Leonardo Vieira de Albuquerque Rezende

Posicionamento de *Edge Servers* para aplicações de baixa latência em *Distributed Cloud* no município de Vitória/ES

Viçosa, MG

2021

Leonardo Vieira de Albuquerque Rezende

Posicionamento de *Edge Servers* para aplicações de baixa latência em *Distributed Cloud* no município de Vitória/ES

Trabalho apresentado ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia II – e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Rodolpho Vilela Alves Neves

Viçosa, MG

2021

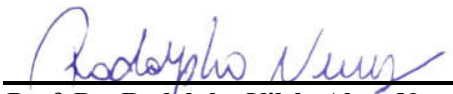
LEONARDO VIEIRA DE ALBUQUERQUE REZENDE

POSICIONAMENTO DE *EDGE SERVERS* PARA APLICAÇÕES DE BAIXA LATÊNCIA EM *DISTRIBUTED CLOUD* NO MUNICÍPIO DE VITÓRIA/ES

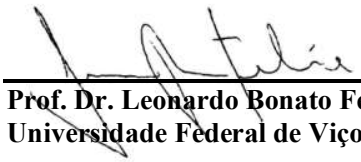
Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia II e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 20 de agosto de 2021.

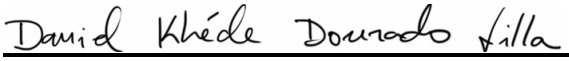
COMISSÃO EXAMINADORA



Prof. Dr. Rodolpho Vilela Alves Neves - Orientador
Universidade Federal de Viçosa



Prof. Dr. Leonardo Bonato Felix - Membro
Universidade Federal de Viçosa



Prof. Me. Daniel Dourado Khede Villa - Membro
Universidade Federal de Viçosa

"So understand
Don't waste your time always searching for those wasted years
Face up, make your stand
Realize you're living in the golden years."
(Adrian Smith)

Agradecimentos

Agradeço primeiramente a Deus por tornar isto tudo possível, e à toda minha família, principalmente minha mãe Renata, meus avós Marlucia, Ivan, Nega e Silvério, meu padrasto Lucas, e meu tio Silvinho, por todo o suporte e por apoiarem o meu sonho sempre.

Agradeço ao meu orientador Rodolpho, por sempre ter paciência de ensinar, pelos conselhos, e por sempre me mostrar o caminho do conhecimento.

Agradeço a Fernanda pelo carinho e por ter me apoiado em todos os momentos, tornando possível a execução deste trabalho em meio as situações adversas.

Agradeço aos meus amigos mais próximos de Viçosa, em especial Matheus, Patrícia, Lucas, Antônio, Adriana e Thaís por sempre me apoiarem em todos os momentos vividos na universidade, seja nos cafés, nas festas, na alegria de passar em uma disciplina difícil ou no desespero das inúmeras entregas ao final dos semestres. Agradeço também aos funcionários, técnicos e professores do DEL por estarem sempre disponíveis a ajudar quando preciso. Também agradeço aos meus amigos fora de Viçosa pela ajuda nos últimos anos, em especial Eduardo, que me ajudou a entender um pouco mais de programação orientada ao objeto.

Resumo

Conforme a demanda por tecnologias como 5G, Indústria 4.0 e IoT crescem a cada ano, os estudos acerca de temas como *edge cloud* se mostram cada vez mais emergentes e necessários para atender os requisitos das novas configurações de rede. Devido a expectativa de crescimento na geração de dados (*Big Data*) e de dispositivos interconectados, o tema *Mobile Edge Computing* se mostra um fator importante na redução de custos e melhora do *Quality of Service* (QoS), ao se posicionar *edge servers* (ESVs) que comportam como pequenos *data centers* próximos ao usuário, responsáveis por aplicações de baixa latência e possibilitando a criação de novos serviços. Este trabalho é um estudo de caso do problema de posicionamento de *edge servers* para aplicações de baixa latência na cidade de Vitória, no Brasil. Primeiro são definidas as funções de minimização: *delay* de acesso e *workload balancing*. Para os testes são utilizados dados abertos da ANATEL contendo as latitudes e longitudes de cada estação rádio-base (ERB). Em seguida são testados três algoritmos: K-means, Random e Top-K e é avaliada a performance de cada um deles na minimização das funções escolhidas, através de dois métodos de posicionamento de ESVs: o primeiro inserindo as ERBs e posicionando os ESVs a uma proporção de 0,1, e o segundo método posicionando os ESVs a partir da localização de 139 ERBs do *dataset*. O método de posicionamento variando as ERBs minimizou o *delay* de acesso em torno de -49,88% (K-means) até -53,75% (Random), porém não obteve reduções significativas no balanceamento da carga de trabalho. O método de posicionamento variando os ESVs obteve reduções de até -89,67% (K-means), e reduções acima de 90% para os três algoritmos, com destaque para o Top-K (-98,35%). Após a análise dos resultados, dois pontos de operação são escolhidos, um atendendo o *delay* de acesso, utilizando K-means com 14 ESVs, e o outro atendendo a demanda, com Top-K e 12 ESVs, e então é apresentado um mapa da cidade de Vitória-ES com as ERBs e as localizações sugeridas para os ESVs em cada ponto de operação. Logo, o K-means se mostra um método mais eficaz para a redução do *delay* de acesso, mostrando que o posicionamento planejado dos ESVs pode acelerar a implementação de aplicações de baixa latência.

Palavras-Chave: Mobile Edge Computing; 5G; Distributed Cloud; Smart Cities; Access Delay; Workload Balancing, Clustering.

Abstract

As the demand for technologies like 5G, Industry 4.0 and IoT grows, the studies about edge cloud are emerging and it is necessary to meet the requirements of the new network configuration. Due to the huge expected growth in data generation (Big Data) and an increase in interconnected devices, the Mobile Edge Computing topic is an important theme in reducing costs and improving QoS by positioning edge servers (ESV) that behave as small data centers near the user, which are responsible for low latency applications and enabling the creation of new services. This paper is a case study of the problem of positioning edge servers for low latency applications in the city of Vitoria (Brazil). First the minimization functions are defined: access delay and workload balancing. For the dataset, open data from ANATEL is used, containing the latitudes and longitudes of each base station (ERB). Then three algorithms are tested: K-means, Random and Top-K and their performance in minimizing the chosen functions is evaluated, through two methods of ESV positioning: the first one inserting the ERBs and positioning the ESVs at a proportion of 0.1, and the second method positioning the ESVs from the location of 139 ERBs in the dataset. The positioning method varying the ERBs minimized the access delay around -49.88% (K-means) to -53.75% (Random), but it did not achieve significant reductions in workload balancing. The positioning method varying the ESVs obtained reductions of up to -89.67% (K-means), and reductions above 90% for the three algorithms, especially Top-K (-98.35%). After the analysis of the results, two operations points are chosen, one serving the access delay, using K-means with 14 ESVs, and the other serving the workload balancing, using Top-K and 12 ESVs, and then a map of the city of Vitoria is presented with the ERBs and the suggested locations for the ESVs for each chosen point of operation. K-means then proves to be a more effective method for reducing access delay, showing that planned placement of ESVs can speed up the deployment of low latency applications.

Key-words: Mobile Edge Computing; 5G; Distributed Cloud; Smart Cities; Access Delay; Workload Balancing, Clustering.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

3G	Terceira geração de padrões de tecnologia de redes móveis
3GPP	*3rd Generation Partnership Project*
4G	Quarta geração de padrões de tecnologia de redes móveis
4K	Resolução de TV quatro vezes maior que o HDTV (3840 x 2160)
5G	Quinta geração de padrões de tecnologia de redes móveis
AP	*Access Point*
CDN	*Cloud Distributed Networks*
CSP	*Communication Service Providers*
eMBB	*Enhanced Mobile Broadband*
ERB	Estação Rádio Base
ESV	*Edge Server*
FPS	Frames por segundo
GSM	*Global System for Mobile Communications*
HDTV	*High Definition Television*
Hz	Hertz, unidade de frequência
IA	Inteligência Artificial
IoT	Internet das coisas
ITU	*International Telecommunication Union*
K	Número de *Edge Servers*
MIMO	*Multiple Input Multiple Output*
ML	*Machine Learning*
mMTC	*massive machine type communications*
N	Número de estações rádio base

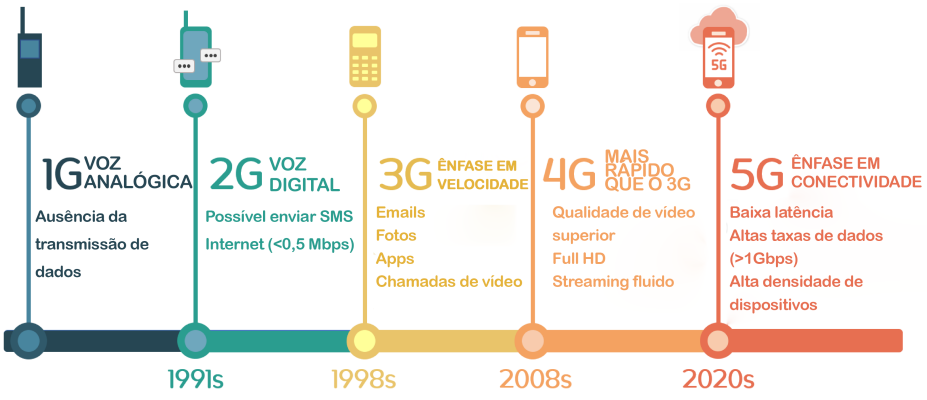
PaaS	*Platform as a Service*
PC	Computador Pessoal
POO	Programação Orientada ao Objeto
QoS	*Quality of Service*
SaaS	*Software as a Service*
IaaS	*Infrastructure as a Service*
UE	User Equipment
URRLC	*Ultra-Reliable Low Latency Communication*

Sumário

1 Histórico das redes móveis

As redes dos sistemas de comunicação móvel são tipicamente classificados em gerações. A primeira geração, nos anos 1980, teve o objetivo de fornecer serviços de voz, baseada em circuitos analógicos em comutação. Devido a uma baixa interoperabilidade entre as tecnologias, a primeira geração não foi amplamente adotada. A partir da segunda geração, começaram a ter padrões mais consolidados pelo 3GPP, como o GSM, padrão muito conhecido nas redes de comunicação móvel 2G nos anos 1990, que oferecia voz e um baixo volume de dados para os assinantes. A terceira geração móvel, chamada de 3G, nos anos 2000, teve como objetivo de projeto melhorar os serviços de dados fornecidos pelas redes móveis para os assinantes, já havendo aplicações de rede como ligações de vídeo, navegação pela internet móvel, jogos *online*, além de envio de imagens e vídeos. Similar ao 3G, a quarta geração móvel (4G), a partir de 2008, teve seu desenvolvimento impulsionado pela demanda dos assinantes e da indústria, a partir da criação de novos serviços que requeriam taxas de transferências de dados mais altas do que as que existiam na época, potencializando o acesso a banda larga, TV móvel, chamadas por vídeo, serviços de *streaming* e televisão (HDTV), todos em alta definição. Para a quinta geração móvel (5G), as evoluções técnologicas dos serviços em nuvem (*cloud services*) motivaram a nova revisão da arquitetura de rede, através da redução da complexidade do *hardware*, eficiência operacional e agilidade na introdução de novos serviços. Os modelos de serviços em nuvem como *Software as a Service* (SaaS), *Platform as a Service* (PaaS) e *Infrastructure as a Service* (IaaS) proporcionam infinitas possibilidades de aplicações a partir de 2020. (MANNWEILER et al., 2020). A Figura 1 propociona uma visão geral da evolução da rede móvel ao longo das gerações.

Figura 1 – Evolução das redes móveis ao longo das gerações.



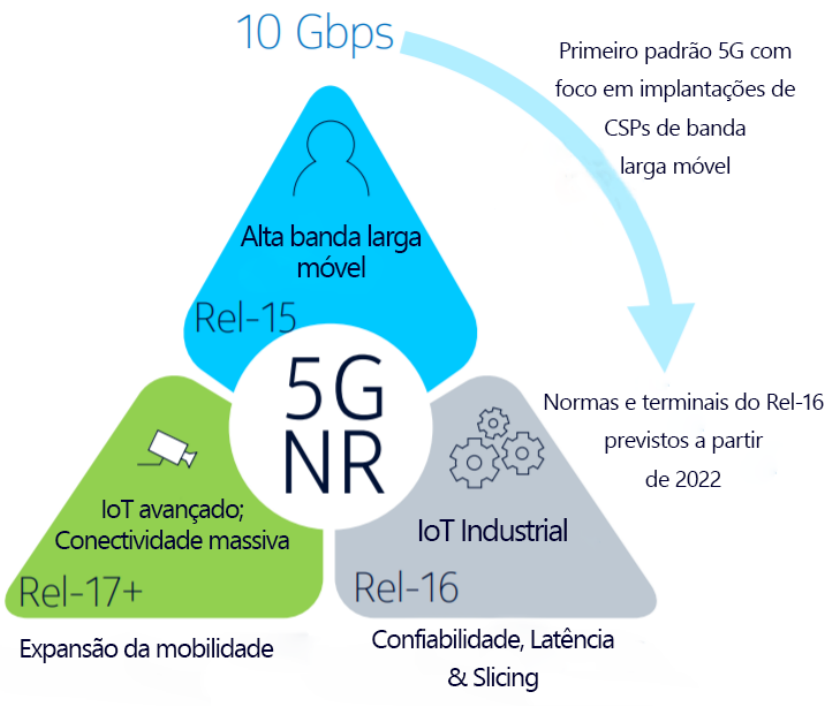
Fonte: Adaptado de 5glearning.org (2021).

1.1 5G

O 5G possui três características principais não vistas na geração anterior. Primeiramente, uma quantidade enorme de dados é gerada. De acordo com a *International Telecommunication Union* (ITU), existiam mais de 7,5 bilhões de dispositivos móveis no mundo em 2017 (YORK; POYNTER, 2018), e o número aumentou para 25 bilhões em 2020 (CUI; WANG, 2020). Em segundo lugar, requerimentos exigentes de *Quality of Service* (QoS) são impostos para aplicações interativas que demandam baixa latência (*ultra-low latency*) e alto *throughput*. Além disso, a terceira característica, um ambiente heterogêneo deve ser suportado para permitir a interoperabilidade em uma diversa gama de *User Equipments* (UEs), como *smartphones* e *tablets*.

Para atender a todas essas demandas, a Figura 2 mostra como o 5G é dividido em três etapas ou, como são chamadas,*releases*: Rel-15, Rel-16 e Rel-17+. Cada uma destas *releases* possibilita a geração de novos serviços através de três domínios de casos de uso principais, nomeados *enhanced mobile broadband* (eMBB), *Ultra-Reliable Low Latency* Communication (URLLC) e *massive machine type communications (mMTC)*.

Figura 2 – Domínios de cada fase do 5G.



Fonte: Adaptado de Ghosh et al. (2019).

A Rel-15, primeira fase do 5G, focada nas implementações dos primeiros UEs 5G disponíveis comercialmente no mundo, nas frequências sub-6 GHz e *mmWave*, destinadas ao aumento da banda disponível. A segunda fase do 5G, Rel-16, é focada em novas *features* como aplicações de *ultra-low latency*, proporcionando latências de 1 ms, *network slicing*,

massive MIMO, entre outras, potencializando a Internet das Coisas Industrial (I-IoT). Já na terceira fase, Rel-17+, ainda estão sendo discutidos os principais requerimentos, porém além da amplificação de todas as *features* anteriores em escala, há o domínio do *Edge Computing*, da utilização de *Machine Learning* (ML) e Inteligência Artificial (IA), e de redes mais integradas por utilizar, armazenar e aproveitar melhor os dados (*Massive Data Collection* e *Data Analytics*) (GHOSH et al., 2019).

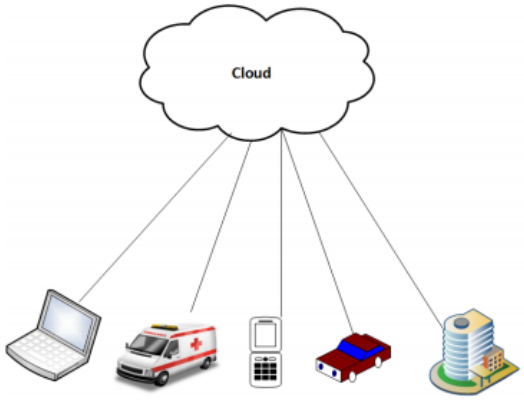
1.2 Distributed Cloud e Edge Computing

Um dos principais *drivers* do 5G em relação a baixa latência é o conceito de *distributed cloud*. A computação distribuída em nuvem, *distributed cloud*, é o primeiro modelo de nuvem que incorpora a localização física dos serviços prestados em nuvem como parte de sua definição. Historicamente, a localização não tem sido relevante para as definições de *cloud computing*. Na verdade, a localização tem sido explicitamente abstraída do serviço, o que inspirou o termo "computação em nuvem" em primeiro lugar. Estima-se que até 2024 a maioria das plataformas de serviços em nuvem fornecerá pelo menos alguns serviços distribuídos em nuvem executados em seu ponto de necessidade (GARTNER, 2021). A tradicional computação em nuvem, que é um paradigma de computação centralizada que fornece acesso contínuo a *data centers* altamente capacitados, tem sido adotado para permitir que os UEs possam escoar esse tráfego de dados e armazenamento para estes *data centers* (SHUJA et al., 2017). Grande parte desses dados salvos na nuvem são raramente utilizados, causando um desperdício de recursos. Tratar esses dados localmente pode ser mais relevante em muitos casos com a ascensão do IoT e dos dispositivos móveis, que gera muito mais dados que um UE tradicional. Desta forma, *Edge Computing* se mostra como uma poderosa ferramenta para tornar possível a descentralização dos processos e melhorar a performance da nuvem (GUSEV; DUSTDAR, 2018). Portanto, *Edge Computing*, como mostrado na Figura 3, refere-se à localização e execução de cargas de trabalho de aplicações o mais próximo fisicamente possível do local onde os dados são criados - por exemplo, onde os usuários estão interagindo com dispositivos como telefones celulares ou leitores de código de barras, ou onde dispositivos IoT como câmeras de segurança ou sensores de máquinas estão coletando e gerando dados (IBM, 2020).

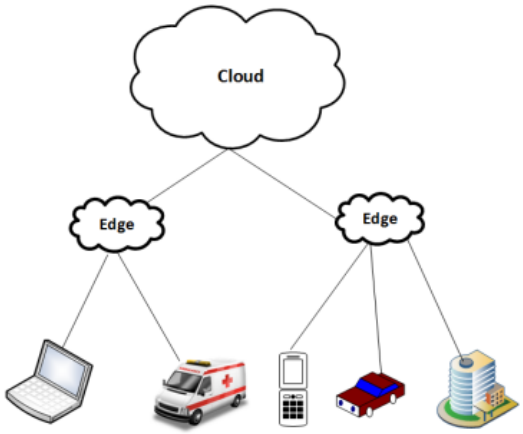
Aplicações de baixa latência

A latência é o atraso antes que uma transferência de dados comece, seguindo uma instrução para sua transferência, e é considerada uma questão importante para os fornecedores globais de *Cloud Distributed Networks* (CDN). O vídeo *on-demand*, por exemplo, se tornou rapidamente o meio dominante pelo qual os usuários finais consomem vídeo, já que o vídeo evoluiu a partir de pequenos clipes de vídeo de baixa qualidade em pequenas janelas em PCs no início dos anos 2000 para os 1080p e até 4k de conteúdo

Figura 3 – Cloud Computing vs Edge Computing.



(a) Cloud computing.



(b) Edge computing.

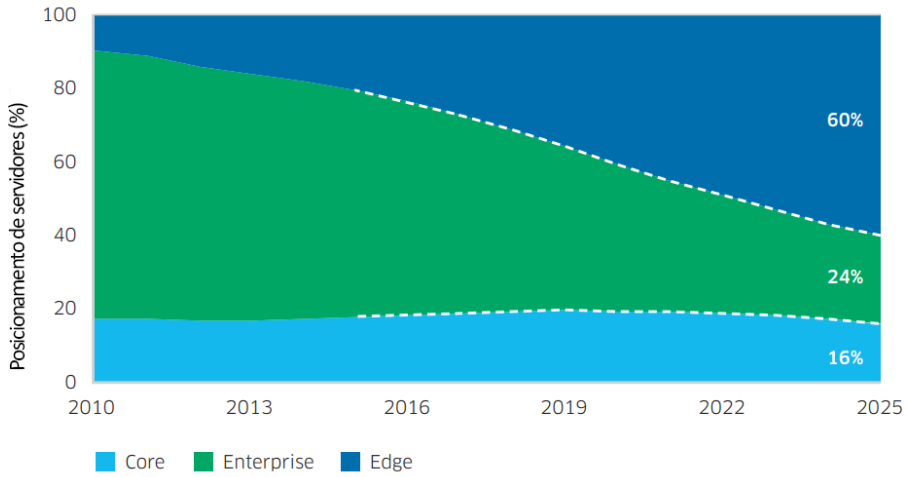
Fonte: Extraído de Hassan, Yau e Wu (2019).

a partir de 2015. O conhecido *streaming* de filmes da Netflix foi introduzido em 2008 e por volta de 2011 já havia 22 milhões de assinantes do serviço de *streaming*, e no ano de 2020 já contava com mais de 204 milhões de assinantes (FOLHA, 2021). A crescente demanda desses serviços faz os provedores estabelecerem acordos com operadoras locais e regionais para prover servidores próximos aos clientes de forma a reduzir custos de entrega e melhorar a performance dos serviços. O *Bell Labs Consulting* prevê um crescimento desta demanda para os próximos cinco anos, e que 60% dos servidores estarão mais próximos aos clientes (WELDON, 2016), na *Edge Cloud*, como mostrado na Figura 4.

Muitas aplicações de 5G dependem do *Edge Computing* para interação em tempo real, processamento local, alta taxa de dados e alta disponibilidade (HASSAN; YAU; WU, 2019), como:

- *Healthcare*: seja em cirurgia remota, monitoramento e diagnósticos em tempo real.
- Entretenimento: *streaming* HDTV, 4K, 3D em tempo real.

Figura 4 – A crescente mudança para o Edge Cloud.



Fonte: Adaptado de Weldon (2016).

- URLLC: aplicações de baixa latência com alta confiabilidade.
- Internet das Coisas: tais como aparelhos inteligentes que conectam dispositivos (por exemplo, eletrodomésticos) à internet.
- Fábricas do Futuro: tais como máquinas inteligentes, para aumentar a segurança e a produtividade. Os operadores podem usar uma plataforma remota para operar máquinas pesadas, particularmente aquelas localizadas em locais de difícil acesso e inseguros, a partir de um lugar seguro e confortável.
- Sistema de transporte inteligente: em que os motoristas podem compartilhar ou reunir informações dos centros de informações de trânsito para evitar veículos que estejam em perigo, ou parar abruptamente em tempo real, de forma a evitar acidentes. Além disso, veículos autônomos não tripulados podem sentir seu entorno e mover-se com segurança.

Quando se fala sobre tecnologia e humanos, alguns valores são levados em consideração, como por exemplo um piscar de olhos, que demora aproximadamente 150 ms, e os impulsos nervosos conseguem alcançar velocidades máximas de 100 m/s pelo corpo humano, portanto, o tempo para um sinal ser propagado do cérebro (sem contar o tempo que o cérebro demora pra processar essa informação) é de aproximadamente 10 ms. Como a latência da rede se aproxima deste mesmo intervalo, é possível permitir interações com um objeto distante sem diferença percebida em comparação com interações com um objeto local. Uma latência suficientemente baixa permitirá que os seres humanos interajam com objetos virtuais renderizados na rede como se estivessem fisicamente co-localizados. A Tabela 1 mostra a demanda de delay máximo tolerado para determinadas aplicações específicas em contraste com a resposta neurológica humana para a aplicação mencionada,

Tabela 1 – Requisitos para aplicações de baixa latência.

Aplicação	Delay máximo	Tempo de resposta neurológica
Negociação de Alta Frequência (HFT)	<1 ms	0.240 ms
Realidade Virtual VR em games	7 ms	13 ms
Veículos autônomos conectados à nuvem	10 ms	90 ms
Cloud gaming (jogos em nuvem)	20 ms	90 ms
VOIP/Video Conferência	150 ms	100 ms
Carregamento de página web	150 ms	1000 ms
Chat de mensagens instantâneas	150 ms	4000+ms

Fonte: Adaptado de (WELDON, 2016).

é importante ressaltar que que nem todas as aplicações possuem os requerimentos mais extremos para serem executadas. Algumas aplicações que não são de tempo real são suportadas pelas CDN atuais, sendo que por exemplo, um pequeno *delay* de um segundo é tolerável para esperar o carregamento de uma página, ou então, utilizar um chat de mensagens instantâneas permite que os usuários também tenham a tolerância de alguns segundos de *delay* (WELDON, 2016). Todavia, há um conjunto de aplicações que não são adequadas para as latências de mais de 100 ms características das implementações atuais, incluindo aquelas que requerem tempos de resposta neurológica humana para serem eficazes (por exemplo, realidade aumentada), e algumas que envolvem *loops* de controle e máquinas que têm tolerância muito baixa para atraso, como *cloud-assisted driving*. Este conjunto de aplicações só é possível, na prática, com o crescimento da *Edge Cloud*.

Um dos desafios relacionados à implementação de *Distributed Cloud* de forma eficaz é o posicionamento dos *edge servers* (ESV), servidores de borda mais próximos do usuário final, pois as localizações são fatores críticos determinantes no *delay* de acesso e na utilização dos recursos, especialmente quando se fala em *smart cities* que incluem uma quantidade significativa de estações rádio base (ERB). Portanto, o posicionamento de forma estratégica desses servidores pode melhorar a performance de várias aplicações na área de *Distributed Cloud*.

1.3 Revisão Bibliográfica

Em relação ao problema de posicionamento de ESVs especificamente para ambientes de redes móveis, que é o tema deste trabalho, Wang et al. (2019b) e Guo et al. (2020), utilizam métodos de clusterização para posicionar servidores na cidade de Xangai, na China, avaliando a performance dos modelos em relação ao *delay* de acesso e ao *workload*. Lee, Lee e Shin (2019) também utilizam o *dataset* de Xangai para fazer análises do posicionamento de servidores para uma rede móvel 5G.

Wang et al. (2019a) disserta acerca da coordenação de microsserviços para a decomposição de *web services* pesados e centralizados em vários pequenos serviços independentes mais simples, aplicando esse conceito ao ambiente de computação em redes móveis, mais especificamente para os problemas de *delay* de acesso.

Xu et al. (2015) analisa o problema do posicionamento de *Cloudlets*, computadores que são instalados em Wi-Fi APs em uma rede e agem para fazer o *offloading* dos dados dos usuários móveis. Existe uma similaridade na literatura entre o posicionamento de *Cloudlets* e de ESVs, sendo que Jia, Cao e Liang (2015) propuseram um modelo de sistema de *offloading* de tarefas de múltiplos usuários, estudando o posicionamento destes *Cloudlets* em relação à alocação dos usuários da rede para os mesmos. Em seguida, foi elaborado um algoritmo para colocação destes *Cloudlets* em regiões com alta densidade de usuários, enquanto eram equilibradas as cargas de trabalho.

Zeng et al. (2019) estuda sobre o posicionamento de ESVs em redes *wireless* metropolitanas (WMAN), buscando minimizar a quantidade de ESVs e atender aos requerimentos de QoS e configurações dos servidores *on-demand* utilizando teoria dos grafos. Ainda sobre QoS, Gupta, Jain e Amgoth (2018) analisam o posicionamento de máquinas virtuais montando uma infraestrutura em nuvem *"QoS-Aware"*, propondo um algoritmo chamado vScale.

Lu et al. (2020) e Cui et al. (2020) focam no desenvolvimento de algoritmos que consideram a análise da robustez dos servidores na definição do problema de posicionamento, visando aumentar a qualidade da experiência dos usuários.

Yuan, Sun e Lou (2020) propõem um esquema dinâmico de posicionamento de *edge nodes* virtuais utilizando *deep learning*, no qual a estratégia de colocação é gerada com base nas informações de previsão, e esse esquema é aplicado de forma *pay-as-you-go* e visa a diminuição de custos e flexibilidade nas demandas dos usuários finais, mais voltado para a precificação de serviços em tempo real.

Xiao et al. (2018) apresentam uma análise heurística de previsão para otimização no mapeamento de colocação dos ESVs, em que o raio de servidores candidatos a serem escolhidos pelo algoritmo é reduzido prevendo o próximo destino da fonte dos dados. O servidor troca informações com a fonte pelo mecanismo de nomenclatura dos dados (*data naming*), acumulando os recursos dos servidores de cada candidato mapeando subtarefas, e assim determinando a localização e número do servidor de acordo com a quantidadede recursos, também sendo um trabalho de minimização de custos para as prestadoras de serviços.

Li et al. (2020) analisam o *trade-off* entre o *delay* de resposta e o consumo de energia em ambientes de rede heterogêneos e não-uniformes, visando minimizar a sobrecarga do sistema para posicionar os servidores em um ponto de acesso ótimo através de um algoritmo

adaptativo de *clustering* denominado Microstructural Topology Optimization (MTO).

Santoyo-González e Cervelló-Pastor (2018) estudam detalhadamente os fatores que levam a uma implementação de *Edge Cloud* para o 5G de maneira eficaz, como latência, *throughput*, confiabilidade, restrições locais e posicionamento das funções de rede virtuais.

Wang et al. (2019c) utilizam estratégias de *reinforcement learning*, com métodos de *Q-Mapping* para resolver problemas de otimização e congestionamento em relação a computação de borda, focados na escalabilidade da rede.

Todas as pesquisas citadas têm relação direta com este trabalho no sentido da otimização da rede móvel para um melhor aproveitamento dos recursos do sistema, seja no posicionamento e na escolha dos dispositivos utilizados, na otimização dos algoritmos, na análise de funções como *delay de acesso*, *workload*, redução dos custos de energia e de infraestrutura, visando uma transição para um novo mundo das telecomunicações e das novas aplicações que são potencializadas.

1.4 Objetivo

Este trabalho tem como objetivo geral estudar o problema do posicionamento de ESVs, visando testar algoritmos para verificar se o posicionamento dos ESVs de maneira estratégica causaria um impacto no *delay* de acesso e na carga de trabalho, denomimada *workload*. Portanto, os objetivos específicos deste trabalho são:

- Obter um *dataset* contendo as ERBs da cidade de Vitória-ES e fazer um tratamento destes dados.
- Determinar os parâmetros e funções de minimização a serem otimizados para avaliar o desempenho dos métodos a serem aplicados no dataset.
- Desenvolver um código para testar três métodos para clusterização no posicionamento de ESV no local: K-means, Top-K e Random.
- Analisar a performance dos modelos e gerar uma visualização geográfica, contendo as ERBs e os ESVs de um ponto de operação do modelo escolhido.

2 Referencial Teórico

Neste capítulo serão apresentados os elementos de rede utilizados, ERBs e ESVs, além de introduzir os materiais e métodos e as conclusões da literatura tomadas como referencial teórico para o problema do posicionamento de ESVs.

2.1 Estação Rádio Base (ERB)

Uma rede móvel consiste de uma grande quantidade de ERBs fixas que determinam a cobertura de rádio da região determinada. Uma ERB é o principal ponto de comunicação entre o usuário e a rede móvel, ou seja, são *transceivers* que recebem e enviam sinais dos dispositivos dos usuários (HOLMA; TOSKALA; NAKAMURA, 2020).

As ERBs possuem uma grande variedade de tamanhos (Macrocell, Smallcell, Microcell) e de frequências nas quais elas trabalham, seja em 4G ou 5G. A Figura 5 mostra uma visão comum de ERB que é presenciada no cotidiano através das torres de telefonia.

Figura 5 – Torre de telefonia contendo ERBs.

Fonte: Extraído de Padriñán (2017).

2.2 Edge Server (ESV)

Um *Edge Server* é um servidor fisicamente mais próximo dos usuários e dispositivos conectados, potencializando aplicações de baixa latência, implicando em um controle local dos dispositivos, mesclando funcionalidades de rádio e *core* em um mesmo local (AHMAD et al., 2020). A Figura 6 mostra um exemplo de ESV. A posição dos servidores em relação a infraestrutura de *Cloud* e as vantagens das aplicações de baixa latência é explicada no capítulo 1.

Figura 6 – Exemplo de *edge server* da fabricante Nokia.

Fonte: Extraído de Nokia (2020).

2.3 Posicionamento de ESVs

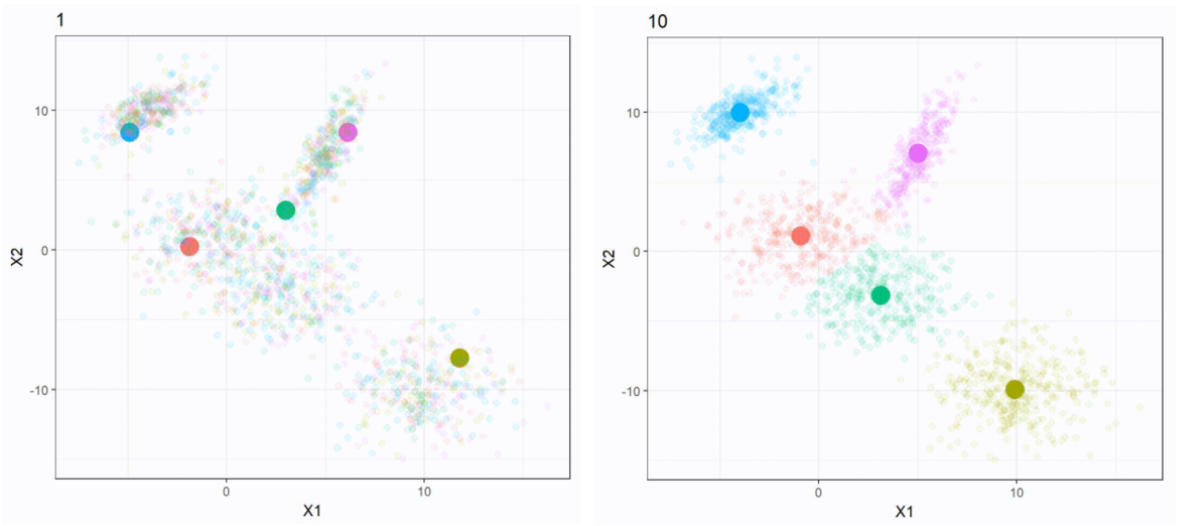
Foram testados três métodos para avaliar o posicionamento dos servidores: K-Means, Random e Top-K. Cada método se baseia em um fator determinante para o posicionamento dos servidores: distância, demanda de usuários e aleatório, a fim de se ter um parâmetro de qual método é mais eficaz na redução das funções propostas. O aleatório é colocado em contraste com os dois outros métodos para se investigar se um posicionamento inteligente de ESVs é mais eficaz do que inserir os ESVs de qualquer forma nas cidades que demandam serviços de *edge cloud*.

K-Means

O método *K-Means clustering* é uma forma bem comum de agrupar objetos (WAGSTAFF et al., 2001). O algoritmo k-means, ajusta a classificação das observações em *clusters* e atualiza a posição dos centroides até que esta posição esteja estável após sucessivas iterações, a Figura 7 apresenta uma exemplificação dos centroides na iteração 1 e após sucessivos ajustes na iteração 10. Neste código, é utilizada a biblioteca Scipy,

com a função vqkmeans, que é uma implementação do algoritmo em que a estabilidade dos centroides é determinada pela comparação do valor absoluto da variação da distância Euclidiana média entre as observações e seus correspondentes centroides em relação a um limiar. Isto produz um livro de códigos mapeando os centroides para códigos e vice-versa (ARTHUR; VASSILVITSKII, 2006). Aplicando o método a este trabalho, o centroide é onde está posicionado o ESV e ele utiliza as distâncias das ERBs, atendendo as estações mais próximas do servidor. Esta abordagem tem o foco em minimizar o *delay* de acesso devido às iterações de alocações dos *clusters*.

Figura 7 – K-means clustering na iteração 1 e após 10 iterações de ajuste dos centróides.



Fonte: Extraído de Lim (2018).

Random

Esta abordagem consiste em selecionar as localizações da ERB k aleatoriamente para as ESVs. Cada ERB é alocada um ESV, que se encontra mais próximo da mesma. A diferença deste método para o K-means é que invés fazer as iterações a partir das distâncias dos centroides para as ERBs, ele não tem um critério de posicionamento definido pela distância ou *workload*.

Top-K

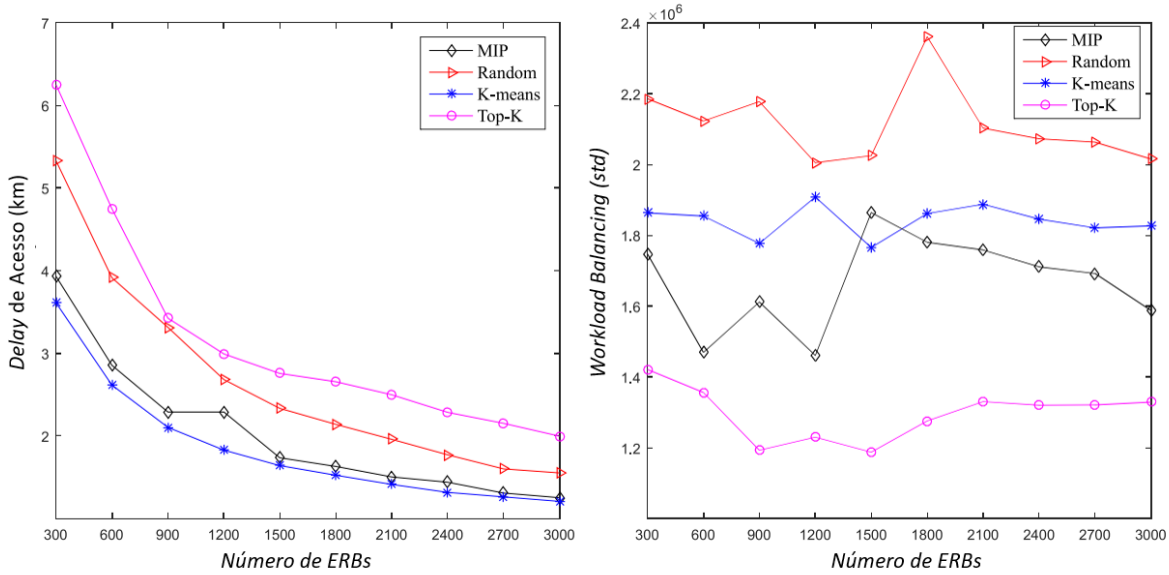
A abordagem Top-K posiciona cada servidor de borda K_{ESV} na mesma localização das ERBs com o maior *workload*. A premisssa adotada é que as ERBs mais ocupadas têm mais *requests* de usuários e precisam ser atendidas com prioridade. A diferença deste método para o K-means ou Random é que não é considerado o fator de distância envolvido nas iterações anteriores, posicionando os ESVs prioritariamente nos locais de alta demanda e atendendo os servidores de menor carga em seguida.

Exemplos da literatura

Wang et al. (2019b) posiciona ESVs na cidade de Xangai utilizando os mesmos algoritmos deste trabalho (K-Means, Top-K e Random), com exceção do Mixed Integer Programming (MIP), que é um método de confecção dos próprios autores e fora do escopo deste trabalho. São utilizados dois métodos para o posicionamento dos ESVs: variando o número de ERBs e variando o número de ESVs, que serão descritos no capítulo 3 e testados os três algoritmos para cada método.

A Figura 8 apresenta os resultados do método de posicionamento de ESVs variando o número de ERBs. E notável método K-means se destacou em relação aos demais algoritmos quanto a função de *Delay* de acesso, e o posicionamento variando as ERBs não se mostrou um método eficaz para se minimizar o workload, apresentando altas variações ao longo do gráfico.

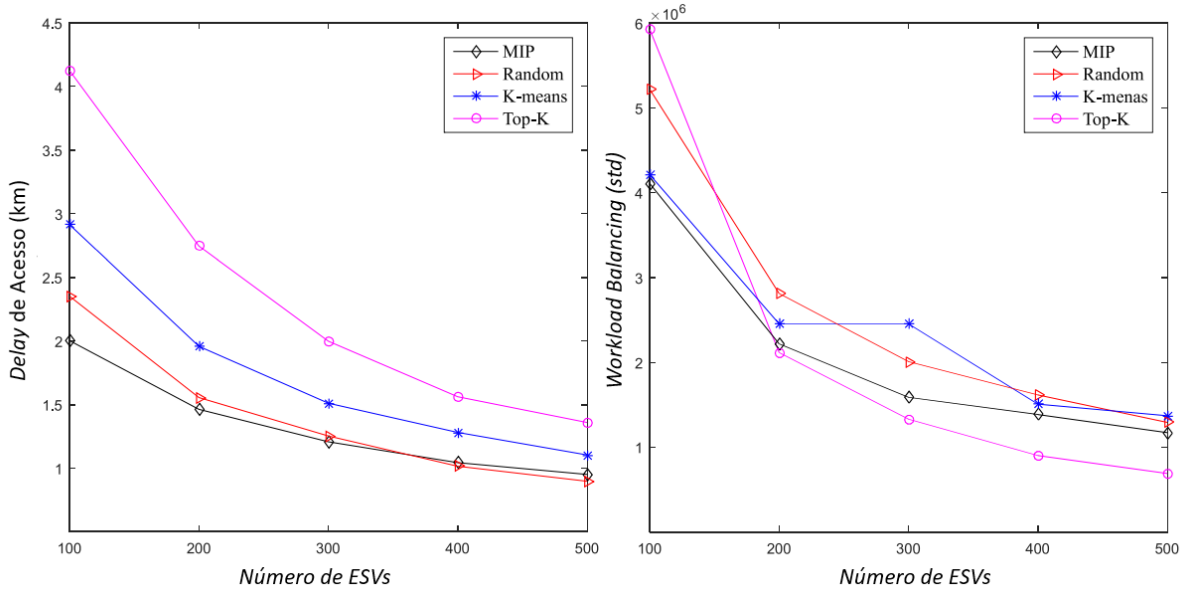
Figura 8 – Resultados da literatura utilizando o método de posicionamento de ERBs.



Fonte: Extraído de Wang et al. (2019b).

A Figura 9 apresenta os resultados do posicionamento de ERBs em Xangai variando o número de ESVs, indo de K=100 até K=500 para as 3000 ERBs do dataset dos autores. A função de *delay* de acesso converge rapidamente de 6 a 5 km de distância para valores de aproximadamente 1 km. Já a função workload é minimizada com sucesso, com o algoritmo Top-K obtendo destaque entre os outros métodos, provavelmente devido a alta demanda de usuários da cidade (26 milhões de habitantes).

Figura 9 – Resultados da literatura utilizando o método de posicionamento de ESVs.



Fonte: Extraído de Wang et al. (2019b).

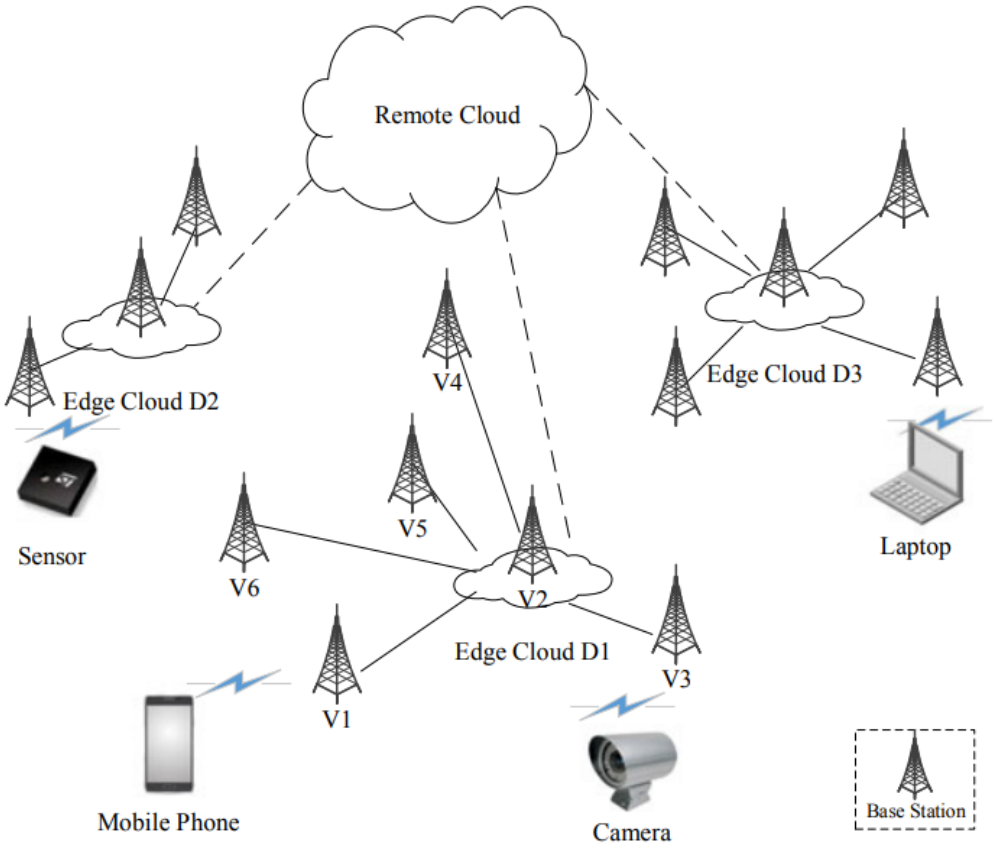
Ainda sobre o posicionamento de ESVs, Zeng et al. (2019) propõem uma modelagem de rede WMAN utilizando teoria dos grafos, com foco em minimizar o número de servidores mantendo os requerimentos de QoS definidos no problema. Após a modelagem da rede e a formulação do problema, ele é dividido em duas categorias e são propostos algoritmos para o posicionamento dessas redes, um *greedy-based* para encontrar os nós mais dominantes da rede e posicionar os clusters de forma iterativa. Já o segundo algoritmo é baseado em um processo da indústria metalúrgica: o cozimento, que leva em consideração critérios como função de custo, condição de parada, mapeamento dos nós vizinhos e temperatura.

Algoritmos de otimização também são utilizados em relação ao posicionamento de outros elementos de rede além de ESVs, Yuan, Sun e Lou (2020) abordam o problema de posicionamento de *edge nodes* virtuais, já em um contexto de redes definidas por *software* (SDN) utilizando uma aproximação com *deep learning*. Primeiramente é definida a forma como são distribuídos os usuários da rede, através de um algoritmo de *cluster tree*, que calcula a quantidade de usuários na rede em uma janela de tempo definida. Já o algoritmo que define a estratégia de posicionamento corta a *cluster tree* definida anteriormente em vários *clusters* para se obter os centroides, escolhendo as *clouds* candidatas baseada nas distâncias das mesmas para estes centroides, designando os usuários finais para estes nós virtuais levando em consideração também a estratégia de preço adotada por cada provedor de infraestrutura em nuvem a fim de se ter uma otimização da utilização destes nós a nível de preço.

3 Metodologia

Para a execução deste trabalho, primeiramente foram realizadas as etapas iniciais de escolha da localidade para estudo de caso, das funções a serem minimizadas, da aquisição e limpeza de dados. Em seguida, foram confeccionados os algoritmos em ambiente de programação Python e a performance dos modelos foi avaliada de duas maneiras: variando o número de ERBs e também variando o número de ESVs. Todas essas etapas serão detalhadas nas seções a seguir. Este trabalho foi baseado na metodologia de análise realizada com um *dataset* de ERBs da cidade de Xangai, na China (WANG et al., 2019b). Uma ilustração do modelo da aplicação deste trabalho pode ser visto na Figura 10. Neste exemplo, os autores se referem a *Remote Cloud* como a nuvem mais distante do usuário, contendo as *Core/Central* e *Metro Cloud*, e *Edge Cloud* como as localizações dos *Edge Servers* D1, D2 e D3. Os dispositivos como Laptops, Sensores, Celulares e Câmera se encontram conectados as ERBs, referidas como *Base Stations* V1, V2 e V3.

Figura 10 – Modelo de posicionamento de ESVs para *Mobile Edge Computing*.



Fonte: Extraído de Guo et al. (2020).

3.1 Estudo de caso e definição das funções a serem minimizadas

A localidade escolhida para este trabalho é o município de Vitória, capital do estado do Espírito Santo, mostrada na Figura 11. Por estar em uma ilha, Vitória possui uma distribuição mais homogênea entre as ERBs e os servidores, quando comparada a outras cidades. A cidade conta com uma população de aproximadamente 365 mil habitantes em 2020, de acordo com o IBGE (2020), e é um bom local para aplicações da Indústria 4.0 já no início desta década, por ser uma cidade portuária, turística, rota de importação e exportação de muitos países.

Figura 11 – Localidade escolhida para o estudo de caso, Vitória/ES.

Fonte: Extraído de Google Maps.

Para o estudo de caso, foram escolhidas duas funções a ser avaliadas no problema de posicionamento de ESVs: a função de *Delay* de Acesso e a função de *Workload*, que serão detalhadas a seguir. O propósito de minimizar essas funções é balancear a carga de trabalho e reduzir o delay de acesso para aplicações de baixa latência, de forma a mostrar possíveis posições iniciais destes ESVs já quando a estrutura inicial 4G começar a ser substituída pelos equipamentos 5G.

Delay de Acesso

Para a avaliação do *Delay* de Acesso entre as ERBs e os ESVs, a latência é representada como uma função direta da distância. Nestes experimentos, a distância média

entre as ERBs e os ESVs representa este *delay*.

Workload Balancing

Utiliza-se o desvio padrão (std) para avaliar o balanceamento da carga de trabalho (*workload*) entre os ESVs. Um total de K ESVs estão distribuídos entre as ERBs, o *workload* é calculado para cada ESV i como T_i, e então o desvio padrão da carga de trabalho WL pode ser determinada por (3.1).

$$WL = \sqrt{\frac{\sum_{i=1}^{K}(T_i - \bar{T})^2}{K}} \tag{3.1}$$

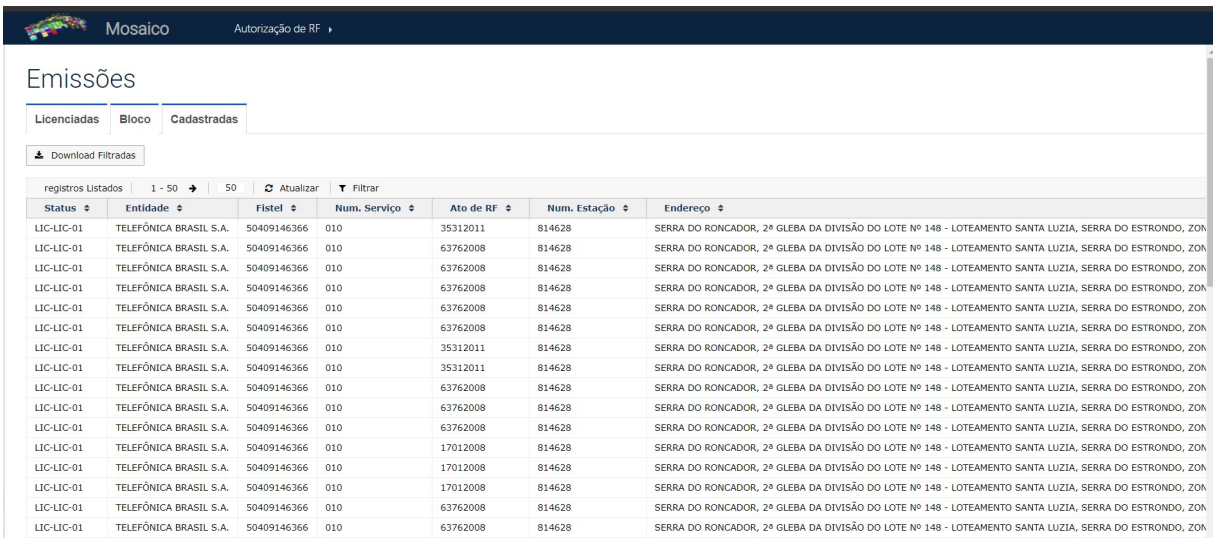
3.2 *Dataset*

Nesta seção, será feita a extração dos dados da internet, além da limpeza e do tratamento dos dados utilizando a linguagem de programação Python.

Extração

Para a aquisição dos dados das ERBs existentes na cidade, há uma preocupação em utilizar apenas dados abertos para este projeto. Portanto, foi utilizado o banco de dados de ERBs que a ANATEL disponibiliza com os endereços das ERBs por município e operadora do Brasil, conforme apresentado na Figura 12. Apesar do portal possuir algumas limitações em relação ao tempo de resposta e a quantidade de visualizações máxima por página, foram extraídos os dados das ERBs das quatro principais operadoras de telefonia que atualmente operam no país (Telefonica Vivo, NET Claro, TIM Brasil e Oi Móvel).

Figura 12 – Portal com dados das ERBs disponibilizadas pela ANATEL.



Fonte: Extraído do banco de dados abertos da ANATEL.

Limpeza e tratamento

Após realizar a limpeza dos dados separando apenas a disposição de antenas de tecnologia 4G atuais, as latitudes e longitudes foram ajustadas para uma precisão de 12 casas decimais. Para complementar a coluna de *workload*, foi simulada uma carga de trabalho fixa fictícia, com valores aleatórios de acessos condizentes com a realidade de uma ERB real de capital (50-400 Acessos por ERB). O dataset do trabalho então possui as seguintes colunas: Endereço, Latitude, Longitude e *Workload*, sendo que uma amostra desse dataset, utililzando a função *pandas.DataFrame.sample()* é mostrada na Tabela 2.

Tabela 2 – Amostra do Dataset após a limpeza de dados.

Endereço	Latitude	Longitude	Workload
Av. Dr. Pedro Feu Rosa, 501, Jardim da Penha	-20.283588	-40.301475	77
Rua João Nunes Coelho s/n, Mata da Praia	-20.278505	-40.295758	325
Rua Guadalajara, 409, Santa Cecília	-20.302352	-40.319677	395
Av. Alziro Zarur, 438, Jardim da Penha	-20.279781	-40.295876	294
Av. Fernando Ferrari, 2225, Goiabeiras	-20.266298	-40.298177	254

Fonte: Acervo pessoal.

Perfil da Carga de Trabalho

A carga de trabalho é um atributo das ERBs e dos ESVs que, para este trabalho no caso dos acessos iniciais das ERBs, foi gerada de maneira artificial, sendo esta simulada considerando uma média de acessos de 300 a 400 pessoas por ERB, sendo este valor escolhido por ser a média de acessos de uma torre 4G atualmente. Como há torres que não trabalham com a totalidade dos acessos por todo o tempo, é gerado então um vetor aleatório, entre 50 a 400 acessos por ERB e esta coluna é anexada aos dados do *dataset*. Já a carga dos ESVs é a soma total das cargas de cada ERB associada ao mesmo. Já a distribuição desta carga, *workload balancing*, é avaliado através do desvio padrão.

3.3 Algoritmos para o posicionamento dos ESV

Para o problema do posicionamento de ESVs, algumas definições iniciais serão comuns para todos os algoritmos utilizados nesse trabalho, e duas premissas são seguidas com a intenção de simplificar o problema inicial de otimização e garantir o bom o funcionamento dos algoritmos:

- Dois ESVs não compartilham a mesma ERB, ou seja, cada servidor está associado a várias ERBs, mas cada ERB está associada apenas a um ESV.

- É considerada uma homogeneidade de ESVs, que compartilham as mesmas características de poder computacional e especificações técnicas.

Seguindo a utilização de POO, tem-se as seguintes definições das classes:

- A classe *BaseStation* contém os atributos id, endereço, latitude, longitude e *workload*, recebendo os dados do *dataset* de ERBs.
- A classe *EdgeServer* possui os atributos id, latitude, longitude, id da ERB, estações base associadas e workload, recebendos os dados dos ESVs.
- A classe *ServerPlacer* contém as funções de cálculo de distância entre ESVs e ERBs, os cálculos de latência (*delay* de acesso) e *workload*, descritos em 3.1.
- As classes herdadas de *ServerPlacer* são especializações para implementações dos métodos específicos, KMeansServerPlacer, RandomServerPlacer e TopKServerPlacer.

Para a confecção dos algoritmos K-means, Top-K e Random, foi utilizada a linguagem de programação *Python*, em um ambiente criado no *Visual Studio Code*, sendo estas ferramentas de programação abertas, gratuitas e disponíveis em grande parte dos sistemas operacionais. Foram utilizados conceitos de programação orientada ao objeto (POO) para a execução dos algoritmos e *Jupyter Notebooks* para visualização e representação dos resultados.

3.4 Avaliação da performance dos modelos

A performance dos modelos é avaliada de forma a se extrair dos resultados valores mínimos de ESVs e ERBs sem afetar as especificações de *delay* e *workload* desejados para a rede móvel, diminuindo o custo dos equipamentos tanto em ESVs quanto ERBs para situações práticas.

Comparação dos resultados variando o número de ERBs

Utilizando o dataset proposto, contendo 139 ERBs, foi calculado o problema de otimização das duas funções aumentando o número de ERBs (N_{ERB}) de 40 a 139, com um passo de 1 em 1 unidade, e configurando a proporção do número de ESVs (K_{ESV}) em relação a N_{ERB} em 0,1. Ou seja, se há 40 ERBs, têm-se 4 ESVs, e se há 139 ERBs, têm-se 13 ESVs. A escolha dos ERBs é baseada na disponibilidade atual de antenas na cidade, dando uma noção inicial de posicionamento dos ESV mesmo com o entendimento que a rede irá escalar em número total de ERBs. Portanto, as iterações ocorrem de 40 ERBs (aproximadamente 30% do total) até o valor máximo de 139 ERBs (100%).

Comparação dos resultados variando o número de ESVs

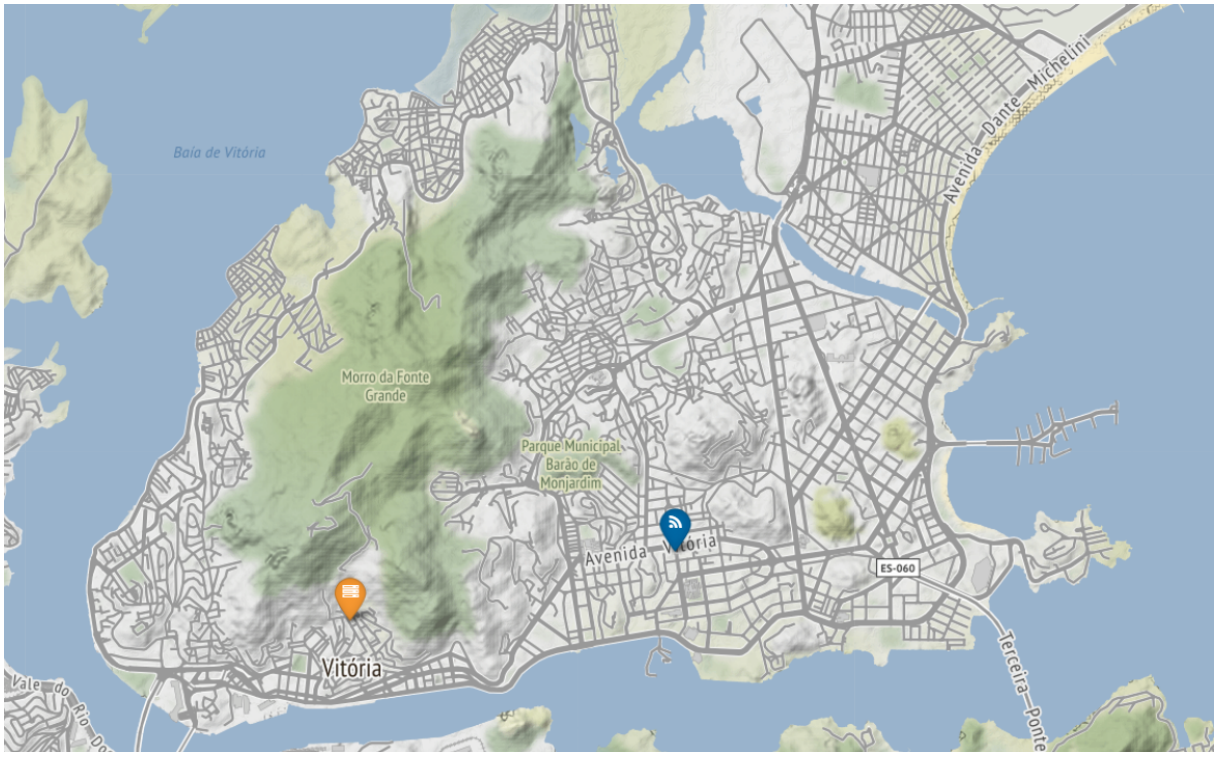
Nesta avaliação, foi fixado o número N_{ERB} totais em 139 (100 % das ERBs) e foram testados arranjos de K_{ESV} de 2 a 68, com um passo de 1 em 1, alcançando uma proporção final K/N de 0.17.

Visualização geográfica de um ponto de operação

Após a análise das curvas para o posicionamento de ESVs, criando *dataframes* com a biblioteca *Pandas*, contendo as latitudes e longitudes das ERBs e das ESVs, é possível utilizar estes dados para se obter uma visualização geográfica para um determinado ponto de operação com N_{ERB} e K_{ESV}.

A biblioteca *Folium* permite a criação da visualização do mapa utilizando a função *Folium.Map()*, e a função *Folium.Marker* é implementada para posicionar marcadores no mapa, com personalização das cores e dos ícones. Uma prévia da cidade de Vitória com um marcador de ESV em laranja, e um marcador de ERB em azul pode ser vista na Figura 13.

Figura 13 – Visualização da cidade de Vitória utilizando a função Folium.Map().



Fonte: Acervo pessoal.

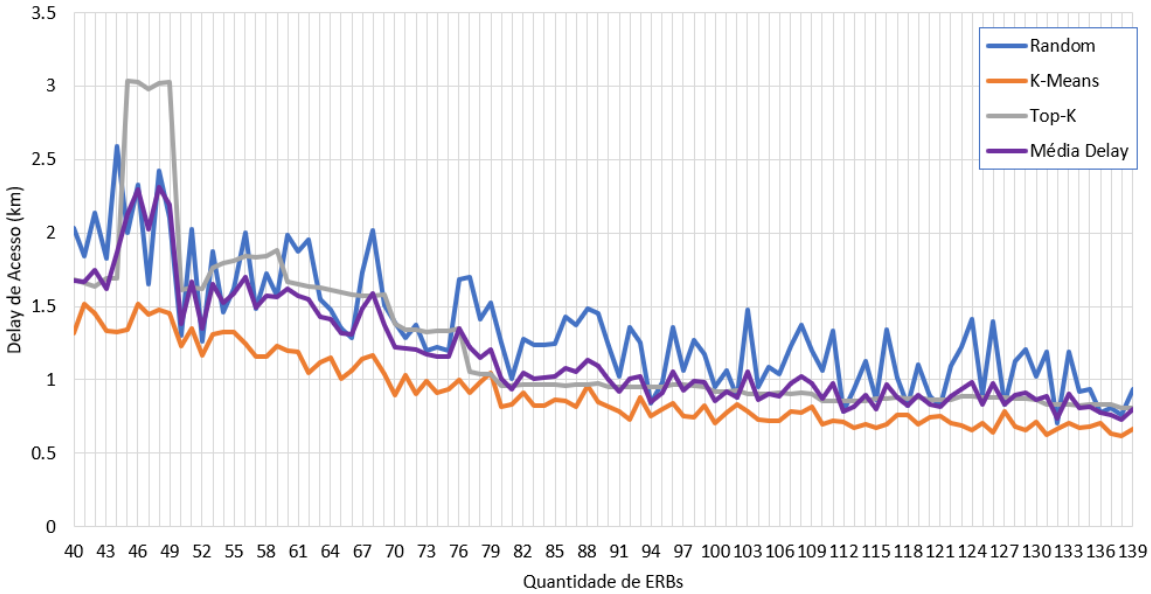
4 Resultados

Neste capítulo serão apresentados os resultados testando os três algoritmos (K-Means, Top-K e Random) para o cenário variando o número de ERBs e o cenário variando o número de ESVs. Ao final dos resultados, será gerada a visualização geográfica para o ponto ótimo de ESVs e serão comparados os gráficos com os resultados da literatura.

4.1 Comparação dos resultados variando o número de ERBs

Após testar os métodos por várias vezes, foram demonstrados as quantidades de ESVs e ERBs em função das variáveis a serem otimizadas: *delay* de acesso e *workload*. Para o problema variando N_{ERB} e mantendo fixa a proporção de ESVs em 0,1, a performance em relação ao *delay* de acesso é mostrada na Figura 14. Após a adição de todas as ERBs, independente do método utilizado, os *delays* de acesso foram minimizados, convergindo para um valor de aproximadamente 0,7 km, mostrado através da curva denominada "Média do *Delay*". O K-means apresenta um desempenho superior aos outros dois performando bem abaixo da média durante todas as iterações, alcançando valores de *delay* de até 0,6 km. Estes valores não possuem um ponto de saturação a princípio pois quanto mais ESV são posicionados, as ERBs mais próximas estarão sendo atendidas pelo ESV, porém há um *tradeoff*, impactando no aumento do custo que deve ser considerado para a aplicação de utilização da rede de cada projeto.

Figura 14 – *Delay* de Acesso variando o número de ERBs.

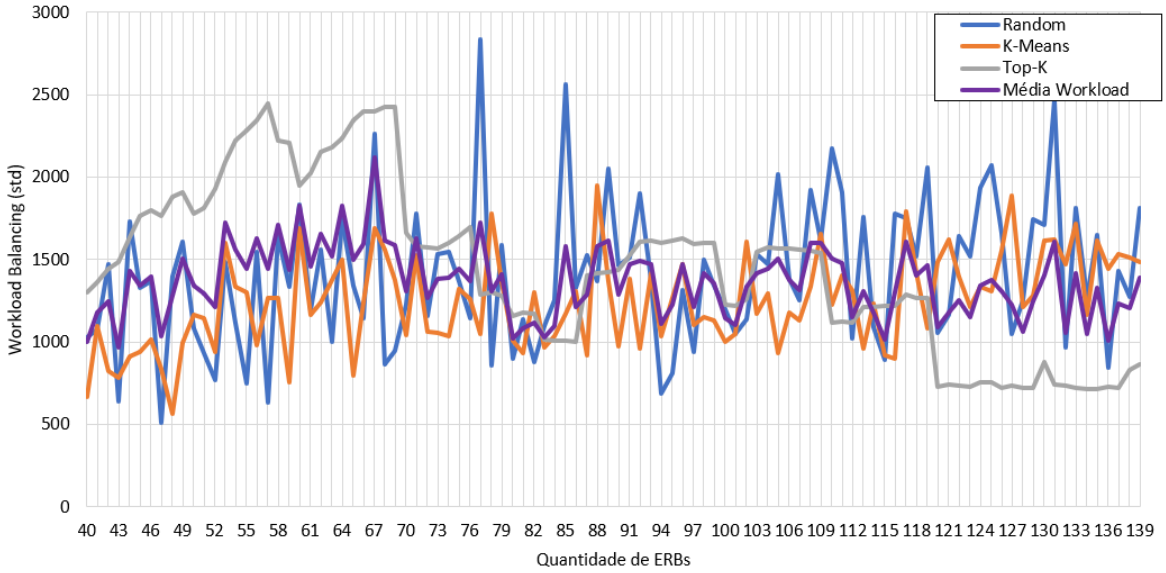


Fonte: Acervo pessoal.

O K-means se mostra superior se comparado ao Top-K, pois este foca mais no *workload* e pode até optar por escolher ESVs mais distantes em troca de atender a carga de trabalho, conforme visto nas iterações de 43 a 49 ERBs. O K-means também se mostra superior em relação ao Random, que possui uma variação aleatória, logo o algoritmo pode apresentar altas oscilações no *delay* de acesso entre as iterações, como por exemplo, no posicionamento das ERBs 64 a 70, e na 82 a 91.

No caso do problema de balanceamento de carga, a Figura 15 mostra que essa aproximação variando as ERBs não é muito efetiva para combater a minimização do *workload*, que é representado pelo desvio padrão (std) do balanceamento total. Apesar disso, após 120 ERBs o algoritmo Top-K consegue causar um equilíbrio, minimizando e estabilizando o desvio padrão em torno de 720, enquanto os demais métodos variam em torno de 1000 a 2000, os valores de média dos três métodos, representado pela curva "Média Workload", é apresentado para se ter uma avaliação de como a performance do Top-K consegue se manter bem abaixo da média enquanto os outros dois oscilam acima da média mesmo após o posicionamento de todas as 139 ERBs.

Figura 15 – *Workload balancing* variando o número de ERBs.



Fonte: Acervo Pessoal.

Fazendo um comparativo com a literatura apresentada no capítulo 2, as Figuras 14 e 15 demonstram que o comportamento das duas funções seguem o mesmo padrão do posicionamento do que as ERBs da cidade de Xangai, conforme a Figura 8, apesar e Xangai possuir 3000 ERBs, uma quantidade vinte vezes maior de ERBs posicionadas em relação a este trabalho.

A Tabela 3 apresenta o percentual de redução de cada algoritmo, para o método de posicionamento de ERBs em relação as duas funções de minimização e a literatura, além dos mínimos atingidos por cada função. O *delay* de acesso todas as funções obteram

um percentual entre -49,88% e -53,75%, com o K-means se destacando em percentual e redução do delay mínimo (0,66 km). Já para o problema do *workload* não houve uma minimização muito efetiva de nenhuma das funções, causando até aumentos perceptíveis no *delay*, como para o caso do K-means (171%), ou até nenhum aumento significativo, como o caso do K-means para os dados da literatura.

Tabela 3 – Comparação das funções de posicionamento de ERBs.

Trabalho	Vitória			Xangai		
Método	K-Means	Top-K	Random	K-Means	Top-K	Random
Variação Delay de Acesso (%)	-49,88%	-50,33%	-53,75%	-82,05%	-59,68%	-71,70%
Variação Workload Std (%)	+171,91%	+14,00%	-16,14%	∼0%	-2,10%	-9,09%
Delay minimo atingido (km)	0,66	0,83	0,93	0,7	2,5	1,5
Workload minimo atingido (std)	1814,3	1481,7	860,6	1,85e6	1,4e6	0,9e6

Fonte: Arquivo Pessoal.

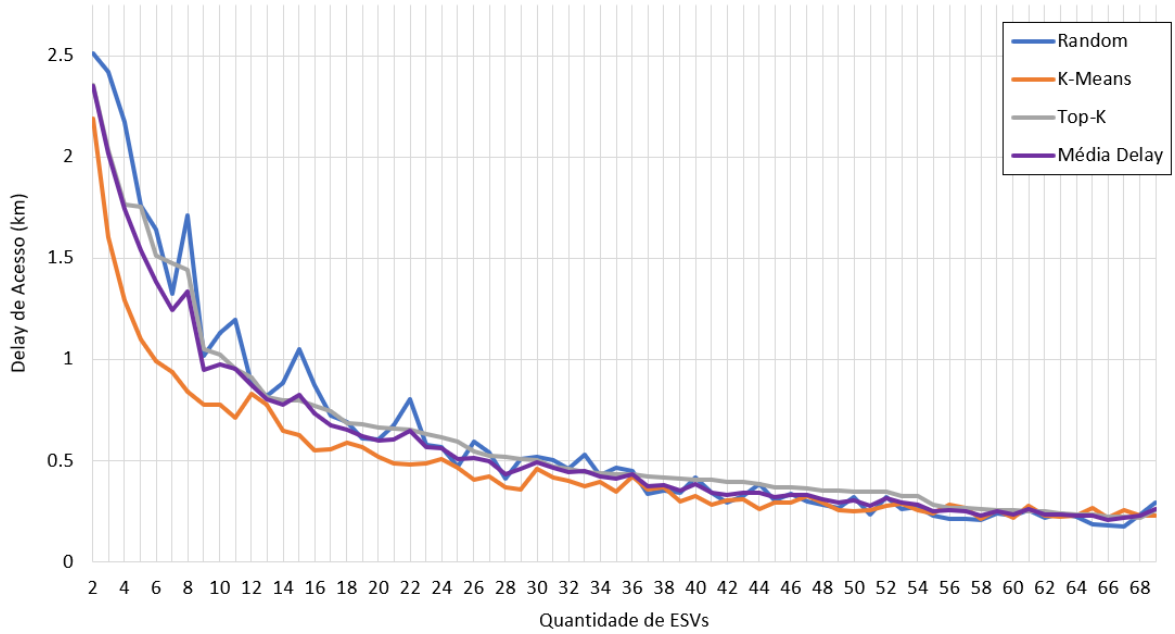
4.2 Comparação dos resultados variando o número de ESVs

Para o problema variando o número de ESVs e mantendo fixo N_{ERB} em 139 (que é o total de ERBs disponíveis para serem atendidas), a Figura 16 apresenta as curvas de cada algoritmo conforme K_{ESV} cresce. Iniciando as iterações com $K_{ESV} = 2$ e aumentando de 1 em 1 unidade até a metade do total de ERBs ($K_{ESV} = 68$). O valor de ESVs é extrapolado de propósito, para se ter uma função mais acentuada e encontrar um ponto ótimo próximo do joelho desta curva onde serão posicionados os servidores.

É analisado então o trade-off entre *delay* de acesso e quantidade de ESVs, afinal, em um cenário prático não se teria um ESV para atender apenas duas ERBs devido ao altos custo da solução. Observa-se que o K-Means apresenta um desempenho melhor no início do algoritmo, porém conforme mais ESVs são adicionados, o *delay* de acesso decresce para todos os algoritmos, pois as probabilidades de se escolher um ESV mais próximo nas iterações seguintes aumentam, minimizando e otimizando o *delay* de acesso de forma geral.

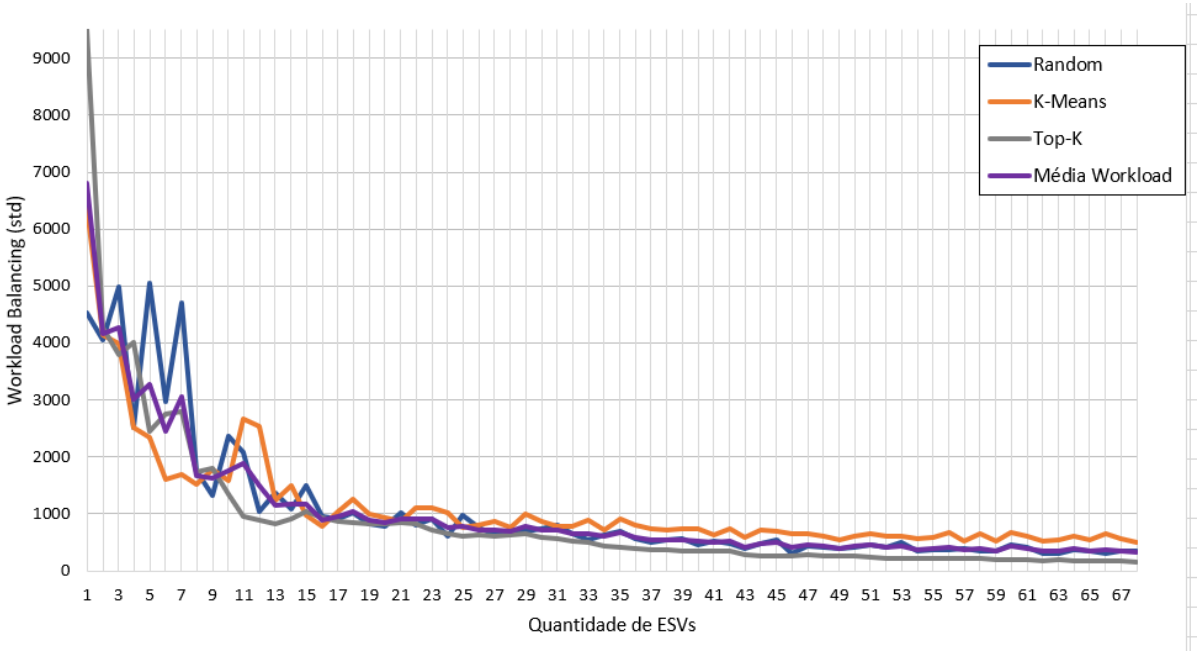
Para o problema balanceamento da carga, a Figura 17 mostra que o resultado do algoritmo K-Means inicialmente consegue ser superior aos outros dois algoritmos, porém o Top-K após 10 a 11 ESVs consegue ultrapassar em desempenho por atacar diretamente as ERBs de maior demanda, e todos os métodos conseguem convergir para um *workload* médio de 400 e equilibrar os ESVs. Um fato interessante a ser observado é que independente do método, após um certo número de ESVs adicionados, a curva estabiliza, e mesmo que o número de ESVs posicionados aumente, o ganho em minimização das funções não é tão alto quanto nas primeiras iterações, uma característica bem comum em *clusters*.

Figura 16 – *Delay* de Acesso variando o número de ESVs.



Fonte: Acervo pessoal.

Figura 17 – *Workload balancing* variando o número de ESVs.



Fonte: Acervo pessoal.

Em comparação com a literatura apresentada, o comportamento das funções nas Figuras 16 e 17 também seguem o mesmo padrão da Figura 9. A Tabela 4 mostra os valores de minimização de cada função para o método de posicionamento de ESVs, em contraste com os resultados da literatura apresentados no capítulo 2. A variação do *delay* de acesso mostra que todas as funções foram bem sucedidas na minimização, com destaque para K-means, com uma redução de -89,67% e um *delay* mínimo atingido de 0,23. Os três

algoritmos também obtiveram sucesso em minimizar a função de *workload*, com destaque para o Top-K, que obteve reduções percentuais na literatura de -88,33%, e percentuais de redução ainda superiores neste trabalho (-98.35%).

Tabela 4 – Comparação das funções de posicionamento de ESVs.

Trabalho	Vitória			Xangai		
Método	K-Means	Top-K	Random	K-Means	Top-K	Random
Variação Delay de Acesso (%)	-89,67%	-89,11%	-88,31%	-58,62%	-65,85%	-62,50%
Variação Workload Std (%)	-92,34%	-98,35%	-92,48%	-68,29%	-88,33%	-42%
Delay minimo atingido (km)	0,23	0,25	0,29	1,2	1,4	1,0
Workload minimo atingido (std)	490	156	339	1,3e6	0,7e6	2,9e6

Fonte: Arquivo Pessoal.

4.3 Visualização geográfica do local utilizando um ponto de operação

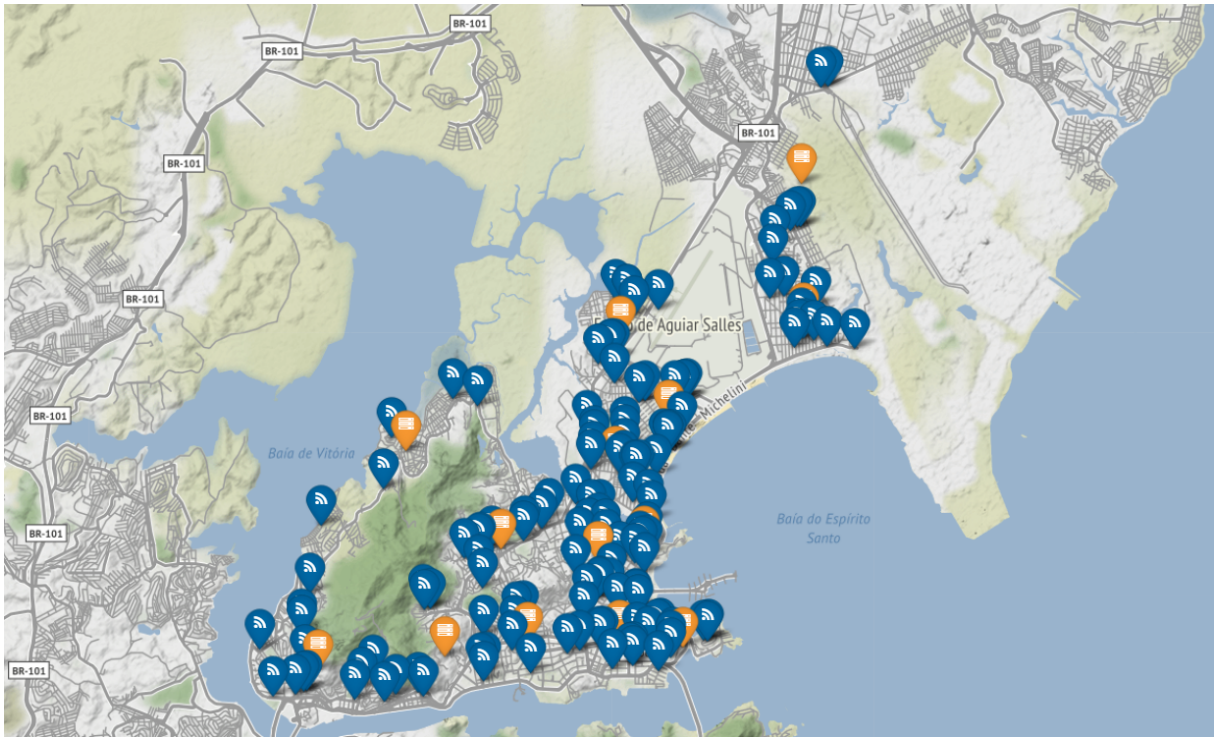
A partir das Figuras 16 e 17, é possível observar que 14 servidores é um valor aceitável para atender os requisitos das duas funções, pois o *delay* de acesso e o *workload* já se encontram no joelho da curva.

Considerando que a função de *delay* é a função mais importante a ser minimizada neste trabalho, afinal, é a redução deste atributo que irá possibilitar que as aplicações de baixa latência se tornem possíveis no futuro, o K-means é o algoritmo escolhido para o ponto de operação já que obteve o maior percentual de redução de *delay*. A Figura 18 apresenta os 14 servidores, representados em azul escuro e as 139 ERBs em laranja para o método K-means. Vale notar que são localizações sugeridas, podendo ser posicionados os ESVs nas localizações de ERBs mais próximas do centroide se for conveniente para o projeto.

Uma outra possibilidade é o foco no balanceamento da carga de trabalho, analisando a Figura 17, é escolhido o ponto de operação utilizando o Top-K com 12 servidores, com um balanceamento já mais para o final do joelho da curva. O posicionamento dos elementos de rede é apresentado na Figura 19.

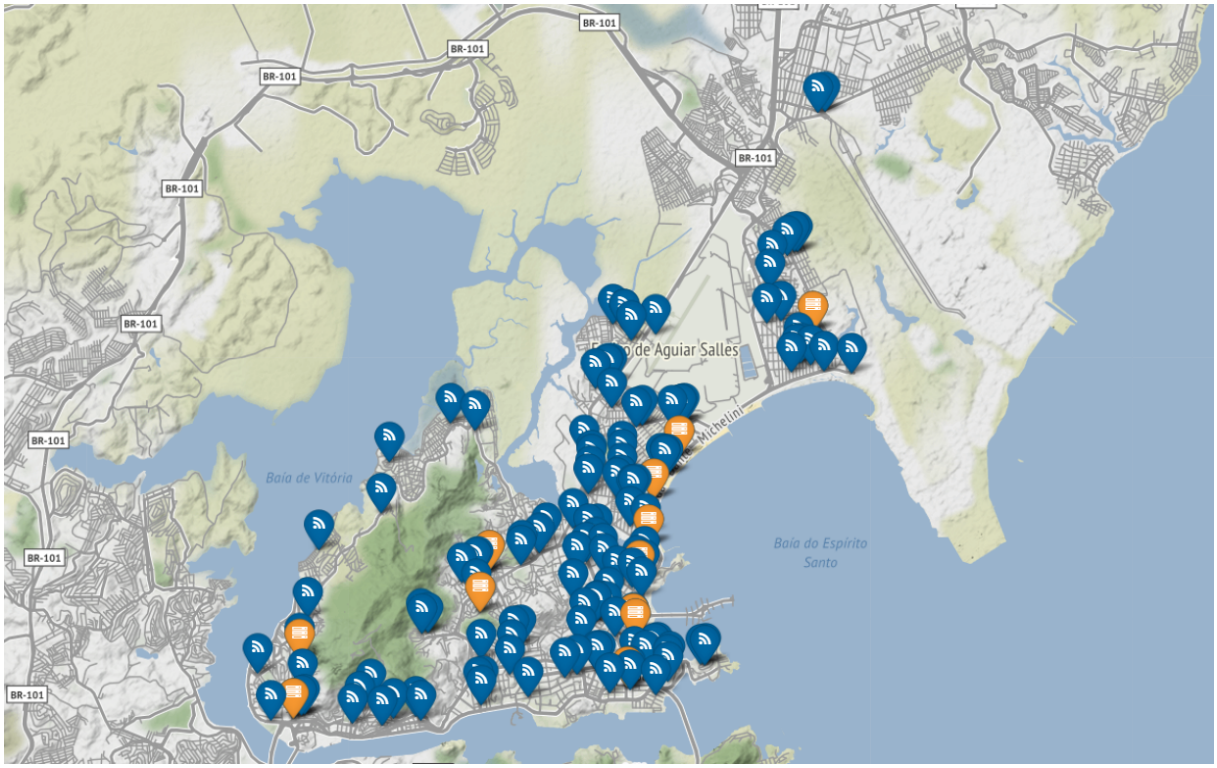
Ainda sobre os mapas apresentados, escolhendo o local correto dos servidores e atendendo a demanda dos usuários, é possível acelerar aplicações práticas como: carros autônomos completamente integrados conversando uns com os outros no trânsito, aplicações da saúde (telemedicina, cirurgias remotas), eventos públicos imersivos (estádios inteligentes), e *smart grids*, melhorando a distribuição e balanceamento de energia elétrica pela cidade.

Figura 18 – Distribuição de 14 edge servers pela cidade de Vitória utilizando K-Means.



Fonte: Acervo pessoal.

Figura 19 – Distribuição de 12 edge servers pela cidade de Vitória utilizando Top-K.



Fonte: Acervo pessoal.

5 Conclusão e Trabalhos Futuros

Edge Computing e suas aplicações, como a proposta deste trabalho, podem ser utilizadas para atender as demandas que remetem ao 5G e a Indústria 4.0, ampliando a capacidade de computação e armazenamento, além de reduzir a latência da rede móvel para os dispositivos móveis e melhorar o *offloading* correspondente à crescente demanda de usuários. A contribuição deste trabalho tem o objetivo de mostrar que o posicionamento inteligente dos servidores pode ser um fator relevante para redução de custos e aceleração da implementação destas tecnologias. Primeiro foi formulado o problema de colocação dos *edge servers*, com as funções de minimização sendo o *delay* de acesso e o balanceamento da carga. Em seguida, foi criado um ambiente de programação para simular três métodos de posicionamento: Top-K, Random e K-Means. E finalmente aplicando estes métodos em um *dataset* real na cidade de Vitória, avaliando as performances dos modelos comparando as abordagens.

Os resultados para o método de variação de ERBs minimizaram o *delay* de acesso, porém não apresentaram reduções significativas quanto ao *workload*. Já para o método de posicionamento variando os ESVs, o K-Means obteve um desempenho ligeiramente melhor que os demais métodos em relação a redução do *delay* de acesso. O Top-K obteve destaque em relação a redução do *workload*. Após a análise dos resultados foi selecionado um ponto ótimo de 14 ESVs, que atenda os requisitos das duas funções e as posições geográficas dos ESVs e das ERBs puderam ser visualizadas em um mapa.

Para aplicações a nível de *Smart Cities*, além de determinar uma quantidade otimizada de ESVs para cada aplicação, alguns fatores devem ser considerados, como a quantidade de fundos de investimentos disponíveis para aplicação em infraestrutura de rede móvel, pois um número maior de ESVs corresponde a um custo maior. Também é necessário determinar o melhor *delay* de acesso dentro de um *range* aceitável para cada aplicação.

Como sugestão de trabalhos futuros, é sugerida a implementação para servidores não-homogêneos, considerando a capacidade individual de cada um, sendo necessária uma avaliação mais aprofundada dos recursos da cidade. Outra sugestão é considerar a utilização de outras bibliotecas de *clustering* e avaliar a performance dos modelos em outros *datasets*, com foco em cidades e regiões maiores.

Referências

5GLEARNING.ORG. *Evolução das redes móveis ao longo das gerações*. 2021. Disponível em: <https://5glearning.org/>. Citado na página 12.

AHMAD, S. et al. A review on edge to cloud: Paradigm shift from large data centers to small centers of data everywhere. In: IEEE. *2020 International Conference on Inventive Computation Technologies (ICICT)*. [S.l.], 2020. p. 318–322. Citado na página 21.

ARTHUR, D.; VASSILVITSKII, S. *k-means++: The advantages of careful seeding*. [S.l.], 2006. Citado na página 22.

CUI, G. et al. Robustness-oriented k edge server placement. In: IEEE. *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. [S.l.], 2020. p. 81–90. Citado na página 18.

CUI, Z.; WANG, J. *Enhanced software-defined network controller to support ad-hoc radio access networks*. [S.l.]: Google Patents, 2020. US Patent 10,609,590. Citado na página 13.

FOLHA. *Netflix ultrapassa marca de 200 milhões de assinantes*. 2021. Disponível em: <https://www1.folha.uol.com.br/mercado/2021/01/netflix-ultrapassa-marca-de-200-milhoes-de-assinantes.shtml>. Citado na página 15.

GARTNER. *Guide to Distributed Cloud*. 2021. Disponível em: <https://www.gartner.com/smarterwithgartner/the-cios-guide-to-distributed-cloud/>. Citado na página 14.

GHOSH, A. et al. 5g evolution: A view on 5g cellular technology beyond 3gpp release 15. *IEEE Access*, IEEE, v. 7, p. 127639–127651, 2019. Citado 2 vezes nas páginas 13 e 14.

GUO, Y. et al. User allocation-aware edge cloud placement in mobile edge computing. *Software: Practice and Experience*, Wiley Online Library, v. 50, n. 5, p. 489–502, 2020. Citado 2 vezes nas páginas 17 e 25.

GUPTA, M. K.; JAIN, A.; AMGOTH, T. Qos-aware virtual machine placement for infrastructure cloud. In: IEEE. *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*. [S.l.], 2018. p. 353–357. Citado na página 18.

GUSEV, M.; DUSTDAR, S. Going back to the roots—the evolution of edge computing, an iot perspective. *IEEE internet Computing*, IEEE, v. 22, n. 2, p. 5–15, 2018. Citado na página 14.

HASSAN, N.; YAU, K.-L. A.; WU, C. Edge computing in 5g: A review. *IEEE Access*, IEEE, v. 7, p. 127276–127289, 2019. Citado na página 15.

HOLMA, H.; TOSKALA, A.; NAKAMURA, T. *5G technology: 3GPP new radio*. [S.l.]: John Wiley & Sons, 2020. Citado na página 20.

IBM. *IBM Cloud Learning hub What is Distributed Cloud*. 2020. Disponível em: <https://www.ibm.com/cloud/learn/distributed-cloud#toc-what-is-di-VlIdBUls>. Citado na página 14.

JIA, M.; CAO, J.; LIANG, W. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Transactions on Cloud Computing*, IEEE, v. 5, n. 4, p. 725–737, 2015. Citado na página 18.

LEE, S.; LEE, S.; SHIN, M.-K. Low cost mec server placement and association in 5g networks. In: IEEE. *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. [S.l.], 2019. p. 879–882. Citado na página 17.

LI, B. et al. Placement of edge server based on task overhead in mobile edge computing environment. *Transactions on Emerging Telecommunications Technologies*, Wiley Online Library, p. e4196, 2020. Citado na página 18.

LIM, T. *K-means Visualization*. 2018. Disponível em: <https://theanlim.rbind.io/post/clustering-k-means-k-means-and-gganimate/>. Citado na página 22.

LU, D. et al. Robust server placement for edge computing. In: IEEE. *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. [S.l.], 2020. p. 285–294. Citado na página 18.

MANNWEILER, C. et al. Evolution of mobile communication networks. *Towards Cognitive Autonomous Networks: Network Management Automation for 5G and Beyond*, Wiley Online Library, p. 29–92, 2020. Citado na página 12.

NOKIA. *Edge Server*. 2020. Disponível em: <https://onestore.nokia.com/asset/210343?_ga=2.53887339.2104470510.1627841165-368745910.1627841165>. Citado na página 21.

PADRIñáN, M. A. *Torre De Sinalização*. 2017. Disponível em: <https://www.pexels.com/pt-br/foto/torre-de-sinalizacao-579471/>. Citado na página 20.

SANTOYO-GONZÁLEZ, A.; CERVELLÓ-PASTOR, C. Edge nodes infrastructure placement parameters for 5g networks. In: IEEE. *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. [S.l.], 2018. p. 1–6. Citado na página 19.

SHUJA, J. et al. Analysis of vector code offloading framework in heterogeneous cloud and edge architectures. *IEEE Access*, IEEE, v. 5, p. 24542–24554, 2017. Citado na página 14.

WAGSTAFF, K. et al. Constrained k-means clustering with background knowledge. In: *Icml*. [S.l.: s.n.], 2001. v. 1, p. 577–584. Citado na página 21.

WANG, S. et al. Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach. *IEEE Transactions on Mobile Computing*, IEEE, 2019. Citado na página 18.

WANG, S. et al. Edge server placement in mobile edge computing. *Journal of Parallel and Distributed Computing*, Elsevier, v. 127, p. 160–168, 2019. Citado 4 vezes nas páginas 17, 23, 24 e 25.

WANG, Y. et al. A reinforcement learning approach for online service tree placement in edge computing. In: IEEE. *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. [S.l.], 2019. p. 1–6. Citado na página 19.

WELDON, M. K. *The future X network: a Bell Labs perspective*. [S.l.]: CRC press, 2016. Citado 3 vezes nas páginas 15, 16 e 17.

XIAO, K. et al. A heuristic algorithm based on resource requirements forecasting for server placement in edge computing. In: IEEE. *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. [S.l.], 2018. p. 354–355. Citado na página 18.

XU, Z. et al. Efficient algorithms for capacitated cloudlet placements. *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 27, n. 10, p. 2866–2880, 2015. Citado na página 18.

YORK, S.; POYNTER, R. Global mobile market research in 2017. In: *Mobile Research*. [S.l.]: Springer, 2018. p. 1–14. Citado na página 13.

YUAN, X.; SUN, M.; LOU, W. A dynamic deep-learning-based virtual edge node placement scheme for edge cloud systems in mobile environment. *IEEE Transactions on Cloud Computing*, IEEE, 2020. Citado 2 vezes nas páginas 18 e 24.

ZENG, F. et al. Cost-effective edge server placement in wireless metropolitan area networks. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 19, n. 1, p. 32, 2019. Citado 2 vezes nas páginas 18 e 24.

Glossário

Cloud assisted driving: Sistema de direção autônomo que permite que o veículo permaneça dentro das faixas, evite colisões, faça ajustes na velocidade e na direção, além de atualizar informações de tráfego e mapeamento, conversando com outros veículos autônomos através da nuvem.

Cloud computing: Computação em nuvem, disponibilidade sob demanda de recursos, como dados e programas de um servidor remoto localizado geograficamente em qualquer lugar.

Cloud gaming: Jogos sob demanda utilizando computação em nuvem, utilizando servidores remotos de data centers sem a necessidade de instalar o jogo na máquina do usuário.

Cloudlet: *Data center* de pequena escala.

Core/Central Cloud: Servidor de alta capacidade de disponibilidade que cobre uma vasta área, com todas as funções centralizadas, caracterizada por uma alta latência (20-100+ ms).

Data center: Local destinado a centralização a rede e serviços de tecnologia, onde são processados todos os dados e informações de uma empresa.

Dataset: Conjunto de dados, coleção de dados representados de forma tabular.

Deep learning: Aprendizagem profunda, ramo do aprendizado de máquina na área de inteligência artificial.

Delay: Atraso antes que uma transmissão de dados comece.

Distributed Cloud: Nuvem distribuída, modelo de rede que estende os serviços de nuvem para próximo do cliente final.

Edge Cloud ou Edge Computing: Computação de dados na borda da rede, junto com os dispositivos (*hardware* e *software*) na localização física, aproximando a computação e o armazenamento de dados economizando o tempo de resposta.

Edge server: Servidor de borda mais próximo do usuário final.

Enhanced Mobile Broadband (eMBB): Ultra banda larga para conectividade sem fio, propriedade do 5G que permite velocidades acima de 1 Gbps.

Feature: Recurso, característica ou propriedade individual mensurável de um fenômeno.

Infrastructure as a Service (IaaS): Serviço de computação em nuvem que

oferece recursos de computação sob demanda pagos conforme o uso.

Internet of Things (IoT): Internet das coisas, interconexão digital e comunicação de dados de objetos cotidianos.

K-means: Método de agrupamento de dados em torno de centróides.

Machine learning: Aprendizagem de máquina.

Massive machine type communications (mMTC): Comunicações maciças do tipo máquina, característica do 5G, que incorpora uma quantidade massiva de dispositivos (IoT).

Massive MIMO: *Multiple Input, Multiple Output*, utiliza múltiplas antenas em uma matriz para acelerar a troca de dados na rede, podendo chegar até dezenas de arranjos no 5G.

Metro Cloud: Região da nuvem intermediária entre a *Edge Cloud* e a *Central Cloud*, com latências médias e localizações mais próximas do que a *Central Cloud*.

Network Slicing: Serviço que permite criar redes personalizadas para serviços críticos, como se houvesse uma "fatia" específica da rede.

Pay-as-you-go: Sistema onde a pessoa ou organização paga pelo serviço em tempo real.

Platform-as-a-service (PaaS): Ambiente de desenvolvimento e implantação totalmente hospedado na nuvem, utilizado sob demanda.

Quality of Service (QoS): Conjunto de tecnologias e especificações que asseguram tráfego de alta prioridade e serviços através de filas de prioridade, garantindo uma melhor confiabilidade e uso da rede.

Reinforcement learning: Aprendizagem por reforço, área do aprendizado de máquina.

Software-as-a-service (SaaS): Disponibilização de softwares como serviço sob demanda hospedado na nuvem, sem necessidade de hardware ou software local.

Streaming: Forma de distribuição digital, transmitindo dados diretamente pela internet sem a necessidade de baixar o conteúdo.

Transceiver: Transceptor, composto por um transmissor e um receptor, transmitindo e recebendo dados em um dispositivo apenas.

Ultra-Reliable Low Latency Communication (URLLC): um dos pilares do 5G, provendo serviços de confiabilidade da rede de 99,99% e latências de 1 ms na transmissão de pacotes.

Video on-demand: Vídeo sob demanda, podendo ser acessado através de uma

página da internet.

Web services: Solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

Workload: Carga de trabalho.