

**UNIVERSIDADE DO VALE DO RIO DOS SINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

DANIEL LOPES FERREIRA

**DASTDATA: UM MODELO BASEADO EM FOG E CLOUD COMPUTING PARA
RASTREABILIDADE E ARMAZENAMENTO DISTRIBUÍDO DE DADOS IOT EM
SMART CITIES**

São Leopoldo
2022

DANIEL LOPES FERREIRA

**DASTDATA: UM MODELO BASEADO EM FOG E CLOUD COMPUTING PARA
RASTREABILIDADE E ARMAZENAMENTO DISTRIBUÍDO DE DADOS IOT EM
SMART CITIES**

Artigo apresentado como requisito parcial para
obtenção do título de Bacharel em Ciência da
Computação, pelo Curso de Ciência da Compu-
tação da Universidade do Vale do Rio dos Sinos
(UNISINOS)

Orientador(a): Prof. Dr. Rodrigo da Rosa Righi

São Leopoldo
2022

DASTDATA: UM MODELO BASEADO EM FOG E CLOUD COMPUTING PARA RASTREABILIDADE E ARMAZENAMENTO DISTRIBUÍDO DE DADOS IOT EM SMART CITIES

Daniel Lopes Ferreira¹

Rodrigo da Rosa Righi²

Resumo: Neste trabalho é apresentada uma solução para o armazenamento distribuído de dados em Cidades Inteligentes. Através de uma arquitetura Edge-Fog-Cloud que particiona os dados por meio da técnica de Sharding é proposto um modelo hierárquico que manipula dados IoT gerados por Smart Cities. O problema está relacionado com as abordagens utilizadas para conseguir promover um ambiente integrado. Os trabalhos relacionados costumam utilizar abordagens concentradas em Cloud que geram altas taxas de latência e os que utilizam Fog Computing, apenas usam a camada como um *middleware*, não explorando maiores possibilidades de aproveitamento. Nesse contexto, este trabalho apresenta o modelo DASTData que visa possibilitar menores taxas de latência, mais segurança para os dados, tolerância a falhas, alta disponibilidade e consultas concorrentes para promover uma melhor experiência na gestão e disponibilidade de dados em cidades inteligentes. Além disso, nossa contribuição para a literatura, diferentemente dos trabalhos relacionados, é relativa a proposição de uma arquitetura focada em possibilitar a rastreabilidade dos usuários que têm um comportamento móvel na cidade, proporcionando a capacidade de analisar padrões e ocorrências através da consolidação de dados de um ou mais indivíduos. Nos resultados obtidos através dos testes realizados neste trabalho, observa-se que, em consultas, um modelo descentralizado como o DASTData é até 73% mais eficiente que um modelo centralizado.

Palavras-chave: Armazenamento Distribuído. Rastreabilidade. Fragmentação de Dados. Fog Computing. Cloud Computing.

Abstract: In this work, a solution for the distributed storage of data in Smart Cities is presented. Through an Edge-Fog-Cloud architecture that partitions the data through the Sharding technique, a hierarchical model is proposed that manipulates IoT data generated by Smart Cities. The problem is related to the approaches used to promote an integrated environment. Related works tend to use cloud-focused approaches that generate high latency rates, and those that use Fog Computing only use the layer as *middleware*, not exploring greater possibilities for use. In this context, this work presents the DASTData model that aims to enable lower latency rates, more data security, fault tolerance, high availability and concurrent queries to promote a better experience in data management and availability in smart cities. In addition, our contribution to the literature, unlike related works, is related to the proposition of an architecture focused on enabling the traceability of users who have a mobile behavior in the city, providing the ability to analyze patterns and occurrences through the consolidation of data from one or more individuals. In the results obtained through the tests carried out in this work, it is observed that in queries a decentralized model such as DASTData is up to 73% more efficient than a centralized model.

¹Graduando em Ciência da Computação pela Unisinos. Email: danielferreira17@edu.unisinos.br

²Professor e pesquisador no PPG em Computação Aplicada da Unisinos. Email: rrrighi@unisinos.br

Keywords: Distributed Storage. Data Traceability. Data Sharding. Fog Computing. Cloud Computing.

1 INTRODUÇÃO

Com o crescimento da população urbana nos grandes centros as *Smart Cities* tornam-se uma realidade cada vez mais próxima. A possibilidade de integração entre as diversas áreas de uma cidade trazem a expectativa de melhorias na gestão pública em todas as suas atribuições como mobilidade, saúde, educação e infraestrutura (LAI et al., 2020). Integrar os dados gerados também é um desejo dos próprios usuários que podem manter um controle ativo da sua saúde, sua agenda e sua casa a fim de melhorar sua produtividade, ter mais tempo e qualidade de vida. Contudo, as tarefas de armazenamento e consulta desses dados precisam ser o mais performáticas e transparentes possíveis.

A gestão dos dados é o fator chave para o sucesso de um ecossistema inteligente (BENHAMIDA et al., 2022). Isso se deve ao fato de que as cidades inteligentes, a partir de seus sensores, geram uma enorme quantidade de dados (SASUBILLI; KUMAR; DUTT, 2020) que necessitam ser armazenados de forma eficiente a fim de garantir melhor performance no processamento e análise desses dados. Neste contexto, nota-se que lidar com dados na borda da rede reduz a latência, aumenta a segurança e diminui a carga na rede quando comparado há um modelo distante como o baseado apenas em Cloud Computing (VILELA et al., 2020). Uma maneira de implementar essa manipulação mais próxima dos dados é através de clusters (ou nós) de Fog Computing (NAEEM et al., 2019) que distribuem-se na cidade e são mais eficientes nos tratamentos de dados.

Ademais, os Sistemas de Gerenciamento de Bancos de Dados Relacionais (RDBMS) têm muita dificuldade ou agregam mais complexidade neste contexto de aplicação que necessita de maior flexibilidade. Já os Bancos de Dados Não Relacionais (NoSQL) têm por característica nativa a flexibilidade no esquema dos dados, bem como, já nasceram para serem facilmente distribuídos e escalados a partir de técnicas como Sharding (ABDELHAFIZ; ELHADEF, 2021) que particiona os dados horizontalmente entre diversas instâncias de banco de dados.

Este trabalho está vinculado ao projeto Minha História Digital e objetiva, principalmente, a proposição e implementação de uma arquitetura de armazenamento em nós hierárquicos no modelo Edge-Fog-Cloud para cidades inteligentes com foco no armazenamento e rastreabilidade de dados vitais gerados através de sensores IoT. Essa arquitetura deverá promover a proximidade geográfica aos clientes, diminuição da latência, aumento da segurança, diminuição do fluxo de dados trafegados, tolerância a falhas, consultas concorrentes e fácil escalabilidade de recursos. Algumas proposições de arquiteturas foram analisadas para embasar os conceitos e abordagens já estudados, buscando encontrar possíveis lacunas na literatura.

Após a análise dos trabalhos relacionados (KUDO, 2018; SINAEEPOURFARD et al., 2018; SHWE; JET; CHONG, 2016; ABREU et al., 2017; LOMOTEY; PRY; CHAI, 2018; ZHANG,

2020; BENHAMIDA et al., 2022) verificou-se a necessidade de se propor uma arquitetura que atenda a todos os requisitos já citados buscando preencher lacunas na literatura relacionadas a capacidade de rastreabilidade dos dados de usuários dispersos em arquiteturas distribuídas e a diminuição de latência de conexão entre serviços e dispositivos IoT. Nesse contexto, este artigo apresenta o modelo DASTData, acrônimo para *Distributed Architecture to Store and Trace Data*, que aborda principalmente como armazenar e rastrear dados em arquiteturas complexas e distribuídas como as de Smart Cities através do uso conjunto de tecnologias como Sharding e conceitos como Fog e Cloud Computing.

Com as proporções que uma arquitetura de uma cidade inteligente pode tomar e observando os problemas de armazenamento distribuído que precisam ser solucionados neste contexto, o presente trabalho tem como objetivo geral desenvolver um modelo de armazenamento e rastreabilidade de dados para smart cities. Para isto destacam-se os seguintes objetivos específicos:

- a) identificar e analisar o formato e topologia ideais para a distribuição dos nós da arquitetura;
- b) disponibilizar uma solução eficiente para o problema da rastreabilidade dos usuários móveis na rede;
- c) propor uma arquitetura robusta quanto a tolerância a falhas, performance e disponibilidade dos dados;
- d) analisar e propor uma solução que resolva o problema da alta latência de um modelo baseado apenas em Cloud.
- e) permitir o armazenamento distribuído dos dados para possibilitar consultas concorrentes.

Este trabalho está disposto em sete seções. Sendo que na Seção 1, contextualizam-se o problema e as justificativas do trabalho, bem como são apresentados os objetivos da presente pesquisa. Na Seção 2 são definidos os conceitos que serão relevantes para a proposição do modelo. Já na Seção 3, exponho a metodologia da seleção dos trabalhos que se relacionam diretamente com esta pesquisa, também comento detalhadamente aspectos relevantes de cada um a fim de verificar lacunas nas proposições.

Na Seção 4, cito as decisões de projeto tomadas à luz dos pontos positivos e negativos dos trabalhos relacionados, e também defino e apresento a arquitetura proposta para resolver os problemas verificados. Seguindo para a Seção 5, discorro sobre como foi realizada a definição da metodologia de avaliação do modelo para provar sua eficiência e finalmente na Seção 6 apresento alguns resultados preliminares obtidos através da implantação do protótipo. Por fim na Seção 7, trago uma síntese do que foi exposto na Seção 4, também trato das contribuições esperadas com a arquitetura proposta e comento os trabalhos futuros.

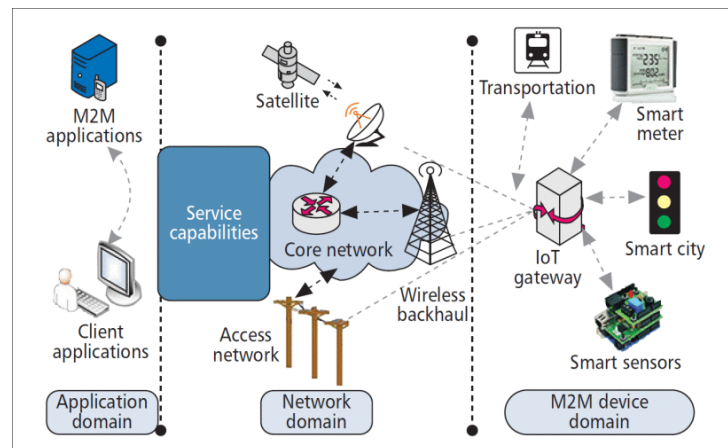
2 FUNDAMENTAÇÃO TEÓRICA

Esta seção aborda os conceitos citados por este trabalho a fim de esclarecer e embasar a pesquisa desenvolvida. A partir da descrição de Computação em Nuvem, Computação em Nebulosa, Internet das Coisas, Cidades Inteligentes e Fragmentação de Dados será possível avançar na pesquisa mantendo os conceitos atualizados com estudos anteriores que definem estes assuntos.

2.1 Internet das Coisas

Internet das Coisas (ou *Internet of Things* - IoT) é um conceito diretamente relacionado a interação – transparente – entre os dispositivos do nosso dia a dia em um modelo M2M³ (PATEL; PATEL et al., 2016). Cada um desses dispositivos está localizado em um nível chamado *Edge* ou borda da rede, pois estão muito próximos dos usuários finais. Eles geram uma enorme quantidade de dados a todo momento através de pequenos sensores, dispositivos vestíveis, câmeras inteligentes ou até mesmo dispositivos que intermedeiam as comunicações como *gateways* e *middlewares*, entre outros conforme a arquitetura de alto nível exibida na Figura 1.

Figura 1: Arquitetura IoT de alto nível



Fonte: (HAKIRI et al., 2015)

Uma aplicação de tecnologias IoT pode ser observada nas pulseiras inteligentes cada vez mais populares atualmente. Elas possuem sensores inteligentes capazes de monitorar sinais vitais como batimentos cardíacos, pressão arterial e oxigenação sanguínea e enviam esses dados para o celular do proprietário. Posteriormente alguma aplicação no smartphone pode realizar análises e disparar alertas no dispositivo, ou ainda, pode enviar automaticamente esses dados para um serviço externo que executará algoritmos de Inteligência Artificial e realizará previsões

³Machine to Machine (M2M): conceito que descreve a comunicação entre máquinas, sensores e dispositivos em geral sem nenhuma ou com mínima intervenção humana.

sobre a saúde do usuário, por exemplo.

Há grandes desafios quando falamos em dispositivos *IoT* integrados ao cotidiano das pessoas em ecossistemas inteligentes (CHEN et al., 2014). Carros inteligentes podem interagir com semáforos inteligentes para melhorar o controle de tráfego. Pagamentos instantâneos podem ser realizados apenas por entrar em um estabelecimento, pegar um produto e sair pela porta, tudo integrado com uma carteira digital. Enfim, as aplicações são diversas como cidades, hospitais, casas, supermercados e outros serviços inteligentes que interajam entre si através da internet para facilitar o cotidiano das pessoas.

2.2 Computação em Nuvem e em Neblina

Computação em Nuvem ou *Cloud Computing* (MELL; GRANCE, 2011), é um conceito que se aplica a recursos de rede, servidores, armazenamento, aplicações e serviços sob demanda disponibilizados por meio da internet a diferentes clientes com poucos esforços de gestão e custo reduzido.

Como um novo paradigma, trouxe algumas inovações para a computação. Os recursos podem ser compartilhados pelos provedores com diferentes usuários que pagam somente pelo que for utilizado (sistema *pay-as-you-go*), não gerando desperdício de recursos e possibilitando que seja adquirido o serviço ideal para cada necessidade. Além disso, os recursos não estão necessariamente próximos dos usuários e dos outros serviços, mas sim centralizados em grandes *data centers*, característica que pode favorecer em termos de custos e manutenção.

Embora seja uma tecnologia muito benéfica para diversos casos, há alguns problemas em aberto quando tratamos de cidades inteligentes e dispositivos *IoT*. A centralização dos *data centers* é um problema para aplicações sensíveis a alta demanda (VILELA et al., 2020), pois o congestionamento de rede ou as grandes distâncias geográficas podem gerar uma latência crítica para estes sistemas.

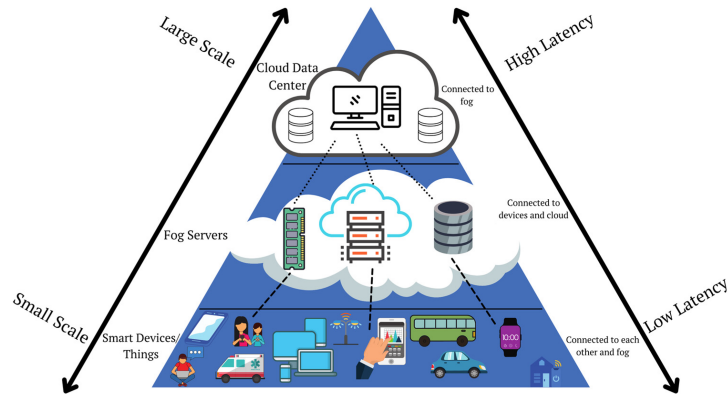
Portanto, com o crescente número de dispositivos *IoT* se viu a necessidade de uma camada intermediária para a comunicação com os serviços na Nuvem (NAEEM et al., 2019), a partir disso, surge o conceito de Computação em Neblina ou Névoa (*Fog Computing*) como uma camada que permite um processamento prévio à *Cloud*, mais próximo dos dispositivos da borda. Com funções bem específicas, ela proporciona melhores tempos de resposta, menos custos com infraestrutura, mais desempenho, melhor escalabilidade e mais segurança sobre os dados.

A Fog é um conceito emergente que busca complementar o uso da Cloud. Em outras palavras, o propósito não é ser uma substituta, mas sim, uma nova camada, um complemento (BONOMI et al., 2012). Conforme a Figura 2, a Fog está mais próxima dos usuários se conectando com os dispositivos na Edge e com os data centers na Cloud.

Essa proximidade permite menores latências nas interações entre clientes e servidores bem como possibilita a redução da quantidade de dados enviados à Cloud, pois a Fog realiza os processamentos triviais para os dispositivos e apenas requisições realmente necessárias são en-

viadas para as aplicações na Cloud (para consolidação de informações ou processamentos mais pesados, por exemplo). Além disso, uma abordagem com Fog, melhora o desempenho na resposta aos dispositivos (VILELA et al., 2020), pois a carga de trabalho por servidor é fortemente reduzida o que diminui as chances de uma sobrecarga em um único ponto da rede.

Figura 2: Hierarquia de arquitetura Edge-Fog-Cloud que exemplifica a escala de dispositivos e as relações de latência dependendo da proximidade com os usuários em cada camada.



Fonte: (FAROOQI et al., 2022)

2.3 Cidades Inteligentes

Devido ao crescimento populacional acelerado dos ambientes urbanos torna-se necessário que tenhamos mais cidades com locais e serviços eficientes para atender aos cidadãos. A partir dessa necessidade, surge o conceito de Cidades Inteligentes (ou *Smart Cities*) que são sistemas complexos, frequentemente chamados de "sistemas de sistemas" (KHATOUN; ZEADALLY, 2016), por se tratarem da integração entre diversos dispositivos e serviços. Conforme ilustrado na Figura 3, a maioria dos modelos de cidades inteligentes consistem em seis componentes: governo, economia, mobilidade, meio ambiente, habitação e pessoas.

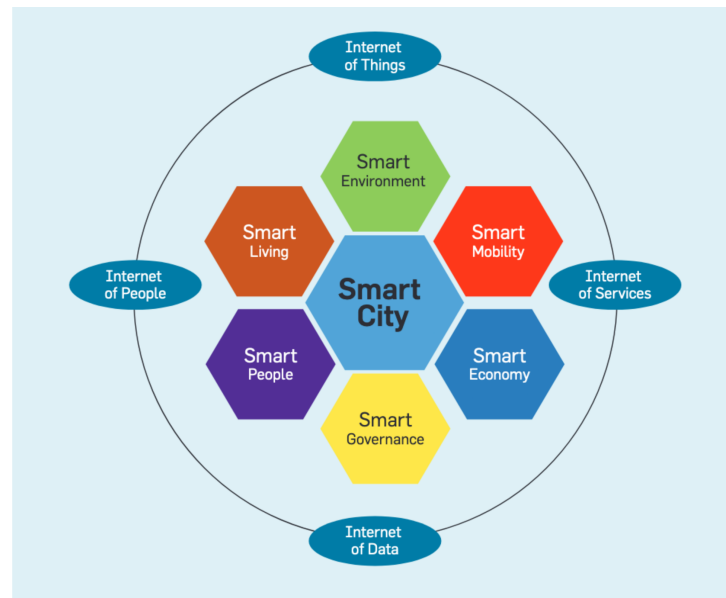
Contudo, existem alguns desafios latentes ao tratarmos de cidades inteligentes e que em sua maioria estão ligados a escalabilidade e QoS⁴ do modelo, pois tratam de integrações entre sistemas críticos e de alta prioridade como, por exemplo, gestão de tráfego de trânsito (componente da mobilidade) e gestão de saúde pública (componente da habitação).

O tratamento e armazenamento de dados é um desses desafios já que em uma cidade inteligente a integração entre dados é a chave para uma gestão mais eficiente. Além disso, é enorme a quantidade de dados gerados em Smart Cities (CHANG, 2021) e conseguir compreender, armazenar e recuperar esses dados da melhor maneira possível exige uma arquitetura sólida, escalável e segura. Uma das fontes de dados de cidades inteligentes são os dispositivos IoT

⁴*Quality of Service (QoS)*: conceito aplicado em redes que priorizam aplicações de alto desempenho e que preocupam-se em manter bons parâmetros relacionados a largura de banda, latência, perdas de pacotes e *jitter* (variação de latência).

(Subseção 2.1) e a chave para uma arquitetura robusta que possa atendê-la pode estar na Figura 2 a partir da união entre esses dispositivos (na Edge) com Cloud e Fog Computing (Subseção 2.2).

Figura 3: Componentes de Cidades Inteligentes



Fonte: (KHATOUN; ZEADALLY, 2016)

2.4 Fragmentação de Dados

Em um banco de dados os dados relacional (RDBMS) geralmente os dados estão organizados semanticamente em tabelas, colunas e linhas. As linhas são os registros armazenados, as colunas são os atributos desses registros e as colunas são os agrupamentos desses dados em um conjunto comum. Esta estrutura é muito robusta e atende a diversas soluções acadêmicas e de mercado a anos. Contudo, há uma certa dificuldade de lidar com essa arquitetura quando o assunto é Big Data, armazenamento distribuído, alto desempenho e concorrência.

Para este fim surgiram os bancos de dados não relacionais (NoSQL) que foram concebidos com a ideia de terem esquemas flexíveis, serem escaláveis e preparados para atender às arquitetura que necessitam de alta disponibilidade e performance de armazenamento (JAWARNEH et al., 2021). Ademais, quando falamos nestes tipos bancos de dados um conceito que vem a tona é o de Fragmentação de Dados ou *Sharding*. Sharding não é uma termo novo, tampouco uma exclusividade de NoSQL, mas popularizou-se porque a maioria dos bancos NoSQL trazem este recurso e o aplicam nativamente de maneira simplificada.

O conceito trata da técnica de divisão, a partir de um critério estabelecido, de um grupo de dados em diferentes subgrupos (CORBELLINI et al., 2017) localizados em nós da rede de banco de dados. Existem diferentes maneiras de fragmentar os dados usando Sharding. Um exemplo seria a divisão por meio de um critério de igualdade de quantidade de itens por nó,

buscando-se sempre manter a mesma quantidade em cada um a fim de fornecer o equilíbrio de carga entre eles. Outra aplicação possível seria através da definição de um critério de divisão a partir de um ou mais campos dos dados armazenados e, a partir disso, definem-se faixas ou valores exatos por nó.

Por sua estrutura dinâmica, os bancos de dados NoSQL geralmente possuem o recurso de Sharding e alguns deles, inclusive, oferecem soluções mais robustas para aplicação em uma arquitetura de armazenamento distribuído que trazem inúmeras vantagens como a possibilidade de concorrência para resolver uma consulta, a proximidade geográfica entre os clientes e os dados persistidos, a replicação dos dados para melhorar a tolerância a falhas da arquitetura e até mesmo o cache fragmentado por unidade de divisão.

3 TRABALHOS RELACIONADOS

Nesta seção serão detalhados os trabalhos relacionados, ou seja, aqueles trabalhos que possuem algum vínculo direto com esta pesquisa que estou conduzindo. A seleção dos trabalhos foi feita através da busca em bases de dados confiáveis como IEEE Xplore⁵, SpringerLink⁶, ScienceDirect⁷, MDPI⁸ e ACM Digital Library⁹ e objetivou principalmente encontrar trabalhos relacionados que tivessem foco em arquiteturas de armazenamento distribuído, arquiteturas hierárquicas em camadas e rastreabilidade de dados distribuídos em cidades inteligentes que possuam uma arquitetura composta por Fog e/ou Cloud. A busca e análise resultou em 7 artigos, conforme o Quadro 1, considerados aderentes a um ou mais aspectos relevantes desta pesquisa e que contribuem diretamente no embasamento do trabalho a ser desenvolvido.

Em (KUDO, 2018) os autores propuseram uma arquitetura para armazenamento dos dados gerados por dispositivos *IoT*. Pela grande quantidade de informações que esses sensores podem gerar, é proposto que esses dados sejam pré-armazenados a nível de *Fog* em Bancos de Dados não relacionais, pela sua alta capacidade de escalabilidade e pelas estruturas de dados dinâmicas suportadas por este modelo. Além disso, esses bancos de dados trabalham muito bem de forma distribuída, o que para uma *Smart City*, por exemplo, é algo extremamente positivo, pois possibilita a proximidade geográfica aos usuários. Neste trabalho utilizou-se MongoDB pelo recurso de GridFS que permite armazenar documentos maiores de 16 MB (limite de tamanho dos documentos do MongoDB) e também permite acesso as porções de dados sem a necessidade de carga total do registro. Esta *feature* permite armazenar grandes quantidades de dados, pois ao invés de armazenar todos os dados em um único documento ela os divide em partes.

Ainda neste artigo, o autor propõe um modelo de armazenamento baseado em três níveis. O primeiro armazena os dados originais dos sensores na *Fog*. Já o segundo e terceiro níveis

⁵IEEE Xplore: <<https://ieeexplore.ieee.org/>>

⁶SpringerLink: <<https://link.springer.com/>>

⁷ScienceDirect: <<https://www.sciencedirect.com/>>

⁸MDPI: <<https://www.mdpi.com/>>

⁹ACM Digital Library: <<https://dl.acm.org/>>

Quadro 1: Seleção de Trabalhos Relacionados

Base	Trabalho	Título
IEEE	Kudo (2018)	Fog computing with distributed database
IEEE	Sinaeepourfard et al. (2018)	Data preservation through fog-to-cloud (F2C) data management in smart cities
IEEE	Shwe, Jet e Chong (2016)	An iot-oriented data storage framework in smart-city applications
Springer Link	Abreu et al. (2017)	A resilient Internet of Things architecture for smart cities
Springer Link	Lomotey, Pry e Chai (2018)	Traceability and visual analytics for the Internet-of-Things (IoT) architecture
Science Direct	Zhang (2020)	Design and application of fog computing and Internet of Things service platform for smart city
MDPI	Benhamida et al. (2022)	Dynamic architecture for collaborative distributed storage of collected data in fog environments

Fonte: Elaborado pelo autor.

localizam-se na Cloud. No segundo nível são armazenados os dados extraídos do processamento primário. E o terceiro nível armazena os resultados das análises realizadas sobre os dados extraídos. Esta arquitetura prevê que o acesso aos dados seja realizado tanto pela Cloud, quando diretamente nos nós da Fog, porém não se preocupa com a mobilidade dos usuários.

Em (SINAEPOURFARD et al., 2018) é notável a preocupação do trabalho com a gestão do ciclo de vida dos dados o que o vincula diretamente a esta pesquisa. O modelo proposto foi validado com êxito na cidade de Barcelona na Espanha e se preocupa com três fases principais que cobrem todo o ciclo de vida dos dados: a aquisição, o processamento e a preservação. Essas fases foram propostas para uma Smart City que utiliza uma arquitetura de três camadas: camadas de Fog 1 e 2 e camada de Cloud. A primeira camada, Fog 1 ou Edge, é a camada mais próxima dos usuários que realiza a interface dos usuários e seus dispositivos com a Fog 2. A segunda camada, Fog 2, é uma camada intermediária entre a Cloud e a Fog 1 e responsável pelo processamento e pelo armazenamento de dados com maior taxa de acesso. A terceira e mais superior camada, a Cloud, é o ponto mais afastado dos usuários o que gera mais latência, ela é responsável por armazenar dados históricos e por realizar processamentos mais pesados.

No trabalho (SHWE; JET; CHONG, 2016) propõe-se um framework de armazenamento distribuído que está estruturado em duas camadas. A primeira camada é uma rede backbone em malha que armazena os dados do sistema em tempo real de maneira distribuída. Já a segunda camada é responsável pelo armazenamento central. Ambas camadas foram idealizadas para a Cloud, mas o trabalho não exclui a possibilidade de implementar alguns serviços como Fog na primeira camada (malha). Cada nó (ou Access Point) na primeira camada, possui sua própria rede de armazenamento e contribui para a formação de uma rede em malha onde cada nó possui acesso aos dados dos demais. Dados de acesso recorrente e sensíveis a latência são armazenados na primeira camada e dados históricos e de longo prazo são enviados para a camada superior

através de rotinas de agregação executadas em segundo plano.

Em (ABREU et al., 2017) tem-se como objetivo principal a preocupação com a resiliência e tolerância a falhas de uma arquitetura IoT para Smart Cities. Essa pesquisa foi aplicada a cidade de Lisboa em Portugal e implementa serviços capazes de fornecerem dashboards que auxiliem na tomada de decisões por instituições governamentais e que possibilitem, através da análise de dados, a predição e atuação em rotinas na cidade sem intervenção humana. O modelo proposto é dividido em três camadas (IoT, Fog e Cloud) e em cada camada são implementadas múltiplas instâncias de microsserviços. Dessa maneira, a indisponibilidade de um único serviço não incapacita os demais e a redundância de instâncias juntamente com o balanceamento de carga garante maior disponibilidade das aplicações executadas. O trabalho reafirma a importância da união Fog-Cloud para distribuição da carga de trabalho e diminuição da latência e coloca essa estrutura como fator chave para uma arquitetura resiliente em cidades inteligentes.

Em (LOMOTÉY; PRY; CHAI, 2018) os autores propuseram um modelo multicamadas de arquitetura IoT que facilita a rastreabilidade por meio de um armazenamento baseado em Cloud que registra informações dos dispositivos e metadados das requisições realizadas. Sob demanda o repositório de metadados deve ser solicitado para responder o rastreamento de determinado dado entre os dispositivos da rede. Além disso, por meio de ferramentas de visualização as metodologias propostas permitem determinar o vínculo entre os dispositivos IoT a fim de compreender o fluxo de dados na rede. Essa arquitetura busca resolver o problema da correlação entre diferentes dados de mesmos usuários, pois a mobilidade na rede de uma Smart City bem como os diversos dispositivos IoT geram uma enorme quantidade de dados. Contudo, por se tratar de um modelo baseado em Cloud a latência tanto de gravação quanto de escrita são um desafio a ser enfrentado.

Em (ZHANG, 2020) enfatizou-se as vantagens do uso colaborativo de Fog e Cloud Computing para lidar com dados IoT em Smart Cities. A arquitetura proposta é baseada em um modelo de camadas que trata os dados vindos dos dispositivos IoT na camada de Fog Computing e posteriormente na camada de Cloud Computing. Cada nó na Fog possui alguns serviços virtualizados que realizam operações de agregação e consolidação. Quando há sobrecarga (ou identificação de possível sobrecarga) os dados na Fog são encaminhados para a Cloud, embora essa estratégia aumente a latência pode ser vantajoso para o modelo como um todo. A proposta foi feita pensando em cidades da China, país com acelerado processo de urbanização. Como trabalhos futuros trouxeram a preocupação com a otimização de alocação dos recursos computacionais para a Fog, pois esta camada possui recursos limitados por hardware.

Por fim, em (BENHAMIDA et al., 2022) os autores propuseram uma arquitetura dinâmica para armazenamento distribuído de informações coletadas em ambientes de Fog. A dinamicidade está relacionada a mobilidade inerente da Fog e dos dispositivos IoT e a distribuição com o armazenamento de dados nos nós de Fog. Esta pesquisa também aborda um modelo colaborativo entre Fog e Cloud para minimizar os acessos à Cloud e, por consequência, reduzir a latência. A arquitetura é hierárquica dividida em três camadas: Fog Edge (com Fog Nodes),

Fog Server (com Fog Servers) e Cloud. Em suma, o modelo traz para si a responsabilidade de localização dos registros através de armazenamento local em cada Fog Server atualizado periodicamente em "tabelas de atribuição" que vinculam o sensor IoT ao Fog Node. Além disso, as requisições de autorização são síncronas e os dados e tabelas de atribuição são replicados em cada camada.

Os trabalhos foram analisados entre si através de nove competências importantes para o contexto de uma arquitetura distribuída de armazenamento de dados em Fog e Cloud Computing em Cidades Inteligentes a fim de permitir a rastreabilidade de informações e usuários, ter um sistema resiliente e atingir baixos níveis de latência (conforme o Quadro 2). As características analisadas são:

- Rastreabilidade (C1) — os dados são facilmente rastreáveis a partir de requisições feitas a algum nó da rede.
- Interoperabilidade (C2) — os diversos serviços da arquitetura são integrados facilmente entre diferentes camadas e nós de modo que a comunicação seja padronizada e transparente.
- Hierárquico (C3) — a arquitetura proposta tem hierarquia ou alguma divisão em camadas para melhor dividir as responsabilidades e carga facilitando a gestão dos serviços.
- Possui Fragmentação de Dados (C4) — os dados são armazenados de modo fragmentado permitindo o armazenamento distribuído e consultas concorrentes as diferentes porções de um ou mais dados.
- Possui Cache (C5) — possui algum sistema de cache que melhore o tempo de resposta e evite operações de E/S desnecessárias para informações já consultadas previamente.
- Possui Replicação (C6) — possui alguma forma de redundância dos serviços propostos e dos dados armazenados de maneira a ser uma arquitetura tolerante a falhas e ter um melhor QoS.
- Validado em alguma cidade real (C7) — a arquitetura proposta já foi validado em alguma cidade real ou simulação que a reproduza.
- Possui Fog (C8) — a arquitetura engloba nós de Fog Computing próximos aos dispositivos e/ou na borda da rede.
- Possui Cloud (C9) — a arquitetura engloba uma comunicação com serviços em Cloud que não estão necessariamente próximos dos dispositivos.

Analisando o Quadro 2 é possível ver que todos os trabalhos tem processamento em Cloud Computing e que, na maioria das ocorrências, essa camada é responsável por processar dados de longo prazo que são extraídos dos nós mais abaixo, mas que também responde por requisições

imediatas, o que traz maiores latências para o modelo já que não há proximidade geográfica nesse caso. Já sobre Fog Computing, a maioria dos trabalhos possui uma camada de Fog que em alguns casos é uma camada opcional ou com potencial inexplorado, sendo mencionada apenas como uma possibilidade e não abordada como aplicação principal do modelo. Logo, percebe-se que embora tenha muitas vantagens, o uso de um processamento intermediário mais próximo dos usuários ainda não é unanimidade em todas as arquiteturas propostas.

Quadro 2: Comparativo entre os trabalhos selecionados

Trabalho	C1	C2	C3	C4	C5	C6	C7	C8	C9
Kudo (2018)	N	S	S	S	N	N	N	Sim, dados na Fog são posteriormente extraídos para a Cloud	Sim, analisa dados pré-processados pela Fog
Sinaeepourfard et al. (2018)	N	S	S	N	S	N	S	Sim, dados na Fog são os dados que precisam de análises em tempo real ou que podem suportar pouco delay	Sim, faz processamento massivo e análises complexas
Shwe, Jet e Chong (2016)	N	S	S	N	N	S	N	Sim, rede em malha que armazena distributivamente dados em tempo real	Sim, armazenamento central de dados históricos e de longo prazo
Abreu et al. (2017)	N	S	S	N	S	S	S	Sim, pré-processa os dados para lidar com a heterogeneidade e serem processados nos serviços da Cloud (ou alguns na Fog)	Sim, camada de serviços e middleware IoT, porém, ambas camadas podem ter parte na Fog e Cloud para lidar com demandas específicas
Lomotey, Pry e Chai (2018)	S	S	N	N	N	N	N	Não, foca na rastreabilidade entre dispositivos IoT e a Cloud	Sim, os serviços de análise dos dados estão na Cloud
Zhang (2020)	N	S	S	N	N	N	N	Sim, a Fog está em uma única camada horizontalizada que processa as demandas críticas quanto a latência	Sim, a Cloud processa dados mais complexos e que não foram capazes de serem processados pela Fog
Benhamida et al. (2022)	S	S	S	N	N	S	N	Sim, possui duas camadas de Fog: uma mais próxima do usuário na borda da arquitetura e outra mais acima que realiza os processamentos. Os dados são replicados em cada camada e há uma tabela de atribuição local em cada nó	Sim, a Cloud realiza os processamentos mais exigentes e de dados históricos. Também possui uma cópia da tabela de atribuição e também possui os dados das camadas inferiores replicados

Fonte: Elaborado pelo autor.

Referente a validação em alguma cidade real (C7) é possível ver que somente dois trabalhos validaram sua arquitetura em um ambiente real ou aplicado a uma simulação do escopo real. Segundo Sinaeepourfard et al. (2018) realizou-se a validação na cidade de Barcelona na Espanha

e em Abreu et al. (2017) na cidade de Lisboa em Portugal. Além disso, vemos um outro extremo quanto as competências de Rastreabilidade (C1), Fragmentação de Dados (C4), Cache (C5) e Replicação (C6) que juntas trazem uma contribuição enorme para um modelo distribuído e que em nenhum caso foram mencionadas na mesma proposição de arquitetura.

Portanto, é possível observar algumas lacunas nas proposições dos trabalhos relacionados. Primeiramente, que não temos um modelo que integre diversas características importantes para a rastreabilidade de dados em uma arquitetura de armazenamento distribuído, ou seja, faltam proposições que abordem maneiras eficientes de rastrear os dados armazenados. Além disso, encontram-se muitas abordagens que realizam a replicação em excesso de metadados e/ou registros dos dispositivos IoT em diversas camadas do modelo o que gera um *overload* na rede que poderia ser evitado utilizando-se algum mecanismo centralizado de armazenamento de metadados que consiga localizar os dados armazenados localmente em cada Fog Node. Por fim, poucas abordagens especificam qual tipo e/ou sistema de banco de dados será utilizado na arquitetura, o que é um ponto importantíssimo para decisões de implementação do modelo.

4 MODELO

Esta seção visa descrever o modelo DASTData (acrônimo para *Distributed Architecture to Store and Trace Data*). A partir da Fundamentação Teórica (Seção 2) e dos Trabalhos Relacionados (Seção 3), foi possível definir aspectos importantes para se projetar um modelo de armazenamento para Smart Cities. Dessa maneira, foi definido que o foco do modelo é a proposição de uma arquitetura em três camadas (Edge, Fog e Cloud) para armazenamento distribuído em smart cities. Essa arquitetura deve contemplar a rastreabilidade e acesso concorrente aos dados dos dispositivos da camada Edge armazenados na Fog.

Para detalhar cada aspecto, na Subseção 4.1 cito e justifico as decisões de projeto que foram tomadas na elaboração da arquitetura. Em seguida, na Subseção 4.2 descrevo detalhadamente as características da arquitetura a fim de explicar e exemplificar seus objetivos. Por fim, na Subseção 4.3 concluo com a definição dos serviços e rotinas que serão propostos inicialmente.

4.1 Decisões de Projeto

Conforme visto na Seção 3 referente aos trabalhos relacionados, especialmente a partir da análise realizada e resumida no Quadro 2, existem algumas competências essenciais na construção de um modelo para uma arquitetura de armazenamento. Portanto, algumas decisões de projeto foram tomadas quanto a definição do que o modelo deve possuir e atender, chegando-se no seguinte cenário:

- Teremos três camadas na arquitetura chamadas Edge, Fog e Cloud. Elas se comunicarão através de interações bloqueantes e não bloqueantes. Também deverão trabalhar de maneira integrada para lidar com os dados que irão percorrer os seus serviços na arquitetura.

- Será utilizado banco de dados NoSQL devido as suas vantagens em um contexto distribuído e de dados com estrutura dinâmica, como já mencionado anteriormente.
- As camadas devem ser hierárquicas de modo que cada subcamada lide com dados específicos de sua competência a ser definida.
- O armazenamento dos dados dos dispositivos deve ser particionado via Sharding e distribuído nos nós da Fog, de modo a possibilitar consultas concorrentes e proximidade geográfica em relação aos usuários.
- A arquitetura deverá ser capaz de lidar com dados de usuários móveis de modo a permitir a rastreabilidade desses usuários entre os diversos nós que a compõe.
- As instâncias de banco de dados em cada nó da Fog estarão interconectadas através da rede formada pelo uso de Sharding. Esta abordagem permite que ações de balanceamento de carga entre os nós sejam menos custosas, pois não há necessidade de transmissão prévia dos dados para que um novo nó seja responsável pela demanda.
- A fim de garantir maior disponibilidade dos dados e redundância para casos de falha, os bancos de dados deverão possuir replicação. Essa abordagem é transparente para o usuário já que os acessos aos dados são centralizados e o próprio SGBD cuida da gerência da replicação.

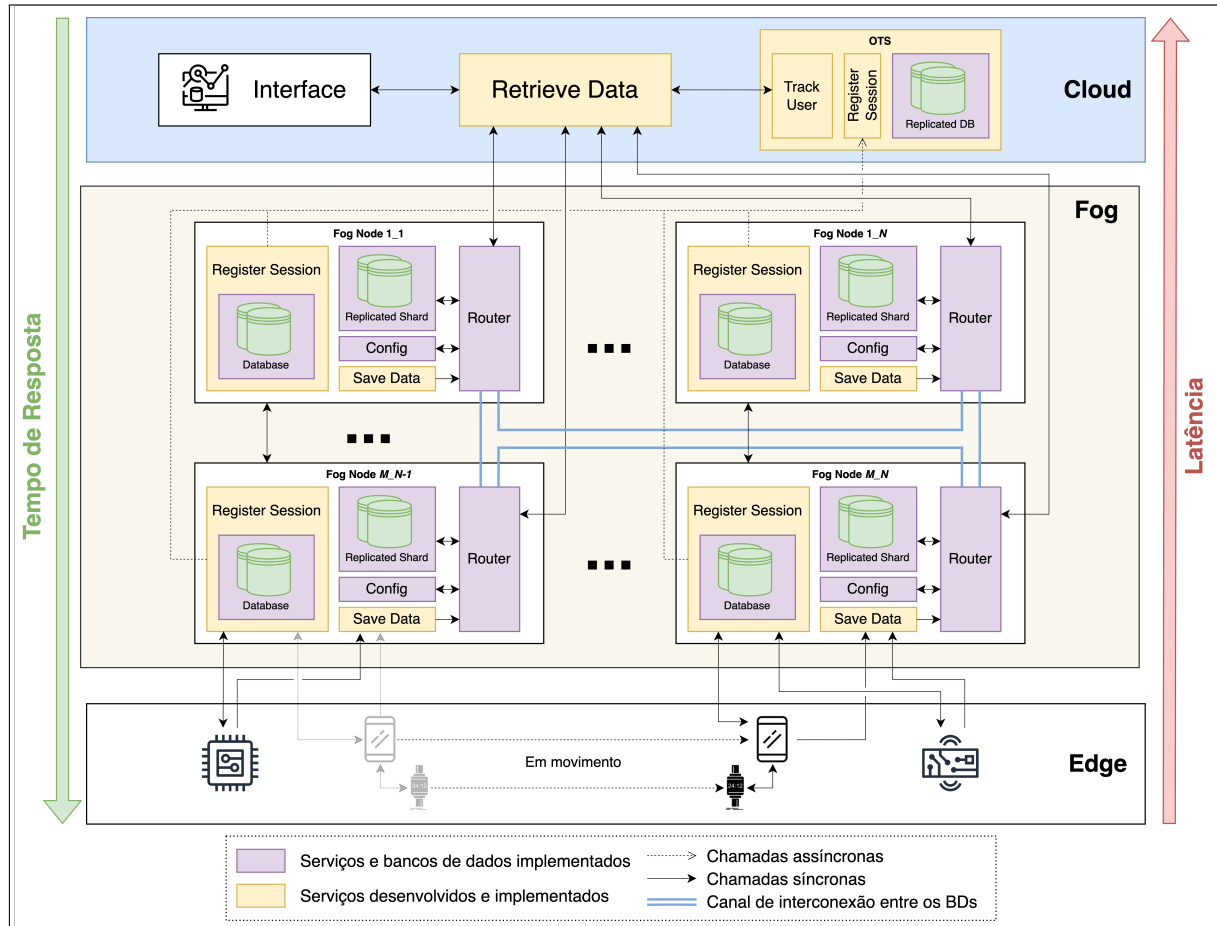
4.2 Arquitetura

A arquitetura proposta a partir das decisões de projeto mencionadas na subseção anterior está dividida em três camadas: Edge, Fog e Cloud. Neste contexto, temos que o foco do modelo é no armazenamento distribuído na Fog e na rastreabilidade dos dados. Para concentrar serviços sensíveis à latência, os nós de Fog Computing serão dispostos próximos geograficamente aos clientes que estão na Edge. Referente a Cloud, essa camada geralmente estará mais afastada, podendo facilmente estar em outro estado ou país, a depender do provedor do serviço. Essa estrutura é voltada para uma Smart City para uso genérico, mas poderia ser usada em diversos contextos como, por exemplo, a gestão de saúde pública. Neste caso, dados de sinais vitais gerados na Edge a partir de sensores IoT como pulseiras inteligentes, se comunicam com um Fog Node que armazena esses registros localmente.

Em resumo, a camada Edge é responsável por gerar os dados para o modelo. A Fog por armazenar esses dados de forma distribuída e por realizar o processamento dos dados críticos que necessitam de pronta resposta e baixa latência. Por fim, a Cloud é responsável por computações mais pesadas que demandam mais tempo e por ser o canal de entrada de atores interessados na gerência da Smart City como gestores públicos, instituições, governos, etc. A Figura 4 ilustra a arquitetura proposta e a disposição dos clusters de armazenamento entre os diferentes nós da Fog. Ademais, as subseções a seguir detalham cada nível e aspecto da arquitetura, detalhando

sobre as camadas Edge, Fog e Cloud, a Fragmentação e Replicação dos dados e a Rastreabilidade dos usuários.

Figura 4: Proposta de arquitetura de armazenamento distribuído para Smart Cities.



Fonte: Elaborada pelo autor.

4.2.1 Edge

A camada denominada Edge é composta basicamente por dispositivos e ambientes inteligentes como gateways IoT, smartphones, pulseiras, carros, câmeras, semáforos, casas e hospitais inteligentes bem como todos aqueles dispositivos IoT que os cidadãos podem possuir. Os dispositivos da Edge ou *Edge Devices* se conectam com a Fog primeiramente através de um registro de sessão. Esse procedimento habilita o dispositivo a se comunicar com determinado Fog Node e a utilizar os serviços que ele disponibiliza. Além disso, os dispositivos na Edge podem ter um comportamento móvel, ou seja, em algum momento estão conectados a um Fog Node e posteriormente se movimentam na cidade e acabam conectando-se a outro Fog Node. Essa rotina para o Edge Device é idêntica a primeira conexão a qualquer Fog Node, tornando-se transparente para o usuário final os procedimentos realizados pelo modelo para possibilitar a mobilidade e posterior rastreamento dos dados.

4.2.2 Fog

A camada denominada Fog está localizada na arquitetura entre a Edge e a Cloud e é responsável por realizar os processamentos mais sensíveis à latência para os Edge Devices, como análises de sinais vitais ou de gestão de trânsito, por exemplo. Esta camada possui os serviços que auxiliam na operação da Smart City sendo alguns implementados na arquitetura apresentada nesse trabalho e outros com possibilidade de implementação futura. São possibilidades, por exemplo, serviços de controle de conexão de Edge Device (Subseção 4.3.1), de salvamento (Subseção 4.3.2), busca, análise e consolidação de dados, de balanceamento de carga, entre outros. Dentro da Fog, também ocorre um processo de hierarquização dos nós de modo que nós mais próximos da Cloud armazenam dados mais consolidados que em níveis inferiores em seu banco de dados local. Essa abordagem também possibilita que os serviços sejam escalados em um modelo *bottom-up*, de maneira que, quando julgar necessário, um serviço possa transferir uma operação para o nó pai.

Embora os dados sejam isolados geograficamente e estejam próximos dos usuários através da técnica de *Sharding*, utilizar este recurso faz com que todas as instâncias consigam acessar os dados umas das outras. Dessa maneira, algoritmos de balanceamento de carga, por exemplo, ao decidirem que determinada tarefa deve ser executada no nó pai, não precisam sobrecarregar a rede transferindo previamente os dados de um banco de dados para outro. Como os nós estão todos interligados, os dados necessários para realizar a tarefa, no nó com mais recursos livres, estão acessíveis para processamento, deixando a responsabilidade da carga dos dados necessário para serviço requisitado. Além disso, já que os nós da Fog são independentes e lidam com um conjunto menor de dados, tarefas de consolidação que necessitam de informações de mais de um Fog Node, podem ser realizadas de forma concorrente.

Neste trabalho considero que os nós da Fog estão dispostos na cidade conforme decisões de implantação futuras. Tem-se como princípio que as interconexões de bancos de dados são entre todos os Fog Nodes, porém as rotinas de escala são *bottom-up* como citado anteriormente. Sendo assim, a abrangência dos Fog Nodes é em zonas de maneira que um bairro pode ter um ou mais Fog Nodes responsáveis a depender das análises quanto ao tráfego naquela região da cidade. Outra implementação possível seria organizando de modo que vários bairros adjacentes com menos tráfego formem uma região com apenas um Fog Node. Neste ponto, a arquitetura é flexível quanto a essas implementações.

Para atender aos objetivos deste trabalho relacionados com as capacidades de armazenamento e rastreabilidade dos dados da Edge para Fog, no escopo desta pesquisa, serão implementados dois serviços da Fog que serão abordados na Subseção 4.3. Embora não faça parte e não seja abordado pelo modelo proposto, um ponto relevante a ser mencionado refere-se ao algoritmo de seleção de Fog Node. Por se tratar de uma arquitetura preparada para lidar com a mobilidade dos dispositivos na Edge, há a premissa de que haverá um serviço na Edge e/ou Fog responsável pelo controle de quando solicitar o registro de sessão em um novo Fog Node.

4.2.3 Cloud

Na Cloud estão dispostos o serviço Retrieve Data e o serviço de armazenamento dos metadados de rastreamento do usuário (OTS). Neste modelo, a Cloud é a camada mais distante dos usuários e portanto concentra os serviços que toleram um maior tempo de resposta. A distância da Cloud em relação aos demais serviços pode atingir grandes proporções na escala de diferentes estados e países o que pode gerar altas taxas de latência e é um problema para aplicações que necessitam de pronta resposta. Nesta arquitetura são propostos dois serviços da Cloud que auxiliarão na consolidação dos dados e que toleram maiores tempos de resposta.

O primeiro deles é o OTS, responsável por armazenar dados históricos do deslocamento dos usuários entre os nós da Fog. O segundo serviço é o de consulta de dados, *Retrieve Data*, que se comunica com o OTS e com a Fog. Para compor um registro de rastreabilidade dos dispositivos móveis na cidade, o *Retrieve Data* obtém o histórico contido no OTS e, a partir dessas informações, consulta os dados armazenados nos Fog Nodes. A ideia é que a partir da Cloud os atores interessados nos dados da cidade tenham acesso a gráficos, dashboards e relatórios consolidados. Na Subseção 4.3 descrevo em mais detalhes as estruturas de dados, responsabilidades, objetivos e funcionamento desses serviços.

4.2.4 Fragmentação e Replicação

Os dados dos dispositivos da Edge serão armazenados nos bancos de dados da Fog através da técnica de Sharding realizando a fragmentação a partir do código identificador do respectivo Fog Node. Em cada Fog Node há um banco de dados NoSQL que é uma partição da massa global de dados. Esse banco de dados possui um serviço atrelado a ele denominado *Router* que é responsável pelo acesso ao banco de dados, definindo o melhor plano de execução da operação de leitura ou escrita solicitada. Como descrito na Seção 2.4 é necessário definir um campo chave para ser o critério de fragmentação e um dos indexadores dos registros. Para tanto, como há um serviço de salvamento de dados que sempre vincula o identificador do Fog Node no registro, se definiu o campo *shard_server* como uma boa opção de chave para a divisão geográfica dos dados.

Por fim, cada instância de bancos de dados nos Fog Nodes deve ser replicada em mais duas instâncias. A replicação é importante para aumentar a disponibilidade e adicionar uma camada de tolerância a falhas ao modelo. A replicação irá seguir o modelo *master/slave*, ou seja, ao formar a replicação o SGBD elege uma instância para ser a principal que receberá todas as requisições do Router. Caso ocorra alguma indisponibilidade, é realizada uma nova eleição para definir a instância principal. Esse procedimento não gera perdas, pois os dados estão replicados em cada instância. Uma outra vantagem da replicação é quanto ao tempo de resposta para operações de leitura. Como os dados estão replicados nas instâncias é possível realizar configurações para que múltiplas instâncias respondam às operações de leitura, de modo que

ocorra um balanceamento de carga e possibilite paralelismo para processar essas requisições.

4.3 Funcionamento

O funcionamento deste modelo depende de alguns serviços chave. Nas subseções a seguir descrevo os quatro serviços principais que permitem o armazenamento distribuído e rastreabilidade dos usuários.

4.3.1 Fog Register Session

O módulo *Register Session* dos Fog Nodes é responsável por receber as requisições de conexão inicial dos dispositivos na Edge, gerar um identificador de sessão, disponibilizar essa informação para o dispositivo que requisitou e persistir assincronamente esses dados no OTS. A Figura 5 apresenta o diagrama de sequência da operação de registro de sessão realizada a partir de um Edge Device para um novo Fog Node. Esse procedimento deve ser realizado toda vez que um Fog Node ficar responsável por desempenhar tarefas de um Edge Device. O fluxo inicia-se na requisição de registro realizada pelo Edge Device. O Fog Node recebe essa requisição, gera um identificador UUID versão 1¹⁰ para representar a nova sessão chamado *session_id*, armazena esse registro em um banco de dados local do serviço e retorna o identificador para ser utilizado posteriormente pelo Edge Device nas requisições subsequentes.

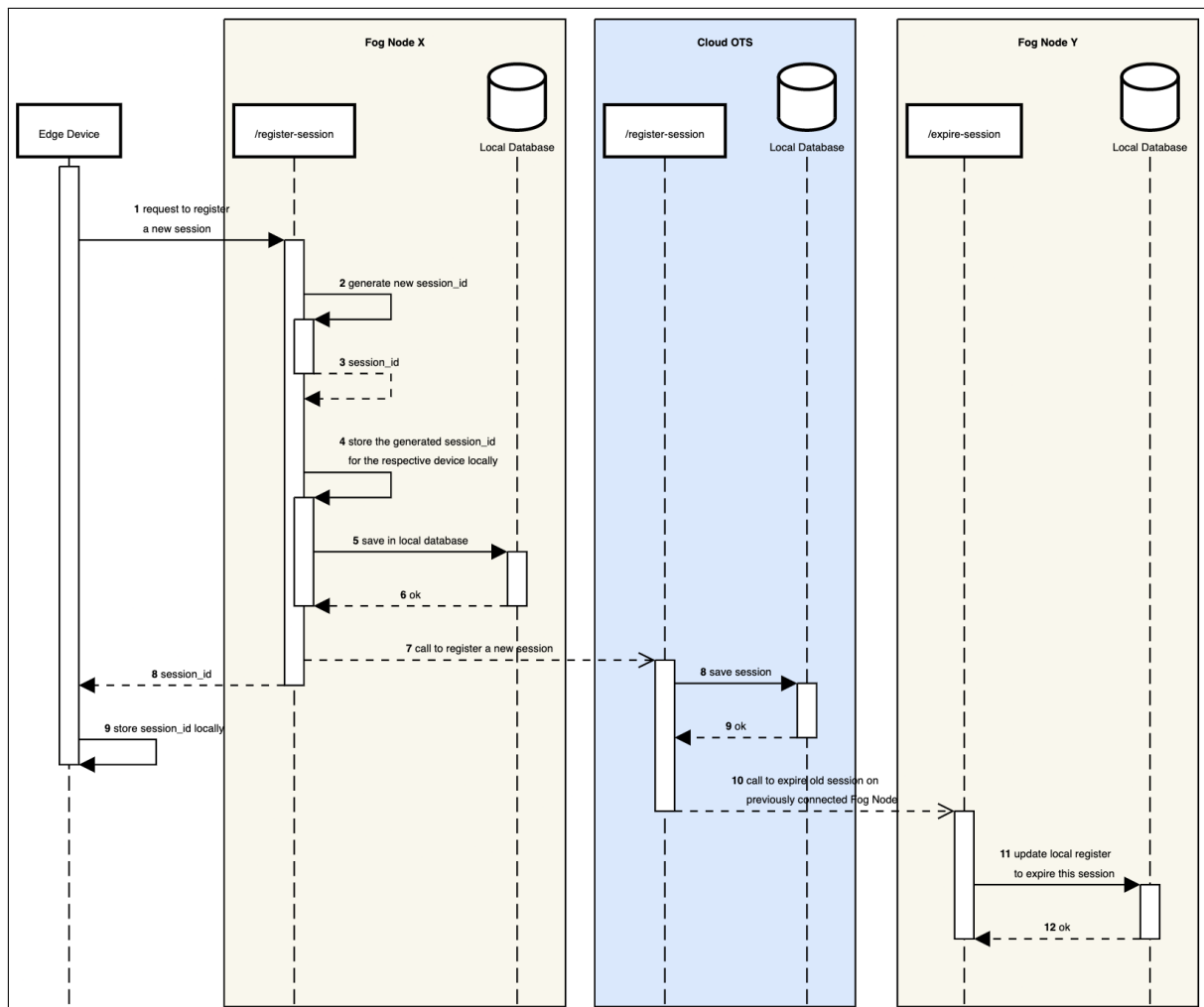
Além disso, o *Register Session* realiza uma chamada assíncrona para o serviço OTS na Cloud que registra essa sessão e, se houver uma conexão prévia em outro Fog Node, requisita assincronamente que a sessão antiga seja expirada. A forma como essas chamadas assíncronas serão realizadas e tratadas não diz respeito ao modelo que está sendo proposto, logo essa decisão fica aberta para as implementações futuras. Contudo, sugere-se que seja usado algum método tolerante à falhas que consiga garantir que os procedimentos serão executados com sucesso, por exemplo, com o uso de serviços de fila ou com o padrão de trocas de mensagens *publish/subscribe*.

4.3.2 Fog Save Data

Save Data é o nome do serviço responsável por receber as requisições que armazenam os dados dos dispositivos da Edge nos Fog Nodes. Ele opera em conjunto com o banco de dados replicado e particionado através de Sharding. Além disso, atua como um *wrapper* agregando mais informações nos dados que serão armazenados, como o identificador do Fog Node, da sessão e do usuário. Sua comunicação com o banco de dados é através do *Router* contido em cada Fog Node juntamente com o serviço de metadados de armazenamento chamado *Config*.

¹⁰UUID v1: é um identificador único universal gerado a partir de um *timestamp* e do endereço MAC do computador que o está gerando.

Figura 5: Diagrama de Sequência das operações síncronas e assíncronas realizadas para registrar-se uma nova sessão em um Fog Node.



Fonte: Elaborada pelo autor.

Na Figura 6 são apresentados dois exemplos de dados armazenados nos Fog Nodes e como seriam os registros de um mesmo usuário que se move entre dois Fog Nodes. O campo *session_id* varia apenas quando há mudança de contexto, o campo *id* trata-se de um identificador único com o formato que melhor se adéqua ao banco de dados da implementação. Cabe destacar que o campo *data* possui estrutura flexível, pois ele contém os dados que foram enviados pelo dispositivo ou serviço para serem armazenados.

Além disso, na Figura 7 é apresentado o diagrama de sequência das operações realizadas no fluxo de salvamento dos dados da Edge para a Fog. O fluxo principal consiste em o Edge Device enviar a solicitação para salvar dados no Fog Node passando um *session_id* válido. O Fog Node valida a sessão informada e realiza o salvamento do registro. As validações quanto a sessão pelo Fog Node são para verificar se foi informado um identificador de sessão e se o valor informado é válido ou não.

Figura 6: Exemplo de estrutura dos dados de um mesmo usuário armazenados em dois Fog Nodes diferentes. No lado (a) representam-se os dados de sessões diferentes para um mesmo usuário, já no lado (b) estão representados dados de uma mesma sessão para o mesmo usuário.

<pre>[... { "id": "<unique-id>", "shard_server": 1002, "session_id": "48a88774-e599-11ec-8fea-0242ac120002", "created_at": "2022-06-01T08:05:00.000Z", "user_id": "d9e4b8cc-b2b0-4ee8-955c-4511f63a7c22", "data": { ... } }, ... { "id": "<unique-id>", "shard_server": 1002, "session_id": "e1c97d5f-e599-11ec-8fea-0242ac120002", "created_at": "2022-06-01T18:31:00.000Z", "user_id": "d9e4b8cc-b2b0-4ee8-955c-4511f63a7c22", "data": { ... } }, ...]</pre> <p style="text-align: center;">(a)</p>	<pre>[... { "id": "<unique-id>", "shard_server": 1003, "session_id": "9fcd4526-beda-43ff-92ff-0242ac120003", "created_at": "2022-06-01T11:46:00.000Z", "user_id": "d9e4b8cc-b2b0-4ee8-955c-4511f63a7c22", "data": { ... } }, ... { "id": "<unique-id>", "shard_server": 1003, "session_id": "9fcd4526-beda-43ff-92ff-0242ac120003", "created_at": "2022-06-01T11:49:34.000Z", "user_id": "d9e4b8cc-b2b0-4ee8-955c-4511f63a7c22", "data": { ... } }, ...]</pre> <p style="text-align: center;">(b)</p>
---	---

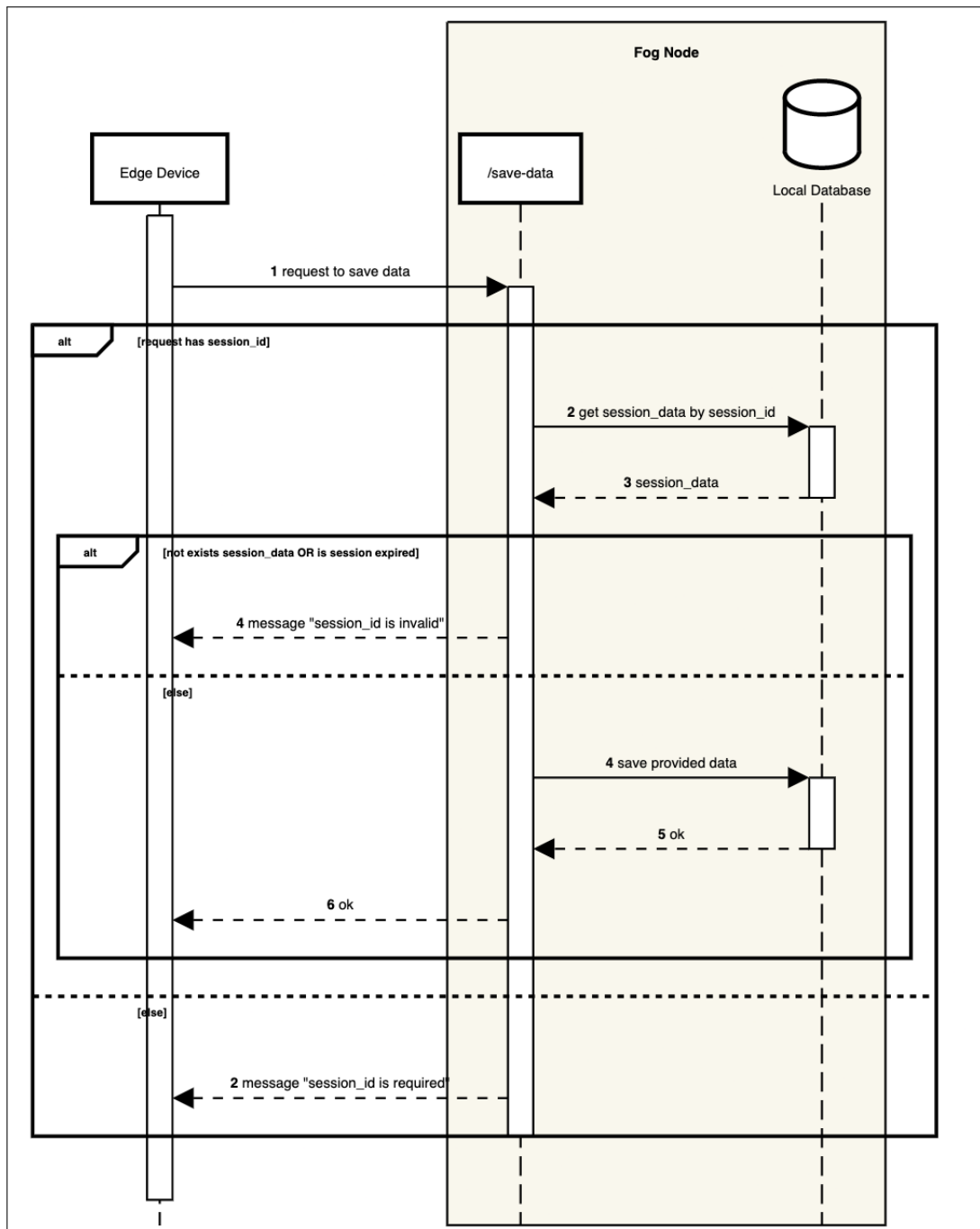
Fonte: Elaborada pelo autor.

4.3.3 Cloud OTS

Este serviço está localizado na Cloud e é o responsável por armazenar os metadados de rastreamento global dos usuários na cidade. Na Figura 8 é apresentado um exemplo da estrutura base dos dados armazenados no OTS e no diagrama de sequência da Figura 5 já foi apresentado anteriormente como os dados chegarão até o OTS por meio de requisições assíncronas. O campo *shard_server* como dito anteriormente será a chave de fragmentação, ou seja, a partir desse campo que será vinculado o dado armazenado à partição correspondente do banco de dados. O campo *session_id* é o identificador da sessão e um dos índices dos dados, ele é utilizado para agrupar dados da mesma sessão, vinculando os registros no OTS com os dados nos Fog Nodes.

A partir desses registros históricos do OTS será possível calcular o tempo de permanência em cada nó através do horário de conexão, realizar o rastreamento e extrair informações úteis desses dados armazenados. A rastreabilidade retornada pelo OTS pode ser utilizada por serviços diferentes para diversos fins. Uma dessas aplicações é a busca de dados nos Fog Nodes a partir de um identificador de usuário e um período apresentada na subseção a seguir com o serviço *Retrieve Data*. Contudo, a aplicação desse histórico do OTS não se limita a busca dos dados nos Fog Nodes, podendo haver serviços "irmãos" ao *Retrieve Data* que manipulam esses dados para obter outras informações.

Figura 7: Diagrama de Sequência do serviço *Save Data* dos Fog Nodes.



Fonte: Elaborada pelo autor.

4.3.4 Cloud Retrieve Data

Este serviço é um exemplo da integração entre o OTS e os dados armazenados nos Fog Nodes para unir o armazenamento distribuído à rastreabilidade dos usuários. Na Figura 9 é apresentado o diagrama de sequência da operação de busca de dados de um usuário em um determinado período. Uma interface genérica solicita os dados para o serviço *Retrieve Data* que encaminha uma consulta ao OTS e após receber o retorno decide a melhor rota para solicitar

os dados, levando em consideração variáveis como a disponibilidade atual dos *Routers*. Assim que definidas as melhores rotas, realizam-se consultas concorrentes diretamente para cada Fog Node dono dos registros. Essa abordagem fornece alto desempenho para o modelo na busca de dados que estão armazenados de maneira distribuída. Após o fim das consultas os dados são aglutinados em formato JSON para serem retornados à interface que realizou a requisição.

Figura 8: Exemplo de estrutura dos dados armazenados no OTS.

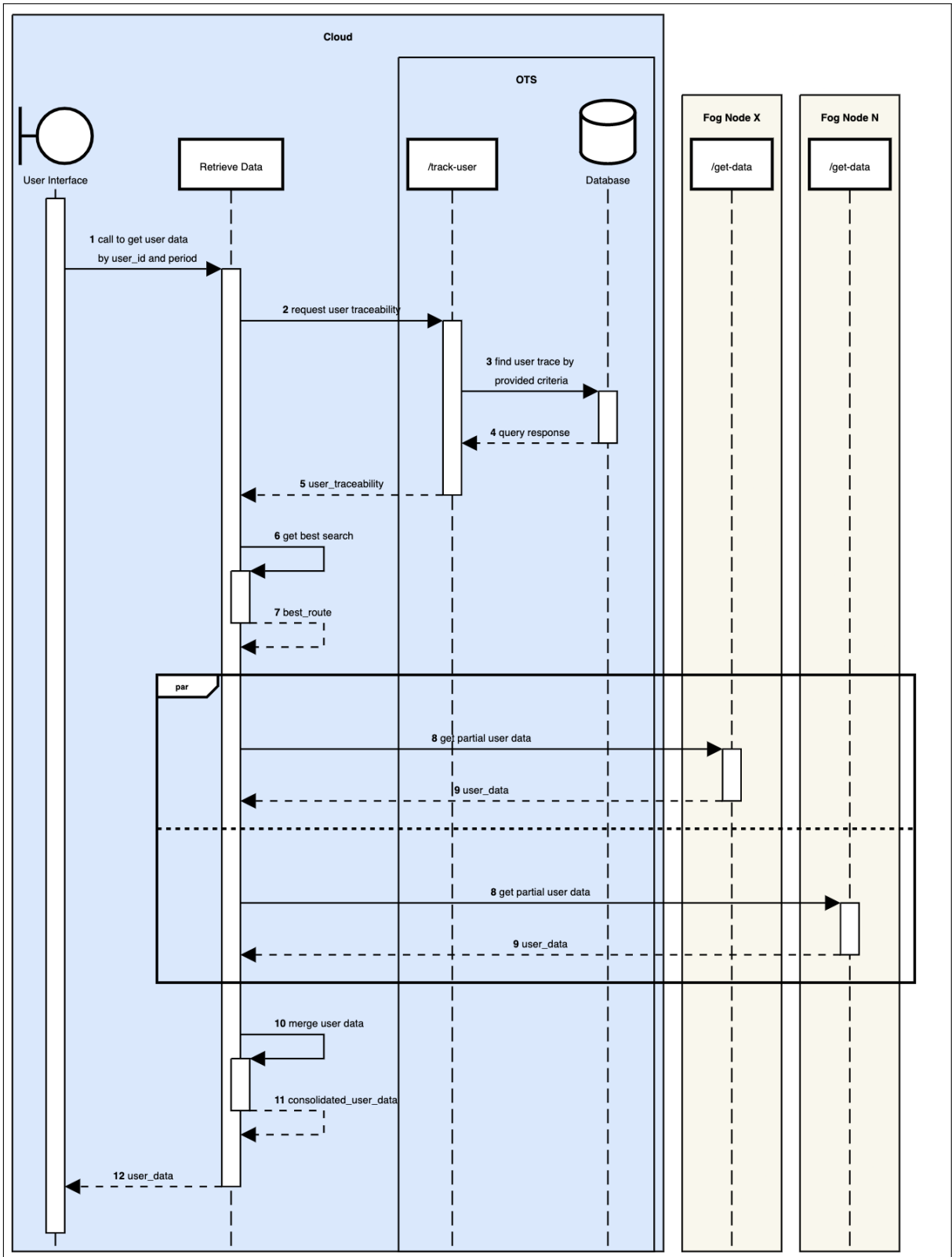
```
[
  {
    "id": "<unique-id>",
    "shard_server": 1002,
    "router_address": "layer-10-node-02.fog.com.br",
    "connected_at": "2022-06-01T08:00:00.000Z",
    "user_id": "d9e4b8cc-b2b0-4ee8-955c-4511f63a7c22",
    "session_id": "48a88774-e599-11ec-8fea-0242ac120002"
  },
  {
    "id": "<unique-id>",
    "shard_server": 1003,
    "router_address": "layer-10-node-03.fog.com.br",
    "connected_at": "2022-06-01T11:45:34.000Z",
    "user_id": "d9e4b8cc-b2b0-4ee8-955c-4511f63a7c22",
    "session_id": "9fcd4526-beda-43ff-92ff-0242ac120003"
  },
  {
    "id": "<unique-id>",
    "shard_server": 1002,
    "router_address": "layer-10-node-02.fog.com.br",
    "connected_at": "2022-06-01T18:30:15.000Z",
    "user_id": "d9e4b8cc-b2b0-4ee8-955c-4511f63a7c22",
    "session_id": "e1c97d5f-e599-11ec-8fea-0242ac120002"
  }
]
```

Fonte: Elaborada pelo autor.

5 METODOLOGIA DE AVALIAÇÃO

Para avaliar o modelo será realizada a implementação de um protótipo que simule a organização de uma Smart City e que possibilite a instalação de diversas instâncias de banco de dados a fim de obter métricas de avaliação que possibilitem a análise de resultados do modelo proposto. Portanto, nesta seção será descrito o processo de validação. Na Subseção 5.1 é descrito como será feita a prototipação e emulação do modelo. Na Subseção 5.2 é apresentada a infraestrutura de testes montada para avaliar a arquitetura e quais dispositivos serão utilizados na emulação. Em seguida, na Subseção 5.3 serão descritas as variantes consideradas para os testes. Por fim, na Subseção 5.4 são definidas as métricas utilizadas para avaliar o modelo.

Figura 9: Diagrama de Sequência do serviço *Retrieve Data*.



Fonte: Elaborada pelo autor.

5.1 Protótipo e Emulação

O protótipo criado para viabilizar a emulação contém a implementação das camadas Edge, Fog e Cloud, conforme a Subseção 4.2. Para atender os objetivos do modelo proposto, o banco de dados selecionado foi o *MongoDB v4.2*¹¹. Por ser NoSQL e orientado a documentos esse banco de dados atende ao modelo, pois possui nativamente estruturas de dados flexíveis e recursos de sharding por zonas e replicação. A linguagem de programação escolhida para os serviços do protótipo foi *JavaScript* em associação com o *runtime* para execução de código do lado do servidor *Node.js v16*¹². A facilidade na integração entre a linguagem de programação e esse banco de dados através do ODM¹³ *Mongoose*¹⁴ foi fator chave para a sua escolha.

Os dados gerados a partir da Edge foram simulados a partir da geração de datasets utilizando-se a biblioteca *Faker.js*¹⁵. No repositório disponível em <<https://github.com/daniellferreira/fog-sharded-storage>> foram implementados a infraestrutura de um Fog Node e seus serviços *Register Session* e *Save Data*. Os serviços da Cloud *OTS* e *Retrieve Data* foram implementados nos repositórios <<https://github.com/daniellferreira/ots>> e <<https://github.com/daniellferreira/ots-retrieve-data>>, respectivamente. A emulação buscou reproduzir operações de leitura e escrita nos bancos de dados dos Fog Nodes de uma Smart City, tanto para a manipulação unitária dos registros nos Fog Nodes como a partir da busca da rastreabilidade de determinado usuário a partir da Cloud.

5.2 Infraestrutura de testes

A infraestrutura utilizada para avaliar o modelo foi composta dos seguintes itens por camadas:

- Edge – foi utilizado um computador com processador hexa-core Intel Core i7-9750H com clock de 2,6 GHz, 16 GB de memória RAM DDR4 com frequência mínima de 2667 MHz e sistema operacional macOS versão 11.6.5. Localmente foram simuladas múltiplas instâncias de Edge Devices através da aplicação Apache JMeter¹⁶
- Fog – foi utilizado um computador para representar dois Fog Nodes através da execução de containers Docker independentes. Foram implementados os serviços *Save Data*, *Config* e *Router* da Fog e o banco de dados MongoDB com sharding e replicação em duas instâncias. Referente a configuração deste computador trata-se de um processador quad-core Intel Core i5-8265U com clock de 1,6 GHz, 20 GB de memória RAM DDR4 com

¹¹MongoDB: <<https://www.mongodb.com/docs/manual/>>

¹²Node.js: <<https://nodejs.org/>>

¹³Object Document Mapper (ODM): aplicação que faz o mapeamento e abstração dos documentos (registros) no banco de dados para estruturas conhecidas pelas linguagens de programação.

¹⁴Mongoose: <<https://mongoosejs.com/>>

¹⁵Faker.js: <https://fakerjs.dev/>

¹⁶Apache JMeter: <<https://jmeter.apache.org/>> que realizou chamadas para os serviços da Fog.

frequência mínima de 2666 MHz e sistema operacional Linux distribuição Linux Mint 20.3. Além disso, foi usado o serviço ngrok¹⁷ para ser o proxy dos Fog Nodes, expondo as aplicações executadas localmente para a internet e acessíveis para a Cloud.

- Cloud – foram implantados na plataforma Heroku¹⁸ os serviços Retrieve Data e OTS e o banco de dados do OTS foi implantado na plataforma MongoDB Atlas¹⁹. Ambas plataformas são abstrações de Cloud para servidores e bancos de dados e utilizam os recursos da *Amazon Web Services*²⁰ (AWS) para hospedar seus serviços. Dessa maneira, foi selecionado o mesmo *data center* da AWS nos EUA (*us-east-1*), assim é possível reduzir a latência da comunicação entre os serviços e bancos de dados da Cloud.

5.3 Cenários e parâmetros

A fim de avaliar o modelo, nesta pesquisa foram implementados os dois cenários a seguir:

- a) No primeiro cenário foi avaliado o tempo de resposta para operações de leitura e escrita a partir da Edge no banco de dados de um único Fog Node. Como parâmetro, variou-se o número de Edge Devices, progredindo o número de dispositivos de 1 até 30. Além disso, foram realizadas 10 chamadas por Edge Device, de modo que na última iteração foram realizados 30 requisições simultâneas e um total de 300 requisições. Também foi utilizado um atraso aleatório de no máximo 100ms para que o envio das requisições tivesse um comportamento um pouco mais natural. Esse cenário permitiu reproduzir a operação da Smart City referente a interação dos Edge Devices com os Fog Nodes.
- b) Já o segundo cenário simulou a rastreabilidade dos usuários realizada a partir do serviço *Retrieve Data* da Cloud. Utilizou-se como parâmetros o número de Fog Nodes (de 1 a 4) ou uma abordagem centralizada. Em cada Fog Node foi inserido um dataset com 1000 registros dos quais deveriam ser retornados aproximadamente 100 registros na consulta realizada. Desse modo, para a abordagem centralizada a massa de dados variou de 1000 a 4000 registros devendo retornar de 100 a 400 registros, aproximadamente.

Este cenário variou em 3 abordagens e foi executado 10 vezes para cada uma. A primeira é a proposta pelo modelo DASTData: dados distribuídos nos Fog Nodes e a consulta via Cloud realizada de maneira concorrente. A segunda abordagem também utilizou armazenamento distribuído, mas a consulta nos Fog Nodes foi alterada no serviço *Register Session* para ser realizada de maneira sequencial. Por fim, a terceira abordagem utilizou um armazenamento centralizado na Fog, para simular as abordagens que não distribuem os dados pelos Fog Nodes. Cada uma dessas abordagens representa uma implementação

¹⁷ngrok: <<https://ngrok.com/>>

¹⁸Heroku: <<https://www.heroku.com/>>

¹⁹MongoDB Atlas: <<https://www.mongodb.com/atlas/database>>

²⁰Amazon Web Services (AWS): <<https://aws.amazon.com/>>

possível para rastreabilidade de dados distribuídos. Este cenário poderia, por exemplo, ser a consulta dos dados de usuários por meio de um dashboard na Cloud.

5.4 Métricas de Avaliação

As métricas de avaliação utilizadas no primeiro cenário avaliam a performance do modelo, portanto utilizou-se a média, mediana, 90° percentil (P90), 95° percentil (P95), 99° percentil e valores mínimo e máximo da latência em milissegundos. Já no segundo cenário, como a avaliação é comparativa entre abordagens diferentes, se comparou a média da latência em milissegundos para as execuções.

6 RESULTADOS

Nesta seção serão apresentados os resultados obtidos através da metodologia de avaliação apresentada na Seção 5. Além disso, as subseções a seguir fazem referências direta aos cenários descritos anteriormente.

6.1 Cenário A

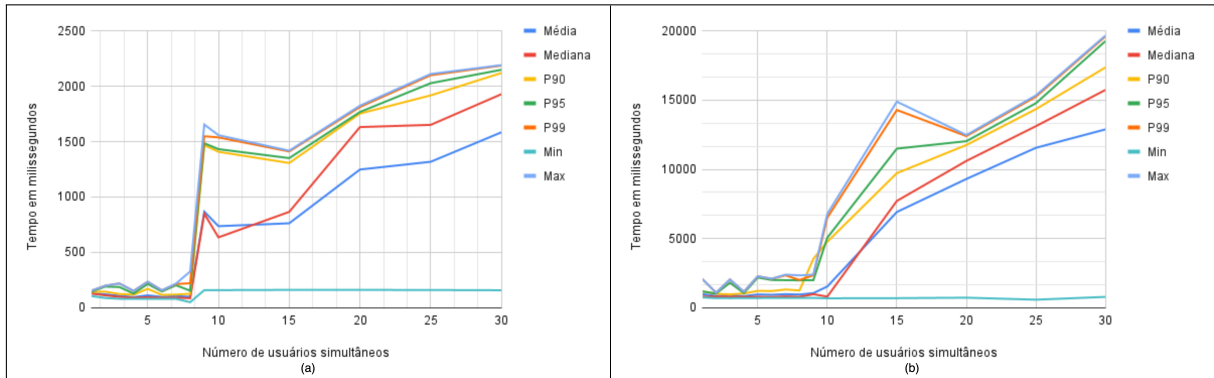
Os resultados apresentados na Figura 10 demonstram que as operações de leitura são mais rápidas do que as de escrita. As médias de latência, por exemplo, mostram que a leitura pode ser até 80% mais rápidas que a escrita. Isso deve-se ao fato de que operações de escrita replicam essas informações em cada instância de banco de dados contida no Fog Node no momento da inclusão da informação. Além disso, como as operações de leitura não comprometem a integridade dos dados elas são realizadas de forma concorrente por cada instância do banco de dados. Outro ponto que observa-se na figura é relacionado ao aumento da latência quando são ultrapassados os oito usuários simultâneos. Isso ocorre porque, como descrito na Subseção 5.2, o processador dos Fog Nodes nesta emulação é um quad-core com 4 núcleos físicos e 4 lógicos.

6.2 Cenário B

Na Figura 11 são apresentados os resultados do cenário de teste que simulou como seria a busca da rastreabilidade do usuário na Cloud (*Retrieve Data* para OTS) e posteriormente a busca dos dados nos Fog Nodes. Analisando este gráfico é possível concluir que a utilização de armazenamento distribuído com consultas concorrentes é mais eficiente do que o armazenamento centralizado. Além disso, também nota-se a vantagem do uso de concorrência na tarefa de consultar os dados distribuídos, pois nesta abordagem não há necessidade de esperar a resposta de um servidor para solicitar os dados ao próximo. Portanto, através dos resultados apresentados neste cenário, observa-se que a abordagem do modelo proposto nesta pesquisa pode ser, apro-

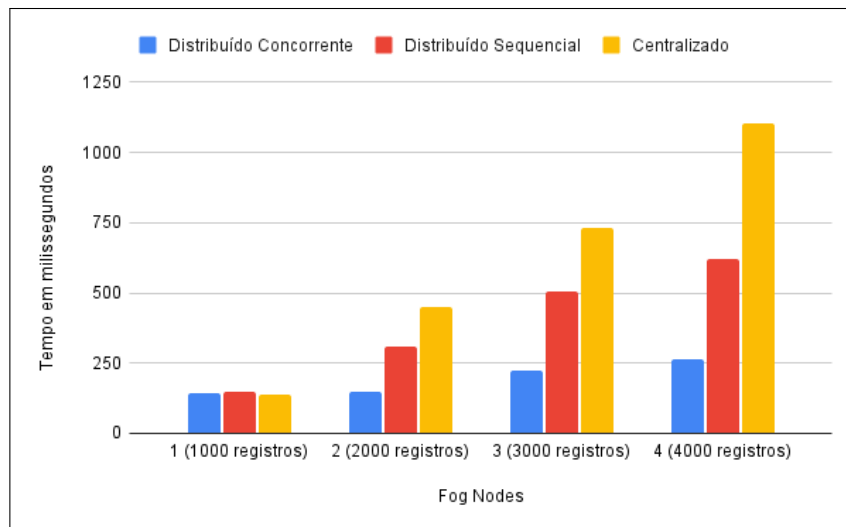
ximadamente, até 74% mais eficiente que em um modelo centralizado e até 53% mais eficiente comparando-se a um modelo distribuído sequencial.

Figura 10: Resultados do Cenário A para operações de leitura (lado a) e escrita (lado b) nos bancos de dados distribuídos pelos Fog Nodes.



Fonte: Elaborada pelo autor.

Figura 11: Resultados do Cenário B para a busca dos dados de um usuário no modelo.



Fonte: Elaborada pelo autor.

6.3 Discussão

A partir dos resultados apresentados é possível discutir alguns pontos da arquitetura relacionados a escalabilidade dos Fog Nodes. Embora o acesso seja descentralizado, ainda assim podem ocorrer problemas relacionados a concentração populacional em determinados Fog Nodes em detrimento de outros. O elevado número de usuários simultâneos mostrou no Cenário A que, a partir de determinado momento, concentrar o acesso em apenas uma base de dados pode ser prejudicial para a latência de resposta do modelo. Por este motivo, uma solução poderia

estar relacionada a utilização sob demanda de subshardings em cada Fog Node para prover escalabilidade horizontal internamente. Neste caso, poderiam ser implementadas mais instâncias de bancos de dados que dividam igualmente a carga de trabalho para resolver operações de escrita e leitura dentro de um único Fog Node.

Outro ponto relevante está relacionado à implementação de MongoDB como principal motor para a avaliação realizada. Talvez alguns dos resultados possuam viés em decorrência do uso de um único SGBD. Portanto, pode ser relevante estudar e implementar o modelo DASTData em outros Bancos de Dados Não Relacionais com suporte a replicação a fim de determinar se os algoritmos próprios do recurso escolhido não estão também colaborando para os pontos de melhoria observados.

7 CONCLUSÃO

Neste trabalho foi apresentado o modelo DASTData que é uma proposta de arquitetura para armazenamento distribuído que possibilita a rastreabilidade dos usuários de Smart Cities. Através de abordagens de alto desempenho como concorrência, replicação e Sharding, esta pesquisa demonstrou as vantagens do DASTData em relação a modelos centralizados e/ou que não utilizem de operações concorrentes. Espera-se que este modelo contribua para a literatura acadêmica como uma possibilidade de arquitetura de armazenamento a ser implantada em Smart Cities.

Embora o modelo atenda aos objetivos do trabalho, os resultados demonstraram algumas limitações relacionadas a alta demanda de diversos Edge Devices aos Fog Nodes concomitantemente. Além disso, o modelo não abrange alguns pontos sobre como implementar a assincronia e os algoritmos de decisão de alternância entre Fog Nodes. Também não foi detalhado sobre como realizar o balanceamento de carga entre Fog Nodes. Portanto, ficam alguns trabalhos futuros para propor melhorias às limitações citadas e também relacionados ao ambiente de emulação. Seria extremamente relevante a realização de testes em proporções de uma cidade a fim de obter-se resultados mais autênticos aos de um funcionamento real. Também realizar o estudo

Referências

ABDELHAFIZ, B. M.; ELHADEF, M. Sharding database for fault tolerance and scalability of data. In: **2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM)**. [S.l.: s.n.], 2021. p. 17–24.

ABREU, D. P. et al. A resilient internet of things architecture for smart cities. **Ann. Telecommun.**, v. 72, n. 1-2, p. 19–30, 2017.

BENHAMIDA, N. et al. Dynamic architecture for collaborative distributed storage of collected data in fog environments. **Wireless Personal Communications**, v. 123, n. 4, p. 3511–3537, abr. 2022.

BONOMI, F. et al. Fog computing and its role in the internet of things. In: **Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing**. New York, NY, USA: Association for Computing Machinery, 2012. (MCC '12), p. 13–16. ISBN 9781450315197. Disponível em: <<https://doi.org/10.1145/2342509.2342513>>.

CHANG, V. An ethical framework for big data and smart cities. **Technological Forecasting and Social Change**, v. 165, p. 120559, 2021. ISSN 0040-1625. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0040162520313858>>.

CHEN, S. et al. A vision of iot: Applications, challenges, and opportunities with china perspective. **IEEE Internet of Things Journal**, v. 1, n. 4, p. 349–359, 2014.

CORBELLINI, A. et al. Persisting big-data: The nosql landscape. **Information Systems**, v. 63, p. 1–23, 2017. ISSN 0306-4379. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0306437916303210>>.

FAROOQI, A. M. et al. A fog computing model for vanet to reduce latency and delay using 5g network in smart city transportation. **Applied Sciences**, v. 12, n. 4, 2022. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/12/4/2083>>.

HAKIRI, A. et al. Publish/subscribe-enabled software defined networking for efficient and scalable iot communications. **IEEE Communications Magazine**, v. 53, n. 9, p. 48–54, 2015.

JAWARNEH, I. M. A. et al. Efficient qos-aware spatial join processing for scalable nosql storage frameworks. **IEEE Transactions on Network and Service Management**, v. 18, n. 2, p. 2437–2449, 2021.

KHATOUN, R.; ZEADALLY, S. Smart cities: Concepts, architectures, research opportunities. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 59, n. 8, p. 46–57, jul. 2016. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/2858789>>.

KUDO, T. Fog computing with distributed database. In: **2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)**. [S.l.: s.n.], 2018. p. 623–630.

LAI, C. S. et al. A review of technical standards for smart cities. **Clean Technologies**, v. 2, n. 3, p. 290–310, 2020. ISSN 2571-8797. Disponível em: <<https://www.mdpi.com/2571-8797/2/3/19>>.

LOMOTÉY, R. K.; PRY, J. C.; CHAI, C. Traceability and visual analytics for the Internet-of-Things (IoT) architecture. **World Wide Web**, v. 21, n. 1, p. 7–32, 2018.

MELL, P.; GRANCE, T. **The NIST Definition of Cloud Computing**. [S.l.]: Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2011.

NAEEM, R. Z. et al. **Fog computing in internet of things: Practical applications and future directions**. Springer Science and Business Media LLC, 2019. 1236–1262 p. Disponível em: <<http://dx.doi.org/10.1007/s12083-019-00728-0>>.

PATEL, K. K.; PATEL, S. M. et al. Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges. **International journal of engineering science and computing**, v. 6, n. 5, 2016.

SASUBILLI, S. M.; KUMAR, A.; DUTT, V. Improving health care by help of internet of things and bigdata analytics and cloud computing. In: **2020 International Conference on Advances in Computing and Communication Engineering (ICACCE)**. [S.l.: s.n.], 2020. p. 1–4.

SHWE, H. Y.; JET, T. K.; CHONG, P. H. J. An iot-oriented data storage framework in smart city applications. In: **2016 International Conference on Information and Communication Technology Convergence (ICTC)**. [S.l.: s.n.], 2016. p. 106–108.

SINAEPOURFARD, A. et al. Data preservation through fog-to-cloud (f2c) data management in smart cities. In: **2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)**. [S.l.: s.n.], 2018. p. 1–9.

VILELA, P. H. et al. Looking at fog computing for e-health through the lens of deployment challenges and applications. **Sensors**, v. 20, n. 9, 2020. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/20/9/2553>>.

ZHANG, C. Design and application of fog computing and internet of things service platform for smart city. **Future Generation Computer Systems**, v. 112, p. 630–640, 2020. ISSN 0167-739X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167739X19331024>>.