodamos o algoritmo no MATLAB e registramos as saídas geradas pelo algoritmo. Essas saídas incluem o tempo total de migração, que é o tempo gasto para transferir as máquinas virtuais (VMs) dos usuários entre as *cloudlets*; o tempo de inatividade, que é o tempo em que os usuários ficam sem conexão com a internet durante a migração; o tamanho das VMs migradas, que é a quantidade de dados que são transferidos entre as *cloudlets*; e o total de migrações realizadas, que é o número de vezes que as VMs são migradas durante o trajeto. Essas métricas nos ajudarão a avaliar o desempenho do algoritmo em termos de eficiência, confiabilidade e qualidade do serviço.

Passo 4 - Análise dos Resultados: O quarto passo é analisar os resultados obtidos após a execução do teste. Para isso, verificamos se o algoritmo está funcionando corretamente e se ele atende aos objetivos propostos. Isso pode envolver a verificação se o algoritmo é capaz de manter a conexão com a internet enquanto o usuário se move ao longo do trajeto; se ele é capaz de migrar as VMs para as *cloudlets* adequadas, levando em conta a proximidade, a disponibilidade e a capacidade das *cloudlets*; e se ele é capaz de minimizar o tempo total de migração e o tempo de inatividade, garantindo uma experiência satisfatória para os usuários.

Passo 5 - Replicação da Pesquisa: O quinto e último passo é compartilhar os resultados com outros pesquisadores para permitir a replicação da pesquisa. Isso ajudará a validar ainda mais os resultados e a melhorar o algoritmo. Para isso, disponibilizamos o código-fonte do algoritmo, as funções auxiliares, os dados dos trajetos e os resultados obtidos no repositório online.

Conclusão: Nesta seção, apresentamos os passos que seguimos para validar o nosso algoritmo de computação em névoa no cenário de Manhattan. Os resultados mostram que o nosso algoritmo é capaz de otimizar o uso dos recursos de processamento distribuídos pela cidade, considerando a mobilidade dos usuários e a dinâmica da rede. Além disso, mostramos que Manhattan é um cenário ideal para testar e validar algoritmos de computação em névoa, devido às suas características únicas que desafiam e facilitam a implementação e a simulação do algoritmo. No Capítulo seguinte será apresentado os resultados.

## 5. RESULTADOS

Neste capítulo, é apresentado os resultados obtidos em dois momentos da pesquisa. Primeiro, avaliou-se todas as métricas descritas na Tabela 7, Seção 4.6, com um único usuário se deslocando no cenário descrito na Tabela 4, Seção 4.2, para diferentes larguras de banda/latência. Depois, no final da pesquisa, avaliou-se o tempo total de migração da VM, o tempo total de inatividade da VM, a soma do tamanho das VMs enviadas e a quantidade de migrações as combinações de abordagem, estratégia e política de migração a partir da matriz de necessidade do usuário apresentadas no item.

### 5.1 RESULTADOS E DESCRIÇÃO

A Figura 13 abaixo, tem-se o resultado gráfico do deslocamento do usuário em um dos trinta trajetos possíveis na validação do algoritmo, a partir de um ponto de origem e para um ponto de destino, buscando sempre o caminho mais próximo e obedecendo o sentido das vias. A linha vermelha representa o trajeto do usuário ao longo das vias da Cidade de Manhattan, as circunferências amarelas representam as ERBs que receberam os *handoffs* e as circunferências verdes representam as *cloudlets* que receberam a migração das VMs.

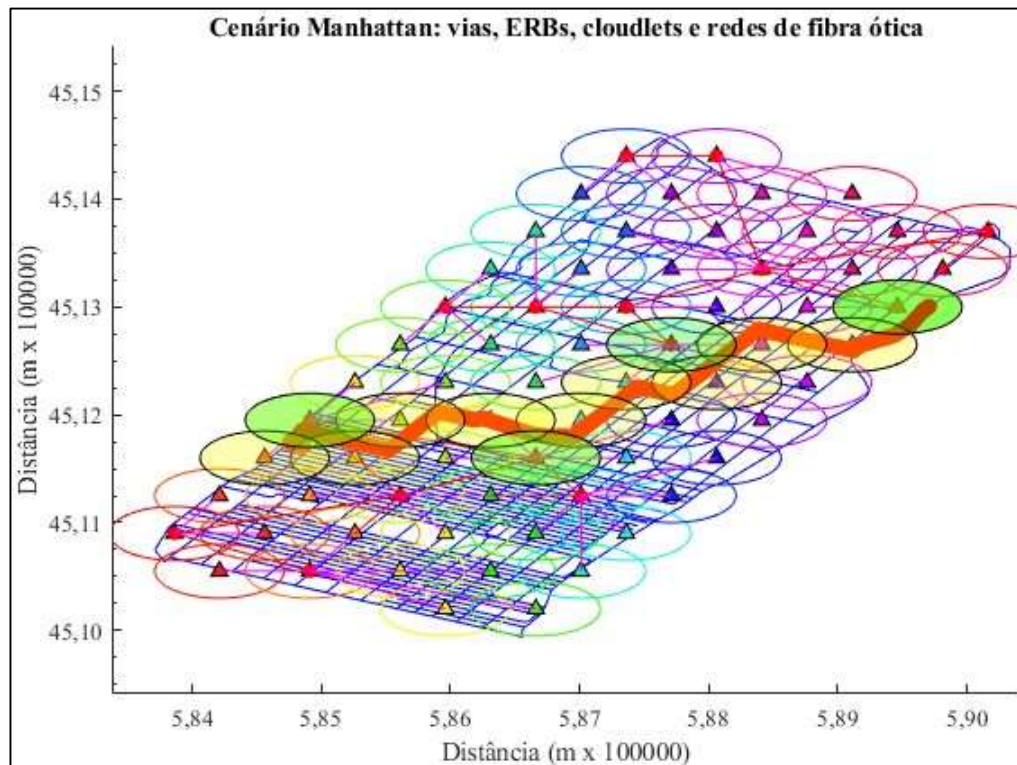


Figura 16 - Resultado gráfico de *handoffs* e migrações de VMs. Fonte: Própria.  
Fonte 1 - Elaboração própria.

Na Figura acima observa-se que ocorreram 12 *handoffs* e 5 migrações de VMs. Durante o deslocamento do usuário o algoritmo realizou a migração da VM entre os *cloudlets* e retornou o tempo total de migração da VM, o tempo de inatividade da VM, o total de migrações de VM realizadas e a soma do tamanho das VMs migradas. A seguir apresentamos os resultados obtidos

### 5.1.1 Avaliação do tempo total de migração da VM para alta demanda de largura de banda

Na Figura 14 abaixo, tem-se, no eixo vertical, a média do tempo total de migração da VM (barra azul) para todas combinações efetivadas pelo algoritmo representadas no eixo horizontal (abordagem, estratégia e política) durante os diversos deslocamentos do usuário. Por exemplo, a menor média do tempo total de migração (barra azul localizada na extremidade direita da figura) efetivada pelo algoritmo ocorreu quando o algoritmo iniciou a abordagem do ponto de migração **estática**, selecionou a *cloudlet* conectada ao ponto de acesso mais próximo do usuário no momento da abordagem (E3) e migrou a VM para a *cloudlet* selecionada utilizando a política **pré-cópia**, no entanto, para a necessidade **alta demanda de largura de**

**banda** a matriz de seleção indicou a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlets* de destino com menor latência (**E1**) e migrou a VM para *cloudlet* selecionada utilizando a política pós-cópia, conforme representado na barra vermelha da figura. A seguir apresenta-se a análise do resultado obtido com o algoritmo de seleção e a com a matriz de necessidades.

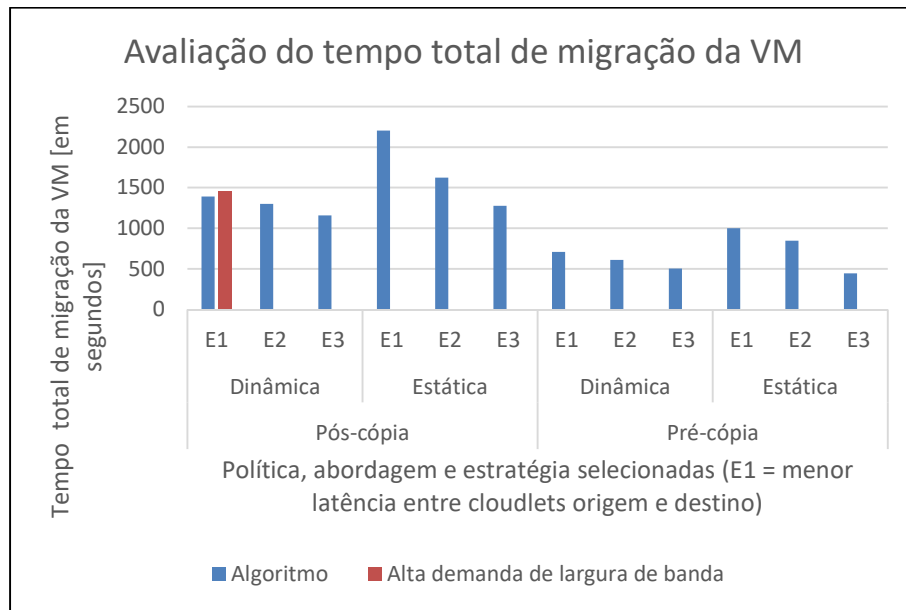


Figura 17 – Avaliação do tempo total de migração. Fonte: Própria.

- E1: Estratégia de migração para menor latência entre o *cloudlet* de origem e um dos *cloudlets* de destino.  
 E2: Estratégia de migração para menor distância entre o usuário e um dos *cloudlets* de destino.  
 E3: Estratégia de migração para *cloudlet* conectado ao PA mais próximo do usuário.

Na Figura 15 abaixo, a barra azul representa a média do tempo total de migração da VM quando o algoritmo selecionou, a partir dos parâmetros de entrada (qualidade do sinal, largura de banda da rede, latência da rede e tamanho da VM), a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlet* com menor latência com relação à *cloudlet* de origem (**E1**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pós-cópia**. A barra vermelha representa o mesmo que a barra azul, no entanto, a combinação de abordagem, estratégia e política foi pré-estabelecida a partir de trabalhos anteriores, para fins de comparação com algoritmo aqui proposto. O algoritmo de seleção apresentou uma **redução** na média do tempo total de migração em **4,5%** em relação ao tempo obtido com a matriz de seleção para a necessidade alta demanda de largura de banda.

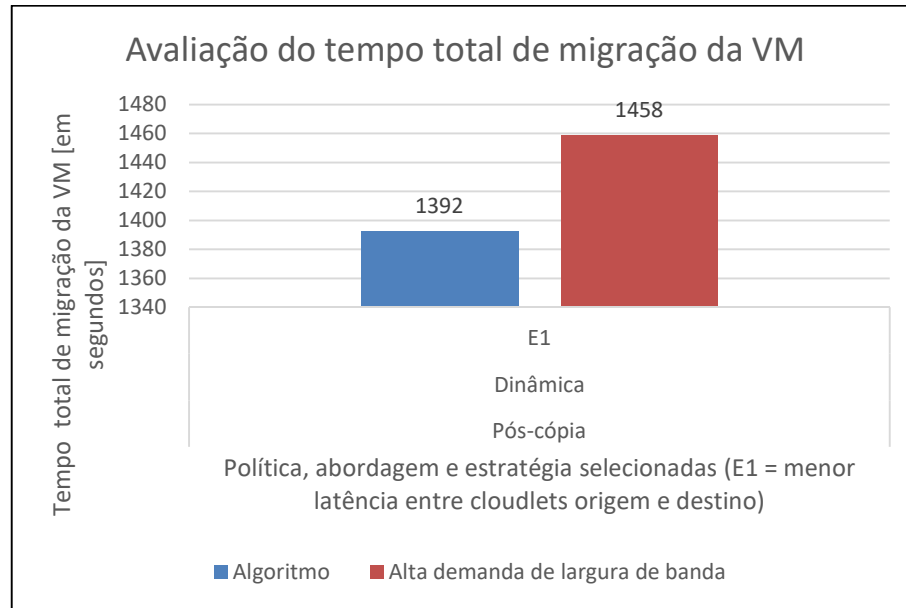


Figura 18 – Média do tempo total de migração da VM com algoritmo e matriz de seleção. Fonte: Própria.

### 5.1.2 Avaliação do tempo em que a VM está indisponível (inatividade) para alta demanda de largura de banda

Na Figura 16 abaixo, tem-se, no eixo vertical, a média do tempo de inatividade da VM (barra azul) para todas combinações efetivadas pelo algoritmo representadas no eixo horizontal (abordagem, estratégia e política) durante os diversos deslocamentos do usuário. Por exemplo, a menor média do tempo de inatividade da VM (barra azul localizada na extremidade direita da figura) efetivada pelo algoritmo ocorreu quando o algoritmo iniciou a abordagem do ponto de migração **estática**, selecionou a *cloudlet* conectada ao ponto de acesso mais próximo do usuário no momento da abordagem (**E3**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pré-cópia**, no entanto, para a necessidade **alta demanda de largura de banda**, a matriz de decisão indicou a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlets* de destino com menor latência (**E1**) e migrou a VM para *cloudlet* selecionada utilizando a política **pós-cópia**, conforme representado na barra vermelha da figura. A seguir apresenta-se a análise do resultado obtido com o algoritmo de seleção e com a matriz de necessidades.

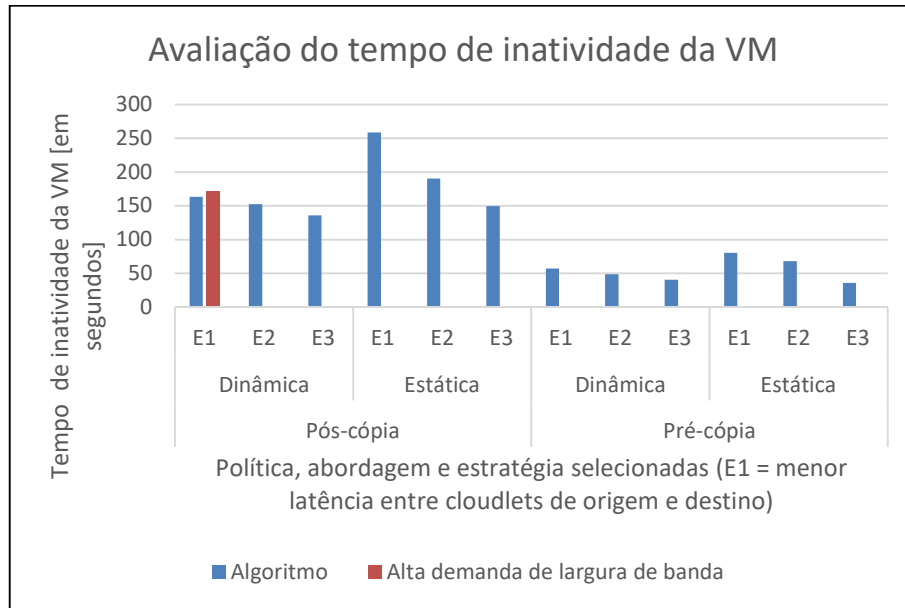


Figura 19 - Avaliação do tempo de inatividade da VM. Fonte: Própria.

E1: Estratégia de migração para menor latência entre o *cloudlet* de origem e um dos *cloudlets* de destino.

E2: Estratégia de migração para menor distância entre o usuário e um dos *cloudlets* de destino.

E3: Estratégia de migração para *cloudlet* conectado ao PA mais próximo do usuário.

Na Figura 17 abaixo, a barra azul representa a média do tempo de inatividade da VM quando o algoritmo selecionou, a partir dos parâmetros de entrada (qualidade do sinal, largura de banda da rede, latência da rede e tamanho da VM), a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlet* com menor latência com relação à *cloudlet* de origem (**E1**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pós-cópia**. A barra vermelha representa o mesmo que a barra azul, no entanto, a combinação de abordagem, estratégia e política foi pré-estabelecida a partir de trabalhos anteriores, para fins de comparação com algoritmo aqui proposto. O algoritmo de seleção apresentou uma **redução** na média do tempo de inatividade da VM em **4,7%** em relação ao tempo obtido com a matriz de seleção para necessidade alta demanda de largura de banda.

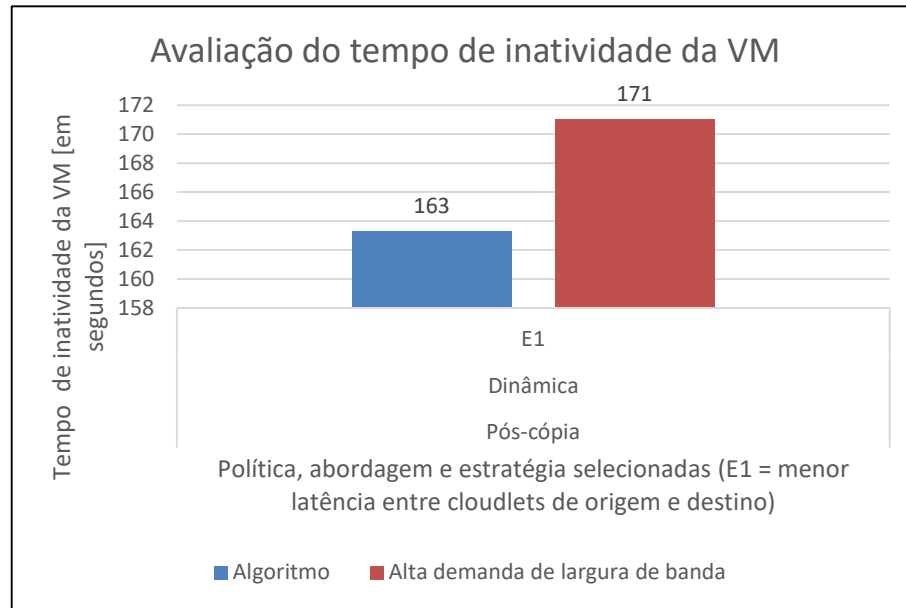


Figura 20 – Avaliação do tempo em que a VM está indisponível (inatividade). E1, E2 e E3<sup>14</sup>. Fonte: Própria.

### 5.1.3 Avaliação do tempo total de migração da VM para movimento rápido

Na Figura 18 abaixo, tem-se, no eixo vertical, a média do tempo total de migração da VM (barra azul) para todas combinações efetivadas pelo algoritmo representadas no eixo horizontal (abordagem, estratégia e política) durante os diversos deslocamentos do usuário. Por exemplo, a menor média do tempo total de migração (barra azul localizada na extremidade direita da figura) efetivada pelo algoritmo ocorreu quando o algoritmo iniciou a abordagem do ponto de migração **estática**, selecionou a *cloudlet* conectada ao ponto de acesso mais próximo do usuário no momento da abordagem (**E3**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pré-cópia**, no entanto, para a necessidade **movimento rápido** a matriz de seleção indicou a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlets* conectada ao PA mais próximo do usuário no momento da abordagem (**E3**) e migrou a VM para *cloudlet* selecionada utilizando a política **pós-cópia**, conforme representado na barra vermelha da figura.

<sup>14</sup> **E1**: Estratégia de migração para menor latência entre os *cloudlets* de destino;

**E2**: Estratégia de migração para menor distância entre o usuário e um dos *cloudlets* de destino; **E3**: Estratégia de migração *cloudlet* conectada ao PA mais próximo do usuário no momento da abordagem.

A seguir apresenta-se a análise do resultado obtido com o algoritmo de seleção e a com a matriz de necessidades.

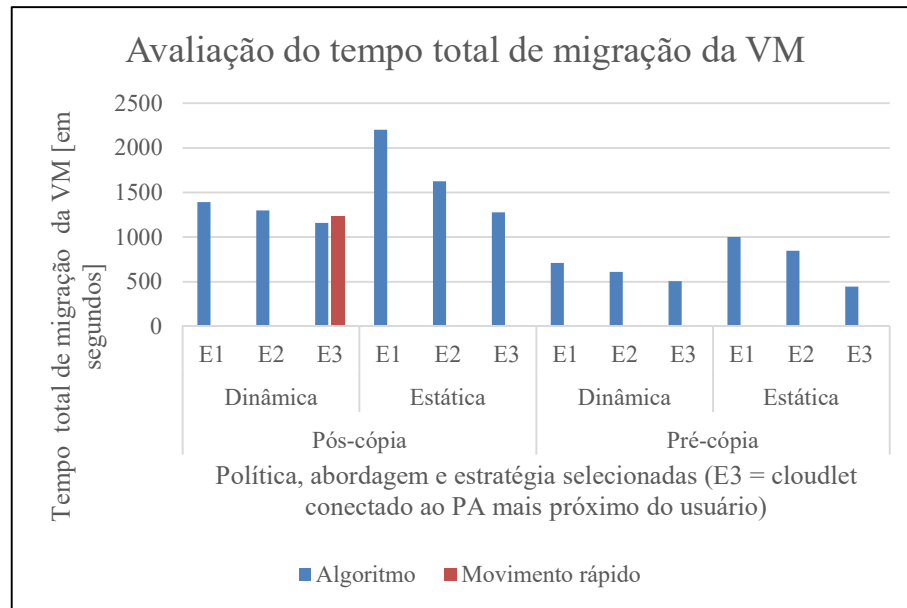


Figura 21 - Avaliação do tempo total de migração. Fonte: Própria.

Na Figura 19 abaixo, a barra azul representa a média do tempo total de migração da VM quando o algoritmo selecionou, a partir dos parâmetros de entrada (qualidade do sinal, largura de banda da rede, latência da rede e tamanho da VM), a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlet* conectado ao PA mais próximo do usuário no momento da abordagem (**E3**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pós-cópia**. A barra vermelha representa o mesmo que a barra azul, no entanto, a combinação de abordagem, estratégia e política foi pré-estabelecida a partir de trabalhos anteriores, para fins de comparação com algoritmo aqui proposto. O algoritmo de seleção apresentou uma **redução** na média do tempo de inatividade da VM em **6,2%** em relação ao tempo obtido com a matriz de seleção para a necessidade movimento rápido.



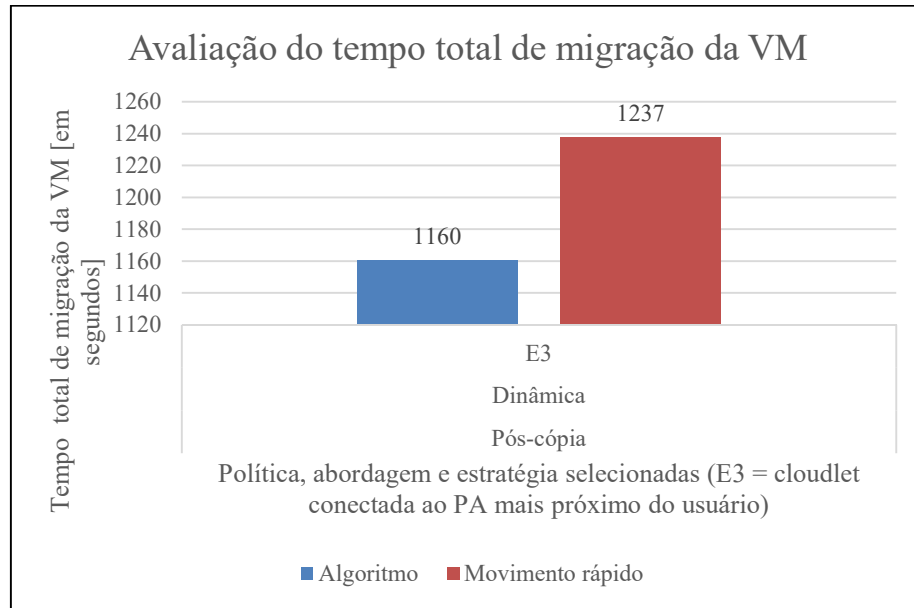


Figura 22 - Média do tempo total de migração da VM com algoritmo e matriz de seleção. Fonte: Própria.

#### 5.1.4 Avaliação do tempo em que a VM está indisponível (inatividade) para movimento rápido

Na Figura 20 abaixo, tem-se, no eixo vertical, a média do tempo de inatividade da VM (barra azul) para todas combinações efetivadas pelo algoritmo representadas no eixo horizontal (abordagem, estratégia e política) durante os diversos deslocamentos do usuário. Por exemplo, a menor média do tempo de inatividade da VM (barra azul localizada na extremidade direita da figura) efetivada pelo algoritmo ocorreu quando o algoritmo iniciou a abordagem do ponto de migração **estática**, selecionou a *cloudlet* conectada ao ponto de acesso mais próximo do usuário no momento da abordagem (**E3**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pré-cópia**, no entanto, para a necessidade **movimento rápido**, a matriz de seleção indicou a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlets* conectada ao PA mais próximo do usuário no momento da abordagem (**E3**) e migrou a VM para *cloudlet* selecionada utilizando a política **pós-cópia**, conforme representado na barra vermelha da figura. A seguir apresenta-se a análise do resultado obtido com o algoritmo de seleção e com a matriz de necessidades.

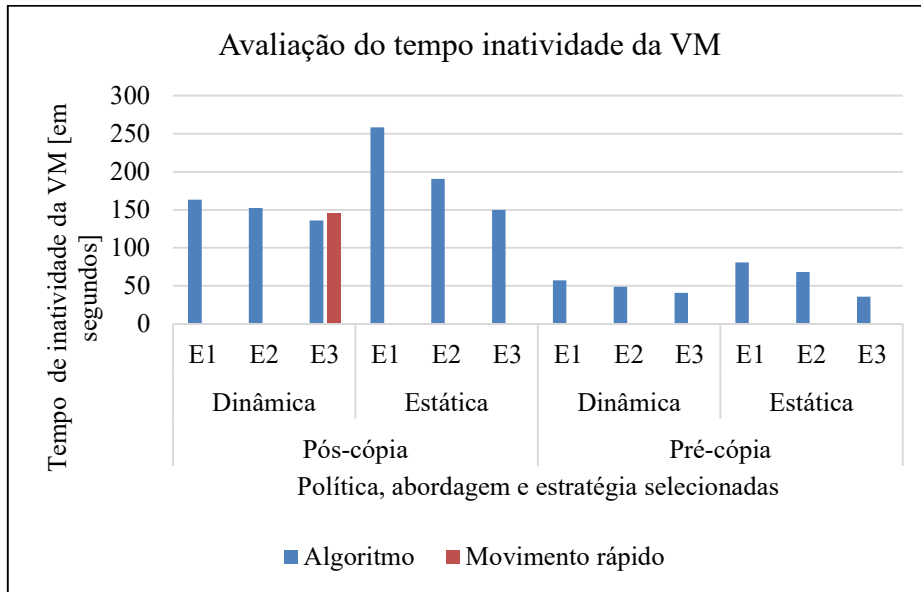


Figura 23 - Avaliação do tempo de inatividade da VM. Fonte: Própria.

Na Figura 21 abaixo, a barra azul representa a média do tempo de inatividade da VM quando o algoritmo selecionou, a partir dos parâmetros de entrada (qualidade do sinal, largura de banda da rede, latência da rede e tamanho da VM), a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlet* com menor latência com relação à *cloudlet* de origem (**E1**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pós-cópia**. A barra vermelha representa o mesmo que a barra azul, no entanto, a combinação de abordagem, estratégia e política foi pré-estabelecida a partir de trabalhos anteriores, para fins de comparação com algoritmo aqui proposto. O algoritmo de seleção apresentou uma **redução** na média do tempo

de inatividade da VM em **6,2%** em relação ao tempo obtido com a matriz de seleção para a necessidade movimento rápido.

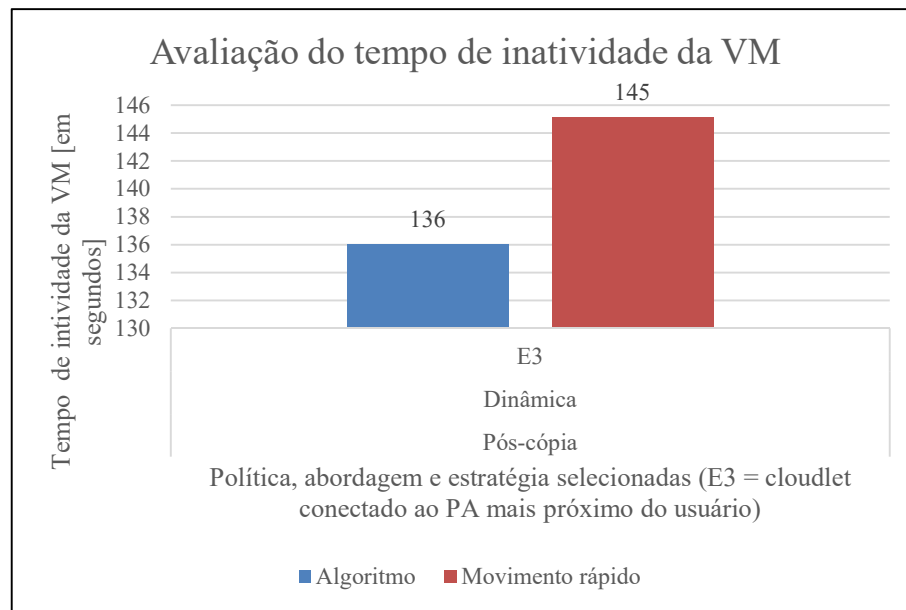


Figura 24 – Avaliação do tempo em que a VM está indisponível (inatividade). Fonte: Própria.

### 5.1.5 Avaliação da soma do tamanho da VM migrada para alta demanda de largura de banda

Na Figura 22 abaixo, tem-se, no eixo vertical, a média da soma do tamanho da VM migrada (barra azul) para todas combinações efetivadas pelo algoritmo de seleção representadas no eixo horizontal (abordagem, estratégia e política) durante os diversos deslocamentos do usuário. Por exemplo, a menor média do tamanho da VM migrada (barra azul localizada na extremidade direita da figura) efetivada pelo algoritmo ocorreu quando o algoritmo iniciou a abordagem do ponto de migração **estática**, selecionou a *cloudlet* conectada ao ponto de acesso mais próximo do usuário no momento da abordagem (**E3**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pré-cópia**, no entanto, para a necessidade **alta demanda de largura de banda**, a matriz de seleção indicou a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlets* de destino com menor latência (**E1**) e migrou a VM para *cloudlet* selecionada utilizando a política **pós-cópia**, conforme representado na barra vermelha da figura. A seguir apresenta-se a análise do resultado obtido com o algoritmo de seleção e com a matriz de necessidades.

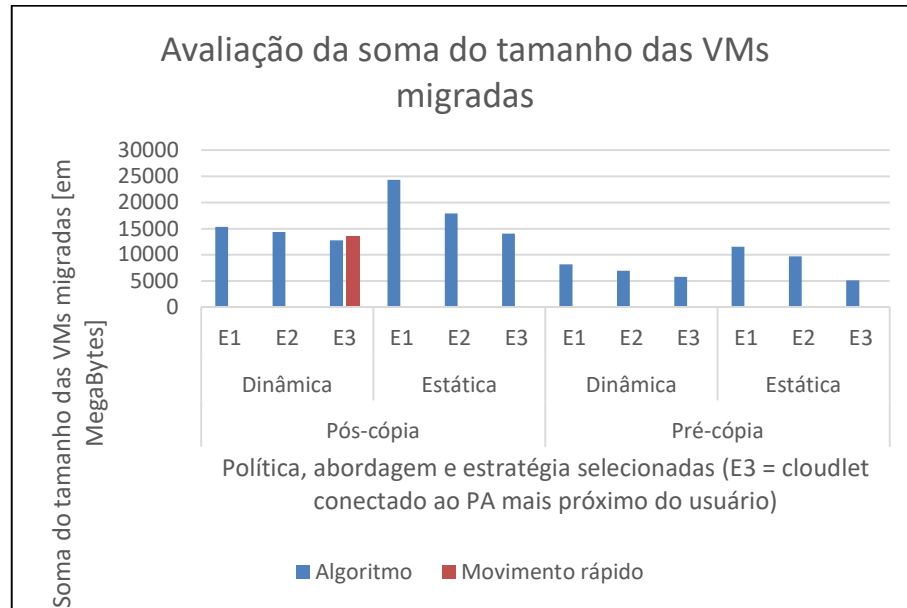


Figura 25 - Avaliação da soma do tamanho das VMs migradas. Fonte: Própria.

Na Figura 23 abaixo, a barra azul representa a média da soma do tamanho da VM migrada quando o algoritmo selecionou, a partir dos parâmetros de entrada (qualidade do sinal, largura de banda da rede, latência da rede e tamanho da VM), a abordagem do ponto de migração **dinâmica**, selecionou a *cloudlet* com menor latência com relação à *cloudlet* de origem (**E1**) e migrou a VM para a *cloudlet* selecionada utilizando a política **pós-cópia**. A barra vermelha representa o mesmo que a barra azul, no entanto, a combinação de abordagem, estratégia e política foi pré-estabelecida a partir de trabalhos anteriores, para fins de comparação com algoritmo aqui proposto. O algoritmo de seleção apresentou uma **redução** na média da soma do tamanho da VM migrada em **12,8%** em relação à média obtida com a matriz de seleção para a necessidade alta demanda de largura de banda.

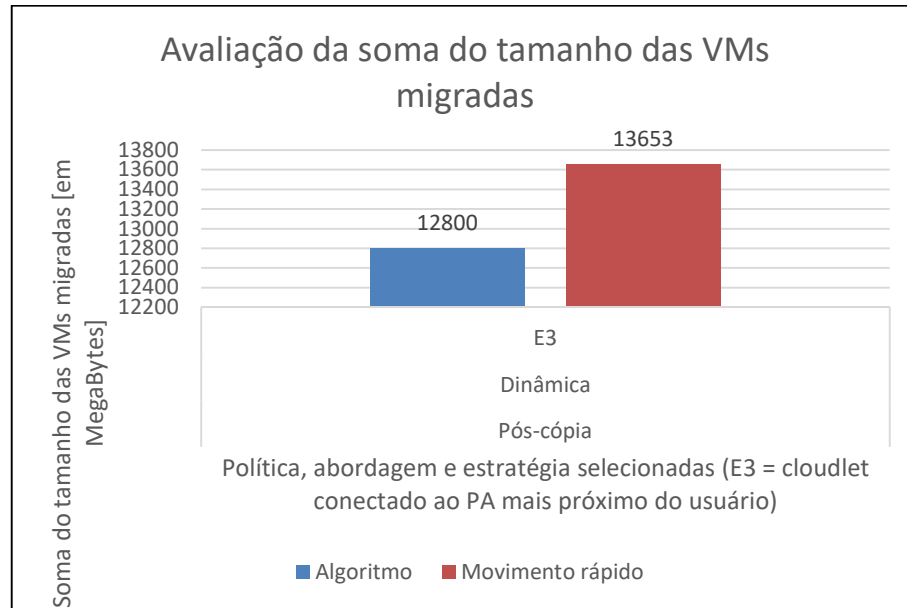


Figura 26 - Avaliação da soma do tamanho das VMs migradas. Fonte: Própria.

## 5.2 ANÁLISE DOS RESULTADOS

Os resultados apresentados indicam que o algoritmo de seleção é eficaz em melhorar a eficiência da migração de VMs em vários aspectos:

**Redução do tempo total de migração:** O algoritmo de seleção reduziu o tempo total de migração em 4,5% em comparação com a matriz de seleção para alta demanda de largura de banda. Isso sugere que o algoritmo de seleção é mais eficiente em termos de tempo quando se trata de migração de VMs que requerem alta largura de banda.

**Redução do tempo de inatividade da VM:** O algoritmo de seleção reduziu o tempo de inatividade da VM em 4,7% para alta demanda de largura de banda e 6,2% para movimento rápido em comparação com a matriz de seleção. Isso indica que o algoritmo de seleção minimiza efetivamente o tempo de inatividade da VM, o que pode ser crucial para manter a continuidade do serviço e a experiência do usuário.

**Redução do tamanho da VM migrada:** O algoritmo de seleção reduziu a soma do tamanho da VM migrada em 12,8% em comparação com a matriz de seleção para alta demanda de largura de banda. Isso sugere que o algoritmo de seleção é capaz de realizar migrações mais eficientes em termos de recursos, o que pode levar a economias significativas em termos de uso de largura de banda e armazenamento.

Em resumo, o algoritmo de seleção parece oferecer melhorias significativas em termos de eficiência de migração de VMs em comparação com a matriz de seleção, particularmente para cenários que exigem alta demanda de largura de banda ou movimento rápido.

---

## 6. CONCLUSÕES FINAIS

Neste trabalho foi desenvolvido um algoritmo que seleciona políticas de migração de máquinas virtuais com base em parâmetros de entrada do sistema, como qualidade do sinal, largura de banda da rede, latência da rede e tamanho dos dados das máquinas virtuais migradas. O algoritmo foi projetado para escolher as melhores abordagens do ponto de migração, as melhores *cloudlets* de destino e as melhores técnicas de migração das máquinas virtuais, levando em consideração os níveis dos parâmetros de entrada.

Foram realizadas uma série de simulações para avaliar o desempenho do algoritmo proposto em comparação com um algoritmo baseado em uma matriz de decisão. As métricas de desempenho consideradas incluíram o tempo total de migração da máquina virtual, o tempo de inatividade da máquina virtual e o tamanho dos dados das máquinas virtuais migradas.

Os resultados das simulações indicaram que a migração proativa de máquinas virtuais pode reduzir significativamente o tempo de resposta e o consumo de energia, antecipando a mudança de localização do usuário e alocando os recursos mais próximos. Além disso, é possível afirmar que a migração de máquinas virtuais pode ser afetada por vários fatores, incluindo a carga de trabalho, o tamanho da memória, a largura de banda da rede e o algoritmo de migração utilizado. No entanto, a migração de máquinas virtuais pode ser otimizada por meio de técnicas de balanceamento de carga.

Com base nesses resultados, propõe-se para trabalhos futuros a exploração do uso de aprendizado de máquina e inteligência artificial para prever a mobilidade do usuário e a demanda de recursos em computação em névoa.

Para trabalhos futuros, propõe-se também a criação de novos algoritmos de migração de máquinas virtuais que levem em conta aspectos como segurança, privacidade, confiabilidade e escalabilidade dos serviços em computação em névoa. A ideia é desenvolver soluções robustas e eficientes que garantam uma migração segura e eficiente, preservando a privacidade dos dados e mantendo a confiabilidade e escalabilidade dos serviços.

- AHMAD, Ishrat; BAKHT, Humayun; MOHAN, Uma. Cloud Computing – A Comprehensive Definition. **Journal of Computing and Management Studies**, [s. l.], v. 1, n. 1, p. 30–2017, 2017.
- BELGACEM, Ali; MAHMOUDI, Saïd; FERRAG, Mohamed Amine. A machine learning model for improving virtual machine migration in cloud computing. **Journal of Supercomputing**, [s. l.], v. 79, n. 9, p. 9486–9508, 2023. Disponível em: <https://doi.org/10.1007/s11227-022-05031-z>.
- BITTENCOURT, Luiz Fernando *et al.* Towards Virtual Machine Migration in Fog Computing. **Proceedings - 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2015**, [s. l.], p. 1–8, 2015a.
- BITTENCOURT, Luiz Fernando *et al.* Towards Virtual Machine Migration in Fog Computing. **Proceedings - 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2015**, [s. l.], p. 1–8, 2015b.
- CHEN, Ji Ming *et al.* A Virtual Machine Migration Strategy Based on the Relevance of Services against Side-Channel Attacks. **Security and Communication Networks**, [s. l.], v. 2021, 2021.
- CHIN, Jeannette; CALLAGHAN, Vic; BEN ALLOUCH, Somaya. The Internet-of-Things: Reflections on the past, present and future from a user-centered and smart environment perspective. **Journal of Ambient Intelligence and Smart Environments**, [s. l.], v. 11, n. 1, p. 45–69, 2019.
- DE MATOS, Alexandre Veloso; MAZIERO, Carlos Alberto. Uma arquitetura para a Gerência de Migração de Máquinas Virtuais. *In: , 2021. Anais [...]. [S. l.: s. n.], 2021.*
- DLAMINI, Sabelo; VENTURA, Neco. Resource management in fog computing: Review. **icABCD 2019 - 2nd International Conference on Advances in Big Data, Computing and Data Communication Systems**, [s. l.], p. 1–7, 2019.
- FAHIMULLAH, Muhammad; AHVAR, Shohreh; TROCAN, Maria. A Review of Resource Management in Fog Computing: Machine Learning Perspective. [s. l.], p. 1–15, 2022. Disponível em: <http://arxiv.org/abs/2209.03066>.
- FRASER, Keegan; FITCHETT, Jennifer M. Live Migration of Virtual Machines. **Weather, Climate, and Society**, [s. l.], v. 14, n. 1, p. 273–286, 2022.
- GONÇALVES, DIOGO M.; BITTENCOURT, LUIZ F.; MADEIRA, Edmundo M. R.. Análise da Predição de Mobilidade na migração de Aplicações em Computação em Névoa. **SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC)**, [s. l.], v. 37, n. 2177–9384, p. 580–593, 2019. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/7388/7272>. Acesso em: 12 mar. 2023.
- GONCALVES, Diogo *et al.* Dynamic network slicing in fog computing for mobile users in MobFogSim. **Proceedings - 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing, UCC 2020**, [s. l.], p. 237–246, 2020.
- GONCALVES, Diogo *et al.* Proactive Virtual Machine Migration in Fog Environments. **Proceedings - IEEE Symposium on Computers and Communications**, [s. l.], v. 2018-June, p. 742–745, 2018.



- HE, Tianzhang. Migration Management in Networking-enabled Edge and Cloud Computing Environments. **THE UNIVERSITY OF MELBOURNE**, [s. l.], 2021.
- JALALI MOGHADDAM, Marjan *et al.* Minimizing virtual machine migration probability in cloud computing environments. **Cluster Computing**, [s. l.], v. 23, n. 4, p. 3029–3038, 2020. Disponível em: <https://doi.org/10.1007/s10586-020-03067-5>.
- KHALEEL, Mustafa I.; ZHU, Michelle M. Adaptive virtual machine migration based on performance-to-power ratio in fog-enabled cloud data centers. **Journal of Supercomputing**, [s. l.], v. 77, n. 10, p. 11986–12025, 2021. Disponível em: <https://doi.org/10.1007/s11227-021-03753-0>.
- LIU, Xialin *et al.* Virtual Machine Consolidation with Minimization of Migration Thrashing for Cloud Data Centers. **Mathematical Problems in Engineering**, [s. l.], v. 2020, n. Vm, 2020.
- LOPES, Márcio Moraes *et al.* MyifogSim: A simulator for virtual machine migration in fog computing. **UCC 2017 Companion - Companion Proceedings of the 10th International Conference on Utility and Cloud Computing**, [s. l.], n. Vm, p. 47–52, 2017.
- MAGALHÃES, Deborah V; SOARES, José Marques; GOMES, Danielo G. Análise do Impacto de Migração de Máquinas Virtuais em Ambiente Computacional Virtualizado. **XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, [s. l.], n. May, p. 235–248, 2011. Disponível em: [http://sbrc2011.facom.ufms.br/files/main/ST05\\_3.pdf](http://sbrc2011.facom.ufms.br/files/main/ST05_3.pdf).
- MORAES, Gilmar *et al.* Mecanismo de Comunicação para Migração de Serviços Ciente da Localização de Nuvem e Névoas. [s. l.], p. 81–90, 2020.
- MOTAKI, Saloua El. Um modelo baseado em previsão para monitoramento de migração ao vivo de máquina virtual em um datacenter de nuvem. [s. l.], 2021.
- ORACLE, Brasil. **Virtualização vs. Computação na nuvem**. [S. l.], 2023. Disponível em: <https://www.oracle.com/br/cloud/what-is-cloud-computing/virtualization-vs-cloud-computing/>. Acesso em: 4 jun. 2023.
- PULIAFITO, Carlo *et al.* MobFogSim: Simulation of mobility and migration for fog computing. **Simulation Modelling Practice and Theory**, [s. l.], v. 101, p. 102062, 2020.
- ROHDEN, Rafael Barasuol. Especificação de Uma Arquitetura para Migração de Máquinas Virtuais Utilizando Ontologias. [s. l.], 2015.
- ROSÁRIO, Denis *et al.* Service migration from cloud to multi-tier fog nodes for multimedia dissemination with QoE support. **Sensors (Switzerland)**, [s. l.], v. 18, n. 2, p. 1–17, 2018.
- SHAKARAMI, Ali *et al.* Resource provisioning in edge/fog computing: A Comprehensive and Systematic Review. **Journal of Systems Architecture**, [s. l.], v. 122, n. November 2021, p. 102362, 2022. Disponível em: <https://doi.org/10.1016/j.sysarc.2021.102362>.
- SOLOMON, S.; ALLEY, R. A Closer Look at the. **Science**, [s. l.], v. 60, n. June, p. 658–674, 2014. Disponível em: [http://www.ey.com/Publication/vwLUAssets/Applying\\_IFRS:\\_A\\_closer\\_look\\_at\\_the\\_new\\_revenue\\_recognition\\_standard\\_\(June\\_2014\)/\\$FILE/Applying-Rev-June2014.pdf](http://www.ey.com/Publication/vwLUAssets/Applying_IFRS:_A_closer_look_at_the_new_revenue_recognition_standard_(June_2014)/$FILE/Applying-Rev-June2014.pdf).
- TANG, Zhiqing *et al.* Migration Modeling and Learning Algorithms for Containers in Fog Computing. **IEEE Transactions on Services Computing**, [s. l.], v. 12, n. 5, p. 712–725, 2019.
- TININI, Rodrigo Izidoro *et al.* 5GPy: A SimPy-based simulator for performance evaluations

in 5G hybrid Cloud-Fog RAN architectures. **Simulation Modelling Practice and Theory**, [s. l.], v. 101, n. October 2019, p. 102030, 2020. Disponível em: <https://doi.org/10.1016/j.simpat.2019.102030>.

TORRE, Ennio *et al.* A dynamic evolutionary multi-objective virtual machine placement heuristic for cloud data centers. **Information and Software Technology**, [s. l.], v. 128, n. December 2019, p. 106390, 2020. Disponível em: <https://doi.org/10.1016/j.infsof.2020.106390>.

VELASQUEZ, Karima *et al.* Fog orchestration for the Internet of Everything: state-of-the-art and research challenges. **Journal of Internet Services and Applications**, [s. l.], v. 9, n. 1, 2018.

VENÂNCIO, Giovanni; TURCHETTI, Rogério C.; DUARTE, Elias P. NFV-COIN: Unleashing The Power of In-Network Computing with Virtualization Technologies. **Journal of Internet Services and Applications**, [s. l.], v. 12, n. 1, 2022.