UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Renata Caroline Francisco dos Santos

Modelo de Disponibilidade para Suporte ao Planejamento de Infraestruturas Computacionais Baseadas em *Fog* e *Edge Computing*

Recife, 2019

RENATA CAROLINE FRANCISCO DOS SANTOS

MODELO DE DISPONIBILIDADE PARA SUPORTE AO PLANEJAMENTO DE INFRAESTRUTURAS COMPUTACIONAIS BASEADAS EM *FOG* E *EDGE COMPUTING*

Monografia apresentada ao Centro de Informática (CIN) da Universidade Federal de Pernambuco (UFPE), como requisito parcial para conclusão do Curso de Engenharia da Computação, orientada pelo professor Paulo Romero Martins Maciel.

Recife, 2019

UNIVERSIDADE FEDERAL DE PERNAMBUCO

RENATA CAROLINE FRANCISCO DOS SANTOS

Monografia submetida ao corpo docente da Universidade Federal de Pernambuco, defendida e aprovada em _____ de _____ de _____.

Banca Examinadora:

Orientador(a): Prof. Dr. Paulo Romero Martins Maciel

Prof. Dr. Eduardo Antônio Guimarães Tavares

Recife, 2019

Dedico este trabalho aos meus pais. Sem os seus incentivos e suas palavras de apoio não teria conseguido.

Agradecimentos

Primeiramente, quero agradecer a Deus por ter me dado saúde, foco e força para superar as dificuldades. E que permitiu que tudo isso acontecesse na minha vida, não só como universitária, mas também como pessoa que me tornei. Também agradeço aos meus pais, pelo amor, incentivo e apoio incondicional. Eles são minha maior fonte de inspiração e é por eles que continuo enfrentando todos os dias a batalha para conseguir ser alguém na vida. Sou grata também a minha família pela força e compreensão nos momentos que estive ausente para cumprir com as demandas da universidade.

Além disso, agradeço aos meus amigos, companheiros de estágio e trabalho que fizeram parte da minha formação e que vão continuar presentes em minha vida com certeza. Também agradeço aos meus colegas da universidade que estiveram comigo nas aulas, nos projetos, nas provas e em todos os momentos que vivi durante a graduação.

Não posso deixar de agradecer ao meu orientador e seu aluno de doutorado, Paulo Maciel e Paulo Pereira, respectivamente, pelo empenho dedicado à elaboração deste trabalho. Em geral, quero agradecer todos os professores por me proporcionarem todo o conhecimento que adquiri nesse processo de formação profissional.

Um agradecimento a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

Resumo

A *Fog Computing* é o novo paradigma que tende aumentar o poder computacional entre a nuvem e o dispositivo, possibilitando maior velocidade e melhorando o tempo de resposta dos serviços de computação. Semelhante à *Fog*, a *Edge* garante processamento mais rápido dos dados. De modo geral, a *Edge Computing* tem por objetivo acabar com os problemas de grande consumo de banda para tráfego de dados e processamento na nuvem [1]. Este trabalho faz uma avaliação de disponibilidade considerando esses dois ambientes, *Fog* e *Edge Computing*. Faz a medição da disponibilidade, gera modelos e por fim analisa os resultados.

Palavras-chave: *Fog Computing*, *Edge Computing*, *Dependabilidade*, *Disponibilidade*.

Abstract

Fog Computing is the new paradigm that tends to increase computing power between the cloud and the device, enabling greater speed and improving the response time of computing services. Similar to Fog, Edge ensures faster data processing. Overall, Edge Computing aims to address the high bandwidth issues for data traffic and cloud processing [1]. This paper makes an availability assessment considering these two environments, Fog and Edge Computing. It measures availability, generates models and finally analyzes the results.

Keywords: Fog Computing, Edge Computing, Dependability, Availability.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

MTTF	Mean Time To Failure
MTTR	Mean Time To Repair
SPN	Stochastic Petri Nets
RBD	Reliability Block Diagram
CTMC	Continuous-time Markov Chains

Sumário

1 Introdução

Neste capítulo são apresentados as principais motivações para realização deste trabalho. Na sequência, são apresentados o contexto, a motivação, a justificativa e os objetivos. Para finalizar, os trabalhos relacionados com a pesquisa são apresentados.

1.1 Contexto

Estamos vivendo de constantes inovações. Inteligência artificial, Internet das Coisas (IoT), Big Data e redes sem fio de alta velocidade trouxeram mudanças além do que poderíamos imaginar. Como resultado, usamos cada vez mais dispositivos para decisões de negócios, com o apoio da análise de dados [2].

Cloud Computing ou computação em nuvem é a entrega da computação como um serviço ao invés de um produto, onde recursos compartilhados, software e informações são fornecidas, permitindo o acesso através de qualquer computador, tablet ou celular conectado à Internet [3]. Pode-se dizer que ela transfere a responsabilidade de processamento e armazenamento de informação para servidores hospedados em datacenters e que podem ser acessados via Internet [4].

Enviar dados para uma determinada plataforma exige, muitas vezes, uma grande largura de banda, principalmente quando os dados são multimídias. Para algumas arquiteturas, as mídias digitais devem estar acessíveis a qualquer hora do dia ou da noite. O grande desafio é identificar uma configuração que atenda aos requisitos de desempenho, disponibilidade e custo, principalmente porque requisitos como mobilidade, localidade e baixa latência não são priorizados nos ambientes de *Cloud Computing*. O conceito de *Fog Computing* define uma arquitetura que estende a capacidade computacional e o armazenamento da nuvem para as camadas de acesso da rede, permitindo que os dados sejam analisados e transformados em informações ou em ações antes de serem simplesmente transmitidos [5]. Isso faz com que a latência da

comunicação seja reduzida consideravelmente.

Fog Computing nada mais é que uma extensão da *Cloud Computing*, pois ela foi concebida com foco nos dispositivos que estão na borda da rede, mais próxima do usuário e assim mais descentralizada [6]. Na opinião de Severiano Leão Macedo, *Cisco Digital Transformation Advisor*, *Fog Computing* ajuda a reduzir os custos das aplicações de IoT. Também endereça questões como performance e confiabilidade do sistema e segurança [5].

O conceito de *Fog Computing* descreve algo muito semelhante em relação à *Edge Computing*. A *Edge Computing* refere-se às tecnologias que permitem que a computação seja executada na borda da rede [7]. Ou seja, o processameto dos dados sobre a aplicação é diretamente no local necessário, não tem a necessidade destas informações serem trafegadas para a nuvem. Essa característica da *Edge* permite que o processamento seja mais próximo do usuário final, melhorando assim a velocidade das respostas.

A principal diferença entre *Edge* e *Fog Computing* reside o local onde decorre o processamento [8]. A *Edge* ocorre diretamente nos dispositivos, enquanto que a *Fog* é uma camada intermediária entre a *Edge* e a *Cloud Computing*.

1.2 Motivação

Imagine que existe um drone que auxilie no reconhecimento de indivíduos suspeitos de crimes e que de imediato envia essa informação para o piloto que controla o drone e este aciona a central de segurança da cidade sobre o suspeito. Se tal informação tiver que ser processada e enviada para a *Cloud* e depois retornada para o piloto, háverá uma perda no tempo de resposta que pode ser crucial. Também conhecida como Computação em Névoa, a *Fog Computing* e também a *Edge Computing* tratam de reduzir a distância que a informação precisa atravessar e redução no tempo de envio para a nuvem.

1.3 Justificativa

Disponibilidade é um dos requisitos mais importantes para as empresas que buscam oferecer serviços sem interrupções, como por exemplo provedores de serviços na nuvem. Mesmo com o avanço da tecnologia, é difícil garantir que um serviço não tenha falhas e passe algum tempo fora de execução. Interrupções em qualquer tipo de serviço podem vir a compremeter o funcionamento do sistema, e dessa forma, ocasionar insatisfação dos clientes. Essa insatisfação pode levar a perdas financeiras irreparáveis. A confiabilidade desse tipo de sistema pode ser afetada por falhas ou atualizações de software, manutenções planejadas e troca de equipamentos [9]. Serviços na nuvem que apresentem, segundo Maria Clara dos Santos Bezerra em [10], um alto ou constante índice de falha, confiabilidade e segurança poderão oferecer uma experiência negativa ao usuário, acarretando em prejuízos ao provedor do serviço. O usuário possui diversas maneiras de criticar um serviço ruim na Internet e ainda tem a possibilidade de utilizar serviços concorrentes em questão de segundos. Assim, um sistema na nuvem está vulnerável a forte rejeição dos usuários, caso não ofereça um serviço de qualidade e de alta disponibilidade.

Este trabalho visa apresentar um modelo de disponibilidade para dar suporte ao planejamento de infraestruturas computacionais baseadas em *Fog* e *Edge Computing*.

1.4 Objetivo

1.4.1 Objetivo Geral

Proposição de um modelo de disponibilidade para planejamento de uma infraestrutura de *Fog Computing* com suporte a um serviço de reconhecimento facial. Essa solução integrada permitirá o planejamento de infraestruturas de *Fog* de acordo com os requisitos estabelecidos com os usuários. A abordagem deve planejar um ambiente de *Fog* e *Edge Computing* para um sistema de reconhecimento facial considerando aspectos de disponibilidade.

1.4.2 Objetivos Específicos

Considerando o desenvolvimento do trabalho e o objetivo geral apresentado, destacam-se os seguintes objetivos específicos:

- Construir modelo de disponibilidade para arquiteturas de *Fog Computing* e *Edge Computing*, com foco em serviços de reconhecimento facial;
- Validar o modelo de disponibilidade da arquitetura mencionada;
- Executar análise de sensibilidade para determinar quais componentes na arquitetura base possuem um maior impacto na disponibilidade do ambiente.

1.5 Trabalhos Relacionados

Um dos principais focos em praticamente todas as áreas da Ciência da Computação, a dependabilidade é a propriedade que define a capacidade dos sistemas computacionais de prestar um serviço que se pode confiar. Devido à sua importância ao usuário e ao ambiente de execução do software, o interesse nesse assunto tem sido alvo de diversas pesquisas que envolvem todos os ciclos de desenvolvimento do software: desde a análise que antecede a implementação até o teste do software.

Em paralelo a esse tema, a *Fog Computing* é a grande tendência tecnológica atual. Ela propõe que a análise das informações esteja mais perto da borda dos dispositivos, diminuindo assim o tempo para resposta. Por isso, várias pesquisas estão sendo feitas sobre o tema.

Neste seção são apresentadas as descrições dos principais trabalhos relacionados ao desenvolvimento dessa pesquisa. Pesquisas recentes relacionadas à *Fog Computing*, *Edge Computing* e Dependabilidade identificadas na literatura e que possui proximidade com o objetivo de pesquisa proposta são destacadas.

Em [11], é realizada uma revisão sobre a *Fog Computing*. O conceito, as características e áreas de atuação são apresentadas nesse trabalho.

Uma proposta de arquitetura de vigilância inteligente que aproveita as vantagens da *Edge Computing* e *Fog Computing* para atingir a meta de processamento de vídeo em tempo real para detecção de humanos é feita em [12]. Nessa pesquisa, os autores usaram um laptop representando o nó da *Fog Computing* e uma Raspberry PI representando o nó da *Edge Computing*.

Em [13], os autores implementam uma *Fog Computing* usando uma Raspberry para medir a temperatura do ambiente. Os dados da temperatura são previstos a partir de uma série temporal e então são enviados para a nuvem para que possa serem exibidos em uma página da web. Já no artigo [14], é feita uma comparação do consumo de energia de aplicativos usando *Data Centers* (DCs) centralizados na *Cloud Computing* com aplicativos usando nano *Data Centers* (nDCs) na *Fog Computing*.

Relacionado à dependabilidade, em [15], o autor desenvolve uma estrutura de rede confiável. Nessa pesquisa, ele investiga as características de confiabilidade da rede usando dados SNMP e Syslog para atender as causas das falhas e seu impacto.

Modelos de desempenho, disponibilidade e custo orientada à capacidade para arquiteturas de *Cloud Computing*, com foco em serviços de vídeo sob demanda (VoD) são construídos na pesquisa realizada em [16]. Já em [17], os autores propõem uma arquitetura para representar o comportamento de um sistema de monitoramento da saúde. Esse monitoramento faz o uso de dispositivos de *Fog* (como a Raspberry) e infraestruturas de *Cloud* para processar e armazenar dados vitais dos pacientes. Para analisar como as falhas afetam a disponibilidade do sistema, é proposto um modelo analítico usando Redes de Petri Estocásticas (SPN) e Diagrama de Blocos de Confiabilidade (RBD). Semelhante ao trabalho anterior apresentado, em [18] a dependabilidade para diferentes infraestruturas em nuvem são avaliadas e representadas por Redes de Petri Estocáticas (SPNs) e Diagrama de Blocos de Confiabilidade.

	Fog/Edge/Cloud Computing	Análise de Dependabilidade e Sensibilidade
[11]	-	Não
[12]	*Fog*, *Cloud* e *Edge*	Não
[13]	*Fog*	Não
[14]	*Fog* e *Cloud*	Não
[15]	*Fog* e *Cloud*	Dependabilidade
[16]	*Cloud*	Dependabilidade
[17]	*Fog*, *Cloud* e *Edge*	Dependabilidade
[18]	*Cloud*	Dependabilidade
Proposta	*Fog* e *Edge*	Dependabilidade e Sensibilidade

Tabela 1 – Comparação dos trabalhos relacionados

2 Fundamentação Teórica

Este capítulo apresenta uma introdução sobre *Fog Computing* e *Edge Computing*. Em seguida, apresenta os conceitos básicos sobre dependabilidade. Em seguida, traz uma breve explicação sobre o Diagrama de Bloco de Confiabilidade (RBD), uma das técnicas de modelagem usada para avaliação de dependabilidade, inclusive esta será usada neste trabalho.

2.1 *Fog Computing*

O termo *Fog Computing* foi adotado pela *Cisco Systems* como um novo paradigma [19]. *Fog Computing* é um modelo de computação onde porções da *Cloud Computing* são deslocadas para a borda da rede formando um ambiente de computação local. Esse modelo de computação distribui os recursos e os serviços de processamento, comunicação, controle e armazenamento mais perto dos dispositivos [5]. Em [20], é mostrada a arquitetura da *Fog* como mostra a Figura 1 abaixo.

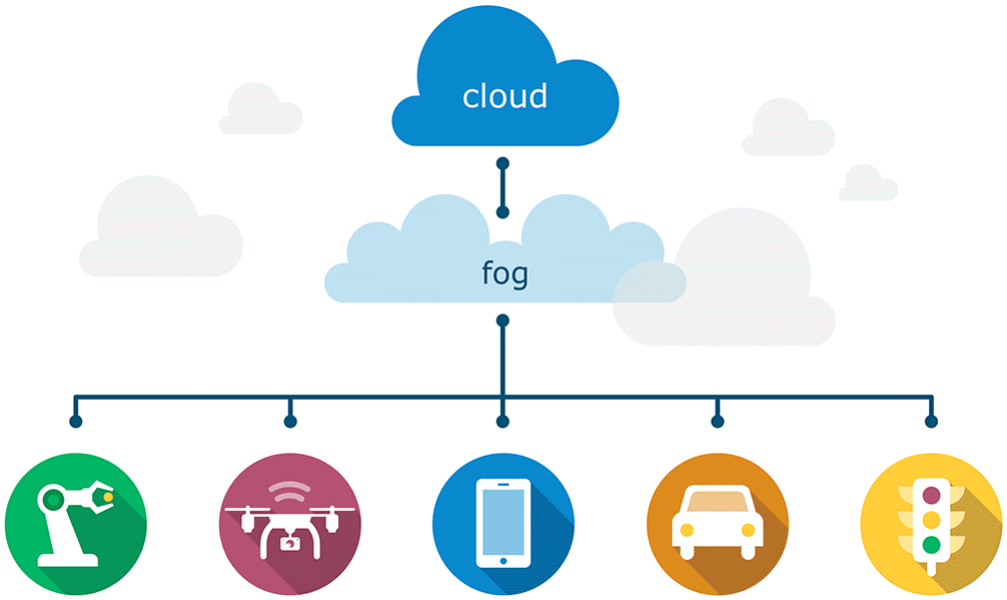


Figura 1 – Arquitetura da *Fog Computing*

A *Fog Computing* apresenta uma nova arquitetura que "leva proces-

samento para os dados", enquanto a nuvem "leva os dados para o processamento"[11]. Vantagens como processamento em tempo real e rápida escalabilidade são algumas das vantagens oferecidas por essa nova arquitetura. Suas principais características são:

- **Heterogeneidade:** A *Fog Computing* é uma plataforma virtualizada que oferece serviços computacionais, de rede e de armazenamento entre a computação em nuvem e dispositivos finais de diferentes tipos e formas [11].
- **Distribuição geográfica:** A computação em *Fog* possui uma implementação amplamente distribuída para oferecer serviços de alta qualidade para dispositivos finais móveis e fixos [11].
- **Proximidade dos dados para os usuários finais:** Para melhorar as preocupações com latência e acesso aos dados, os serviços estão localizados mais perto do usuário final [19].
- **Organização hierárquica:** A *Fog Computing* segue uma arquitetura de várias camadas para oferecer suporte a baixa latência e escalabilidade [19].

Como mostrado na Figura 1, a *Fog* funciona como um link entre os dispositivos e a *Cloud*. Segundo [21], antes de transferir os dados para a *Cloud*, a *Fog* deve ser capaz de decidir qual conteúdo deve ser enviado, qual o formato dos dados e quando enviá-los.

2.2 *Edge Computing*

Como mencionado no Capítulo 1 na Seção Contexto deste trabalho, a *Edge Computing* é um modelo de computação que permite armazenar e processar dados mais perto do usuário final. A *Edge Computing* surgiu para lidar com as demandas por tráfego e processamento de dados, que têm se tornado cada vez mais volumosas e crescentes [22].

Para diminuir o tráfego de dados enviados e a ajudar a reduzir a largura de banda necessária para as comunicações entre os aparelhos e a rede [22], é realizada uma triagem das informações. Isso tem se mostrado cada vez mais essencial, pois a largura de banda por usuário é um gargalo para quem precisa transmitir grandes volumes de dados [22]. Em [23] é mostrado onde a *Edge* está localizada, como mostra a Figura 2 abaixo.

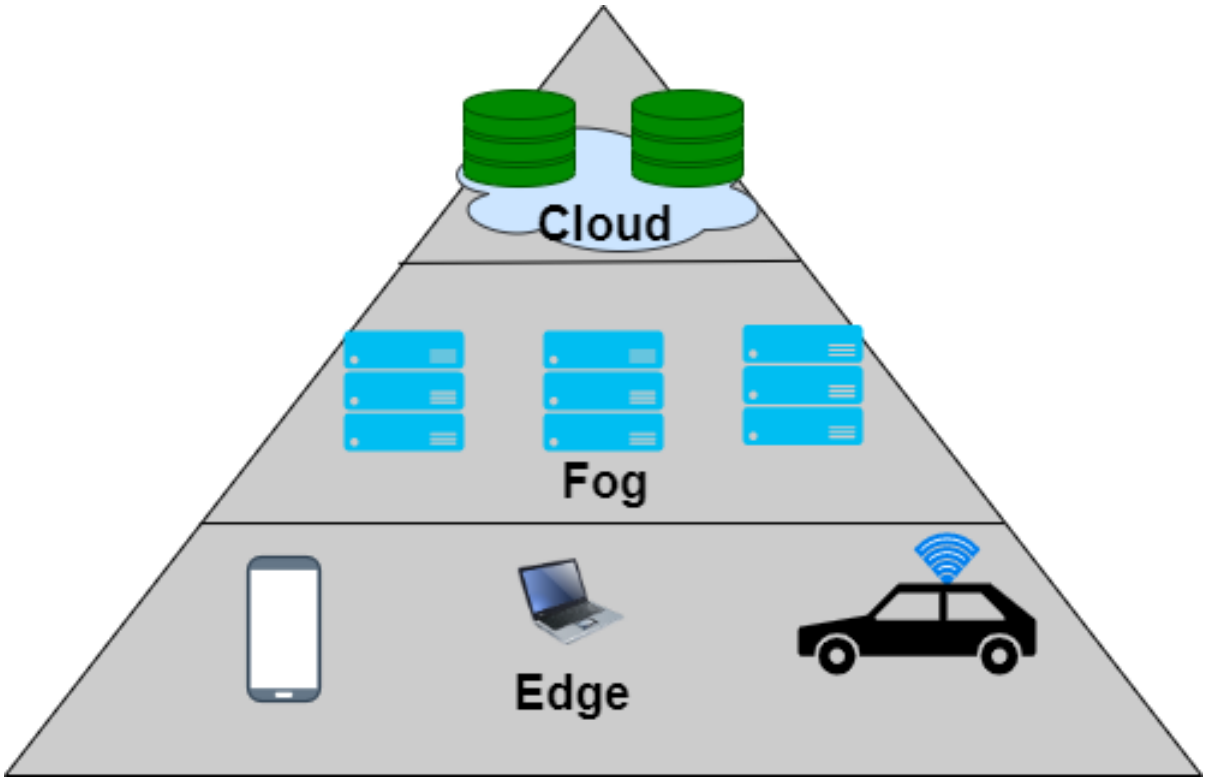


Figura 2 – *Edge Computing*

De forma geral, a *Edge Computing* surgiu como uma solução para os casos em que os dispositivos de IoT (Internet das Coisas) estejam em situações de pouca conexão. As principais vantagens desse modelo de computação são:

- Maior desempenho de processamento, pois o processamento dos dados será feito próximo à sua fonte;
- Menor custo de transmissão de dados;
- Aumento da disponibilidade do serviço;
- Redução da latência, pois os dados não precisam chegar até a nuvem.

2.3 Conceitos Básicos Sobre Dependabilidade

A dependabilidade de um sistema de computação é a capacidade de fornecer um serviço que possa ser justificadamente confiável [24], ou ainda, pode ser definida como confiança justificável de que o sistema executará ações especificadas ou fornecerá resultados específicos de maneira confiável e oportuna [25]. Medidas como a disponibilidade, confiabilidade e segurança são englobadas no conceito de dependabilidade.

Em [9], é apresentada uma taxonomia da dependabilidade, que pode ser observada na Figura 3 e os elementos são mencionados e explicados logo a seguir.

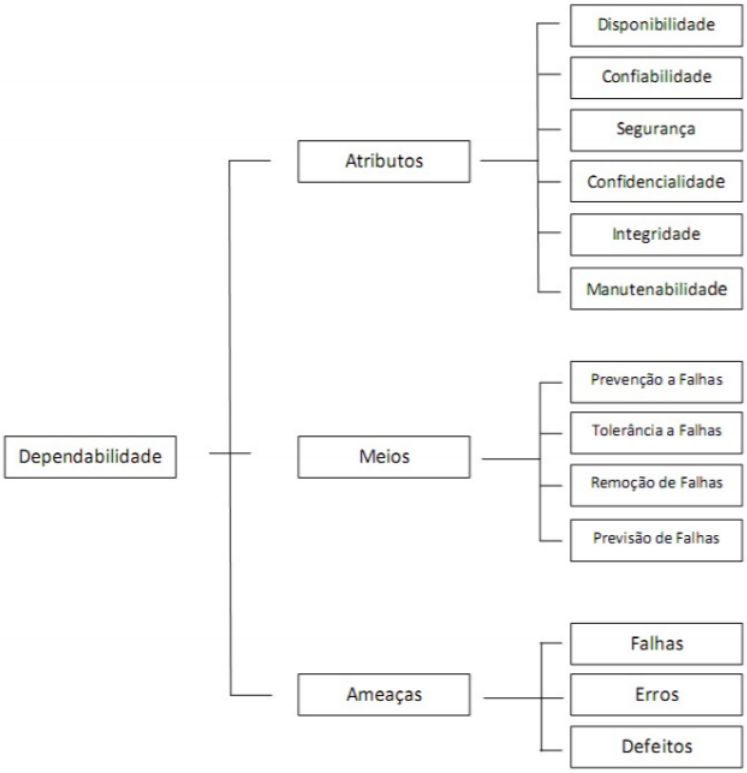


Figura 3 – Taxonomia da Dependabilidade

2.3.1 Elementos da Dependabilidade

A dependabilidade de um sistema é dividida em três partes:

- Atributos

- Meios
- Ameaças

2.3.1.1 Atributos

Os atributos indicam a qualidade do sistema. Eles podem ser avaliados para determinar a confiabilidade do sistema usando medidas quantitativas ou qualitativas. Em [24] é definido os seguintes atributos:

- Disponibilidade
- Confiabilidade
- Segurança
- Confidencialidade
- Integridade
- Manutenabilidade

Disponibilidade

A disponibilidade do sistema é uma métrica que mede a probabilidade de um sistema falhar ou sofrer uma ação de reparo quando precisar ser usado. Ela é importante porque uma empresa, por exemplo, pode ter vários prejuízos graves quando seus dados se tornam indisponíveis [24].

A disponibilidade é usada para medir e investigar a eficiência dos processos e das ferramentas usados pela equipe de manutenção e como eles podem ser aprimorados. Sendo assim, pode-se dizer que a disponibilidade é impactada pela confiabilidade e manutenabilidade, atributos que serão detalhadas mais em breve.

A representação mais simples da disponibilidade 2.1 é uma proporção entre o tempo de atividade (*uptime*) dividido pela soma do tempo de atividade e o tempo de inatividade (*downtime*):

$$A = \frac{uptime}{uptime + downtime} \tag{2.1}$$

A equação da disponibilidade pode ser representada pela relação entre MTTF e MTTR, em que:

- **MTTF** - é o tempo médio para ocorrer falhas o sistema

$$MTTF = \int_0^\infty R(t)dt \tag{2.2}$$

- **MTTR** - é o tempo médio para reparar o sistema depois de uma falha

$$MTTR = MTTF \times \frac{UA}{A} \tag{2.3}$$

Onde UA é a indisponibilidade do sistema e é representada pela seguinte equação:

$$UA = 1 - A \tag{2.4}$$

A disponibilidade pode ser enquadrada em três classes, de acordo com a faixa de valores dada pela probabilidade, a saber: disponibilidade básica, alta disponibilidade e disponibilidade contínua [26].

Sistemas que não possuem nenhum processo complexo de software ou hardware apresentam uma disponibilidade de 99 a 99,9%, eles são considerados sistemas de disponibilidade básica. Já os sistemas que possuem mecanismos de detecção, recuperação e mascaramento de falhas [27] apresentam disponibilidade entre 99,99 a 99,999%, neste caso se enquadram na classe de alta disponibilidade. Por fim, sistemas que não tem indisponibilidade planejada nem não planejada e possuem abordagem ou desenho para atingir 100% apresentam disponibilidade contínua [28].

Confiabilidade

A confiabilidade de um sistema é a probabilidade (P) de que esse sistema execute a sua funcão, de modo satisfatório, sem a ocorrência de

defeitos, por um determinado período de tempo (T) [29] e pode ser vista através da Equação 2.5:

$$R(t) = P(T > t), t \geq 0 \tag{2.5}$$

A confiabilidade não pode ser confundida com a disponibilidade. Apesar desses dois atributos não serem excludentes, as técnicas que implementam cada um podem ser bem diferentes [26].

Segurança

Segurança é a probabilidade de um serviço ser operacional e executar sua função corretamente, ou descontinuar suas funções de forma a não provocar dano a outros serviços ou pessoas que dele dependam [26].

Confidencialidade

A confidencialidade pode ser entendida como a ausência de divulgação não autorizada de informação [9]. Ou seja, significa garantir que a informação não será conhecida por pessoas que não estejam autorizadas para tal [30].

Integridade

A integridade garante que a informação é autêntica, isto é, que ela não foi alterada e sua fonte é segura. É importante manter a integridade dos dados para que os sistemas operem corretamente graças a sistemas com dados adequados [24].

Manutenabilidade

A manutenabilidade significa realizar facilmente a manutenção de um sistema com defeito. Pode-se dizer que é a probabilidade que este sistema seja restaurado a um estado operacional dentro de um determinado período de tempo [26].

2.3.1.2 Meios

Para alcançar a dependabilidade, um conjunto de técnicas deve ser empregado. Até agora, quatro técnicas foram identificadas:

- Prevenção de Falhas
- Tolerância a Falhas
- Remoção de Falhas
- Previsão de Falhas

Uma breve descrição dessas técnicas é realizada nesta seção.

Prevenção a Falhas

Impede a ocorrência ou introdução de falhas. Envolve a seleção de metodologias de projetos de tecnologias adequadas para os seus componentes [26]. Quando alguma falha é identificada e solucionada, a prevenção garante que elas não voltem a ocorrer.

A prevenção de falhas ocorre durante o desenvolvimento [31]. Por isso, é preciso que todas as pessoas que estejam envolvidas no desenvolvimento estejam conscientes da sua importância.

Tolerância a Falhas

Fornece o serviço esperado mesmo na presença de falhas [26]. Embora esse serviço possa estar em um nível degradado, a tolerância a falhas fornece mecanismos que permitirão que o sistema ainda ofereça o serviço desejado.

Remoção de Falhas

A remoção de falhas é dividida em duas subcategorias: remoção durante o desenvolvimento e remoção durante o uso de um sistema. A remoção durante o desenvolvimento requer verificação para que as falhas possam ser detectadas e removidas antes que o sistema seja colocado em produção.

Previsão de Falhas

É uma estimativa sobre a presença de falhas e sobre as consequências causadas por elas [26]. Essa técnica prevê prováveis falhas para que possam ser removidas ou que seus efeitos possam ser contornados.

2.3.1.3 Ameaças

Ameaças são elementos que podem afetar um sistema e causar uma queda na confiabilidade. Isto acontece quando um serviço falha devido ou a uma não conformidade com a especificação funcional, ou a especificação não descreve adequadamente a função do sistema. São estes as falhas, os erros e os defeitos [32].

Uma **falha** é uma instância no tempo em que um sistema exibe um comportamento contrário à sua especificação. Um **erro** acontece em tempo de execução quando alguma parte do sistema entra em um estado inesperado devido à ativação de uma falha. Já um **defeito** pode ser definido como a manifestação de eventos que ocorre quando o sistema desvia do serviço correto. Ou seja, defeitos ocorrem quando erros são propagados dentro do sistema [32].

Em [32], é apresentada a relação entre falhas, erros e defeitos, que pode ser observada na Figura 4.

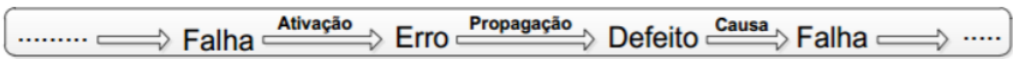


Figura 4 – Relação entre falhas, erros e defeitos

Como regra geral, uma falha, quando ativada, pode levar a um erro e esse erro quando propagado pode levar a um defeito.

2.3.2 Diagramas de Bloco de Confiabilidade

Diagrama de bloco de confiabilidade (RBD) foi a primeira técnica apresentada para determinar a confiabilidade de um sistema [9]. Esse diagrama representa graficamente como os componentes de um sistema são conectados em termos de confiabilidade.

No modelo de diagrama de bloco, os componentes são representados como blocos e são combinados com outros blocos, isto é, outros componentes em séries, paralelos e/ou configurações k-out-of-n [33]. No modelo em série, todos os componentes devem estar funcionando normalmente para que todo o sistema funcione, se um deles falhar todo o serviço falha. Já no modelo em paralelo é preciso ter pelo menos um componente funcionando para que todo o sistema funcione.

As Figuras 5 e 6 representam RBD's em série e em paralelo, respectivamente.

Figura 5 – Série

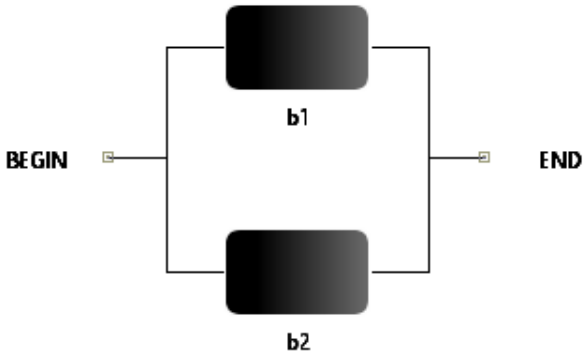


Figura 6 – Paralelo

Sejam $R_1(t)$, $R_2(t)$ e $R_3(t)$ as confiabilidades dos blocos b1, b2 e b3, respectivamente. A confiabilidade desses três blocos conectados em série é obtida pela Equação 2.6.

$$R_s(t) = R_1(t) \times R_2(t) \times R_3(t) \tag{2.6}$$

A confiabilidade do modelo em série, considerando um sistema com n blocos, é dada através da Equação 2.7.

$$R_s(t) = \prod_{i=1}^{n} R_i(t) \tag{2.7}$$

Sejam $R_1(t)$ e $R_2(t)$ as confiabilidade dos blocos b1 e b2, respectivamente. A confiabilidade desses dois blocos conectados em paralelo é obtida pela equação 2.8.

$$R_p(t) = 1 - \prod_{i=1}^{2} (1 - R_i(t)) \tag{2.8}$$

De maneira geral, a confiabilidade do modelo em paralelo, considerando um sistema com n blocos, é dada pela Equação 2.9.

$$R_p(t) = 1 - \prod_{i=1}^{n} (1 - R_i(t)) \tag{2.9}$$

2.4 Considerações Finais

Neste capítulo foram apresentados os conceitos básicos sobre a *Fog* e a *Edge Computing*. Em seguida, conceitos básicos sobre a Dependabilidade foram apresentados, incluindo a técnica de Diagrama de Bloco de Confiabilidade (RBD) para avaliação da confiabilidade.

3 Metodologia

A metodologia de apoio utilizada neste trabalho consiste em duas macro atividade: Estudo preliminar e Avaliação. O estudo preliminar está dividido em três atividades: estudo do sistema, definição de parâmetros e métricas, assim como geração de modelo. Já a avaliação está dividida em: geração de modelo, obtenção de valores de entrada para o modelo e avaliação dos resultados.

3.1 Estudo Preliminar

Esta seção apresenta as três atividades que constituem o estudo preliminar: estudo do sistema, definição de parâmetros e métricas e geração de modelo. O objetivo é apresentar os resultados aqui gerados necessários para a implementação da etapa posterior de avaliação.

- **Estudo do sistema - base para testes:** para planejar uma infraestrutura de *Fog* e *Edge Computing* que atenda a determinados requisitos de um ambiente de reconhecimento facial, é necessário ter um entendimento do funcionamento do sistema, identificando os seus principais componentes, efetuando um levantamento das principais soluções existentes, aplicações e funcionalidades, com o objetivo de delimitar a infraestrutura a ser planejada. A compreensão do sistema requer grande atenção e cuidados especiais por parte do avaliador, para assim, evitar erros de interpretação e comprometimento das demais etapas da metodologia. Essa etapa é essencial, pois possibilita o conhecimento das técnicas que poderão ser adotadas, adaptadas ou que devem ser criadas.

- **Definição de parâmetros e métricas:** o próximo passo é definir os parâmetros e métricas a serem avaliados no sistema que está sendo planejado, pois isso irá afetar diretamente nos tipos de modelos que serão criados mais adiante. Nessa etapa, o avaliador deve definir quais

cenários e métricas são de interesse, focando principalmente naqueles que possuem maior influência nos níveis de qualidade do serviço oferecido.

- **Geração de modelo:** com os parâmetros e métricas estabelecidos, possíveis lacunas sobre o funcionamento do sistema identificadas e contornadas, assim como o foco da avaliação estabelecido, é possível propor modelos de alto nível para obtenção de resultados preliminares do sistema. Dada uma infraestrutura de *Fog Computing*, faz-se necessário criar uma visão geral do sistema para permitir a criação de modelos de alto nível. Dependendo do tipo de métrica a ser avaliada, pode ser adotado formalismos como SPN, RBD ou CTMC.

3.2 Avaliação

Esta seção de avaliação está dividida em duas atividades: obter valores de entrada para o modelo e a avaliação dos cenários. Temos como objetivo utilizar todo o conhecimento adquirido e material desenvolvido nas etapas anteriores para efetivamente avaliar o sistema.

- **Obter valores de entrada para os modelos:** com base no entendimento do sistema, a etapa para obter os valores de entrada para o modelo de disponibilidade é a atividade de coleta, que pode ser através de experimentos. Nesta etapa, scripts de monitoramento serão criados para coleta de valores específicos a fim de alimentar o modelo. Para compreendermos a utilização dos recursos e identificarmos pontos de falha no ambiente serão criados scripts de monitoramneto de recursos e um injetor de falha.

- **Avaliação:** a avaliação do modelo é a atividade que compara os valores de saída com um valor de referência predefinido através de SLAs ou da expectativa do usuário final, ou de administradores de sistema. Quando o valor computado é satisfatório, o processo segue para a próxima etapa. No entanto, se ao menos uma métrica de interesse não alcançou um nível satisfatório, o processo deve reiniciar na atividade de geração de

modelo para que seja feito as alterações necessárias e, dessa maneira, o modelo seja reavaliado.

4 Modelos

Este capítulo apresenta os modelos criados para a avaliação da disponibilidade das infraestruturas de *Fog* e *Edge Computing*. Inicilamente é apresentada uma modelagem hierárquica que representa as infraestruturas e os seus componentes. Posteriormente, por meio dos modelos gerados foi possível obter as fórmulas importantes para calcular as métricas de disponibilidade.

4.1 Modelo de Disponibilidade das Infraestruturas de *Fog* e *Edge Computing*

Esta seção apresenta os modelos propostos para as infraestruturas de *Fog* e *Edge*. Foi adotada uma abordagem para a representação das infraestruturas e avaliação da disponibilidade do sistema por meio de uma modelagem hierárquica.

Por representar o relacionamento entre os componentes que fazem parte de um serviço qualquer, o modelo RBD foi adotado. Esse modelo indica como o funcionamento dos componentes de um sistema afeta a operação do sistema como um todo [34]. Além disso, é possível realizar com esse modelo RBD o cálculo de métricas da dependabilidade, como a disponibilidade [18].

As Figuras 7 e 8 representam os modelos de alto nível RBD das infraestruturas de *Fog* e *Edge*, respectivamente.

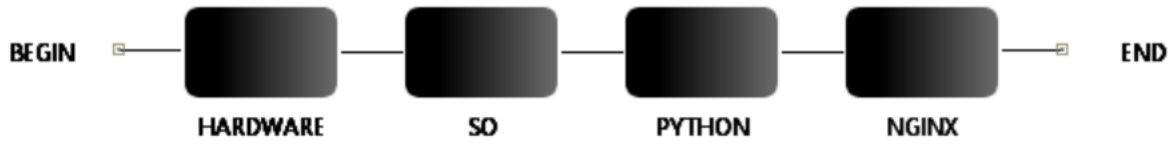


Figura 7 – Modelo RBD da *Fog Computing*

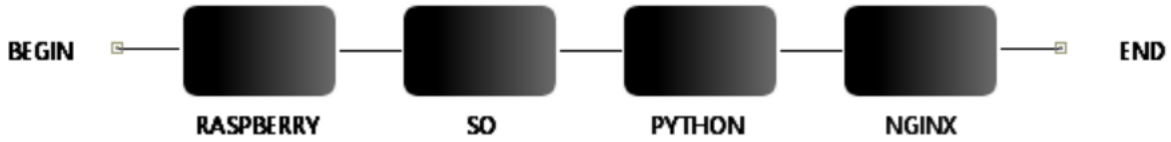


Figura 8 – Modelo RBD da *Edge Computing*

Nos dois modelos, o Python é a aplicação escrita na linguagem Python para reconhecer rostos, essa aplicação recebe um streaming de vídeo e faz o processamento para reconhecer as pessoas que estão no vídeo. O Nginx é um servidor, ele é responsável por receber o streaming do drone e associa a um endereço IP, assim qualquer um que queira ver o vídeo do drone basta acessar esse endereço IP atribuído pelo Nginx. Esse endereço IP é usado pela aplicação em Python para acessar o streaming de vídeo.

Como representado na Figura 7 acima, a infraestrutura de *Fog* é composta por quatro blocos em série: *Hardware*, Sistema Operacional (SO), Python e Nginx. Todos esses quatro blocos são importantes para a disponibilidade, pois o sistema somente estará disponível se todos os blocos estiverem funcionando corretamente. O modo operacional é representado pela Expressão 4.1 adaptada de [34]:

$$OM_{sys} = Hardware \wedge SO \wedge Python \wedge Nginx \tag{4.1}$$

onde Hardware, SO, Python e Nginx são funções *booleanas* que representam o estado de funcionamento dos respectivos componentes do sistema. A disponibilidade total da infraestrutura de *Fog* é calculada através da Equação 4.2:

$$A_{sys} = A_{Hardware} \times A_{SO} \times A_{Python} \times A_{Nginx} \tag{4.2}$$

Onde:

A_{sys} representa a disponibilidade da infraestrutura de *Fog*;

$A_{Hardware}$ representa a disponibilidade do *Hardware*;

A_{SO} representa a disponibilidade do SO;

A_{Python} representa a disponibilidade do Python;

A_{Nginx} representa a disponibilidade da Nginx.

Semelhante ao modelo da infraestrutura de *Fog*, a infraestrutura de *Edge* é representada também por quatro blocos em série, a diferença está no componente Raspberry. A Arquitetura de *Edge* é composta por quatro blocos em série: Raspberry, Sistema Operacional (SO), Python E Nginx. O modo operacional é representado pela Expressão 4.3 adaptada de [34]:

$$OM_{sys} = Raspberry \wedge SO \wedge Python \wedge Nginx \tag{4.3}$$

onde Raspberry, SO, Python e Nginx são funções que representam o estado de funcionamento dos respectivos componentes do sistema. A disponibilidade total da infraestrutura de *Fog* é calculada através da Equação 4.4:

$$A_{sys} = A_{Raspberry} \times A_{SO} \times A_{Python} \times A_{Nginx} \tag{4.4}$$

Onde:

A_{sys} representa a disponibilidade da infraestrutura de *Edge*;

$A_{Raspberry}$ representa a disponibilidade do Raspberry;

A_{SO} representa a disponibilidade do SO;

A_{Python} representa a disponibilidade do Python;

A_{Nginx} representa a disponibilidade da Nginx.

4.2 Considerações Finais

Este capítulo apresentou técnicas de modelagem hierárquica para representar as infraestrutura de *Fog* e *Edge Computing*. O modelo utilizado foi o Diagrama de Blocos de Confiabilidade (RBD) para avaliar a disponibilidade.

5 Estudos de Casos

Este capítulo tem como objetivo apresentar os estudos de casos, onde demonstram a aplicação da metodologia proposta na avaliação de cenários relacionando a dependabilidade, a *Fog* e a *Edge Computing*, para tal apresentamos três estudos de caso. O Estudo de Caso 1 (Seção 5.1) apresenta uma comparação da disponibilidade para as infraestruturas de *Fog* e *Edge Computing*. Com base nos resultados obtidos na comparação, apresentamos no Estudo de Caso 2 (Seção 5.2) uma análise dessa disponibilidade. E por fim, o Estudo de Caso 3 (Seção 5.3) apresenta uma análise de sensibilidade para saber qual componente em cada infraestrutura tem maior impacto na disponibilidade.

5.1 Estudo de Caso 1: Validação dos modelos de disponibilidade de ambientes *Fog* e *Edge Computing*

Neste estudo de caso será descrito os experimentos computacionais utilizados para verificar a precisão do modelo de disponibilidade dos ambientes de *Fog* e *Edge Computing*. Os resultados visam garantir que o modelo represente o sistema real e seja capaz de prover resultados corretos com 95% de confiança. A avaliação foi realizada em um ambiente de *Fog* e *Edge Computing* controlado.

A validação caracteriza-se como requisito primário para comprovação de viabilidade dos modelos apresentados. Neste trabalho, a validação ocorreu através da elaboração de uma aplicação em Python para monitoramento do comportamento do sistema e injeção de falhas.

É importante salientar que foi validado um nó de *Fog* e um de *Edge Computing*, os valores para os demais componentes foram obtidos da literatura [17] e [9]. Optamos pela aplicação dos métodos de injetar falhas e promover reparos de forma intencional desses dois ambientes na infraestrutura de

testes, na tentativa de acelerar a ocorrência dos eventos de falha, em virtude da limitação de tempo para a realização deste trabalho. Logo em seguida, aplicamos o Bootstrap não paramétrico buscando encontrar o intervalo de confiança contendo os valores de disponibilidade obtidos pela experimentação. Em virtude da quantidade de tempo nencessária para a ocorrência de falhas, técnicas conhecidas como "fatores de redução"foram utilizadas no injetor de falhas visando uma aceleração na ocorrência dessa rotina básica em sistemas computacionais, as Tabelas 2 e 3 apresentam os valores utilizados pela rotina, considerando um ano (8760 horas) em 10 horas, o que equivale a um fator de redução de 876 vezes.

Componentes	**MTTF (h)**	**MTTR (h)**
Hardware	10	1
SO	3.3	1
PYTHON	0.9	0,00416667
NGINX	0.9	0,00416667

Tabela 2 – Parâmetros de entrada para a *Fog Computing*

Componentes	**MTTF (h)**	**MTTR (h)**
RASPBERRY	5.4	3.4
SO	3.3	1
PYTHON	0.9	0,00416667
NGINX	0.9	0,00416667

Tabela 3 – Parâmetros de entrada para a *Edge Computing*

A arquitetura básica consiste de um drone, um computador e um Raspeberry Pi. Esses equipamentos representam a quantidade mínima de recursos necessários para testar e validar o ambiente *Fog* e *Edge Computing*. A Figura 9 apresenta uma visão de alto nível desta arquitetura. A definição de uma arquitetura básica torna-se importante, pois através dela somos capazes de compreender melhor o funcionamento do serviço que queremos oferecer e como os principais componentes da infraestrutura interagem entre si, o que possibilita a geração de modelos que representam os modos operacionais do serviço.

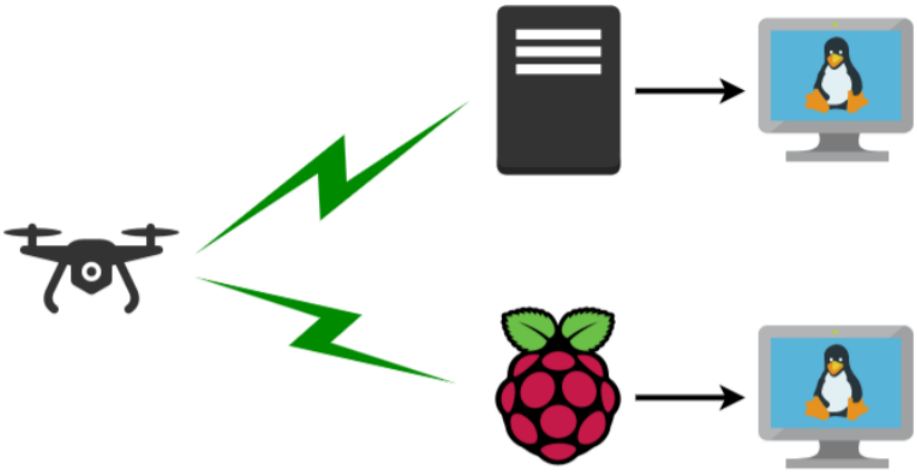


Figura 9 – Arquitetura Base

Dentro do nó da *Fog* e da *Edge* há 4 componentes básicos para o funcionamento do ambiente, são eles: o hardware, o sistema operacional Linux, um servidor Nginx e uma aplicação escrita em Python para reconhecimento facial. A Figura 10 mostra isto.

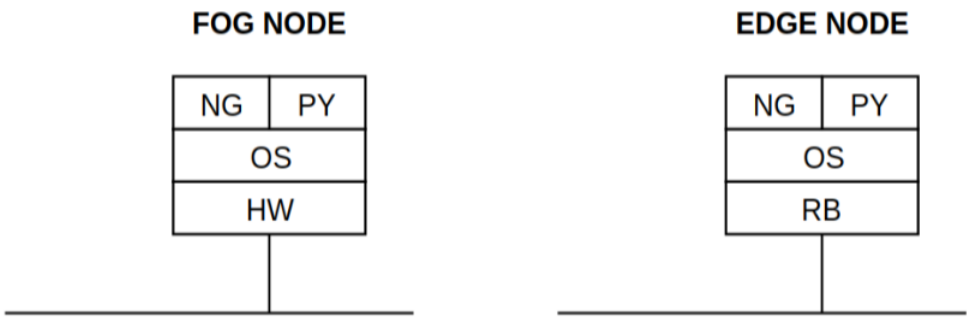


Figura 10 – Componentes básicos

Nesta arquitetura, o serviço apenas estará disponível caso todos os componentes no ambiente *Fog* e no ambiente *Edge* estiverem funcionando. Em outras palavras, uma falha em qualquer componente da *Fog* gera uma indisponibilidade na *Fog*, e uma falha em qualquer componente da *Edge* gera uma indisponibilidade na *Edge*.

A arquitetura que compõe o ambiente *Fog* é formada por CPU Intel(R) Core(TM) i7-3770 CPU 3.40GHz 8 núcleos, com 8GB de RAM e 1TB de HD.

A arquitetura que compõe o ambiente *Edge* é formada por Raspberry Pi zero w com processador single-core de 1 GHz, 512 MB de RAM, slot para cartão microSD, porta Mini-HDMI com saída de 1080p a 60 fps, porta GPIO de 40 pinos, saída para vídeo composto e duas portas Micro-USB.

Executando o experimento por 96 horas para os dois ambientes, obtivemos 73 pontos amostrais da disponibilidade para o ambiente *Edge* e 61 para o ambiente de *Fog*. Em seguida, usamos o Bootstrap não paramétrico que é baseado na reamostragem. A reamostragem é um procedimento que, a partir de uma amostra de dados, uma nova amostra do mesmo tamanho é simulada, considerando todos os valores originais com repetição. Cada valor é obtido com a mesma probabilidade dos outros valores (no nosso caso, 1/73 para a *Edge* e 1/61 para a *Fog*). Para cada reamostragem foi calculada a média. Após 1000 reamostragens, obtivemos 1000 médias, ordenamos todas elas de forma crescente, verificamos qual valor corresponde ao índice 25 e o que corresponde ao 975 para representar o limite inferior e o superior do intervalo de confiança. Esses valores correspondem aos percentis 2,5% e 97,5%, o que implica que 95% dos nossos dados encontram-se entre esses dois valores. Dessa forma, obtivemos os seguintes intervalos de confiança para a disponibilidade:

- ***Edge*:** $0.3472 < \theta < 0.5360$
- ***Fog*:** $0.5630 < \theta < 0.7366$

Calculando a disponibilidade utilizando os modelos RBDs, obtivemos:

- ***Edge*:** 0.4665
- ***Fog*:** 0.6912

O resultado da validação dos modelos com o sistema real, para a disponibilidade, mostram que nossos modelos representam o comportamento do ambiente real.

5.2 Estudo de Caso 2: Avaliação de Disponibilidade nos Ambientes *Fog* e *Edge Computing*

Este segundo estudo de caso tem como objetivo principal avaliar a disponibilidade estacionária e confiabilidade das infraestruturas propostas. A análise de disponibilidade realizada neste trabalho poderá servir como referência para trabalhos futuros, com foco no aumento dessa principal métrica no sistema.

A Tabela 4 apresenta os parâmetros de entrada para a infraestrutura de *Fog*, representado na Figura 7. Os parâmentros dos componentes foram adaptados a partir de [9] e [17].

Componentes	**MTTF (h)**	**MTTR (h)**
Hardware	8760	1.67
SO	2880	1
PYTHON	217.8	0.46
NGINX	217.8	0.46

Tabela 4 – Parâmetros de entrada para a *Fog Computing*

A Tabela 5 apresenta os parâmetros de entrada para a infraestrutura de *Edge*, representado na Figura 8. Os parâmetros referentes à RASPBERRY foram baseados em [17]. Os demais foram baseados em [9] e [17].

Componentes	**MTTF (h)**	**MTTR (h)**
RASPBERRY	4767.8	3.48
SO	2880	1
PYTHON	217.8	0.46
NGINX	217.8	0.46

Tabela 5 – Parâmetros de entrada para a *Edge Computing*

Para a análise dos modelos hierárquicos baseados em RBD utilizou-se a ferramenta *Mercury* [35].

Após inserir os parâmetros de entrada, foi possível realizar a análise de dependabilidade das infraestruturas de *Fog* e *Edge*. Os resultados da análise, em termos de disponibilidade foram:

- ***Fog*:** 0.9952
- ***Edge*:** 0.9946

Neste estudo de caso, propomos avaliar a disponibilidade das duas infraestruturas propostas neste trabalho. Para isto, o modelo hierárquico RBD foi usado para analisar a infraestrutura. Como resultado da avaliação, a infraestrutura de *Fog* alcança uma disponibilidade de 99.52% sem acelerar o tempo médio de falha, o que corresponde a um *downtime* anual de aproximadamente 42 horas. Já a infraestrutura de *Edge* alcança uma disponibilidade de 99.46%, o que corresponde a um *downtime* anual de aproximadamente 46 horas.

5.3 Estudo de Caso 3: Análise de Sensibilidade dos Modelos nos ambientes *Fog* e *Edge*

Neste terceiro estudo de caso, é realizada uma análise de sensibilidade para identificar os componentes mais relevantes que causam impacto na disponibilidade dos modelos das infraestruturas. Pois, os diferentes componentes que constituem um sistema computacional não necessariamente contribuem igualmente para a disponibilidade e desempenho do sistema [34].

Para a realização da análise de sensibilidade, as etapas seguidas foram:

Etapa 1. Definir modelo para análise: Os modelos RBD's representados nas Figuras 7 e 8 foram usados para calcular os valores da sensibilidade para as duas infraestrutura.

Etapa 2. Calcular índice de sensibilidade: Para fazer o cálculo do índice de sensibilidade do modelo RBD, a ferramenta *Mercury* foi utilizada para esse propósito também. A técnica utilizada para o cálculo foi a de diferença percentual.

Após aplicados os valores de entrada, os índices de sensibilidade dos componentes para cada infraestrutura foram alcançados. Os índices das infraestruturas *Fog* e *Edge* estão listados nas Tabelas 6 e 7, respectivamente.

Parâmetros	$S_\theta(A)$
MTTF Aplicações (Python e Nginx)	5.6303 $\times 10^{-}2$
MTTR Aplicações (Python e Nginx)	4.0037 $\times 10^{-}2$
MTTF OS	4.6264 $\times 10^{-}4$
MTTR OS	3.4353 $\times 10^{-}4$
MTTF Hardware	2.5408 $\times 10^{-}4$
MTTR Hardware	1.8866 $\times 10^{-}4$

Tabela 6 – *Ranking* dos índices de sensibilidades do Ambiente *Fog*

Parâmetros	$S_\theta(A)$
MTTF Aplicações (Python e Nginx)	5.6303 $\times 10^{-}2$
MTTR Aplicações (Python e Nginx)	4.2003 $\times 10^{-}2$
MTTF Raspberry	4.6530 $\times 10^{-}4$
MTTF OS	4.6264 $\times 10^{-}4$
MTTR Hardware	3.4669 $\times 10^{-}4$
MTTR OS	3.4353 $\times 10^{-}4$

Tabela 7 – *Ranking* dos índices de sensibilidades do Ambiente *Edge*

Etapa 3. Análise dos resultados: Nesta etapa foi realizada a análise dos resultados obtidos. Pela Tabela 6 pode-se observar que os parâmetros de MTTF Aplicações (Python e Nginx) e MTTR Aplicações (Python e Nginx) possuem os valores de sensibilidades mais altos. Com isso, pode-se afirmar que a Aplicação tem maior impacto na disponibilidade considerando a infraestrutura de *Fog*. O mesmo vale na infraestrutura de *Edge*, como pode-se observar na tabela 7.

Para fazer a análise dos resultados, variou-se um parâmetro de cada vez, mantendo os outros fixos, assim o efeito de cada parâmetro sobre a disponibilidade foi observado.

As Figuras abaixo representam graficamente os resultados quando o parâmetro é variado um por vez para cada infraestrutura. O componente Nginx não será representado pois, como já foi mencionado anteriormente na Etapa 2, a sensibilidade desse componente não afeta a disponibilidade do sistema.

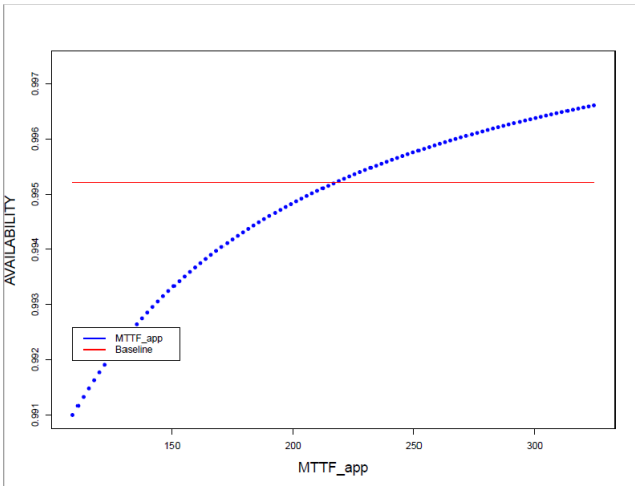
A linha contínua representada em vermelho indica o valor da disponibilidade da Arquitetura base, ou seja, sem variar nenhum parâmetro, por isso é

uma reta uniforme. E a linha pontilhada representada em azul indica o valor da disponibilidade conforme o MTTF ou MTTR do componente aumenta.

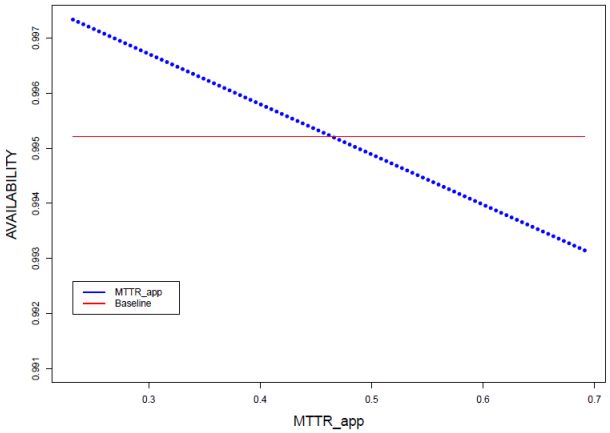
As Figuras 11(a) e 11(b) apresenta a variação dos parâmetros do componentes de maior impacto, no caso a Aplicação.

É observado que com a redução do MTTR da Aplicação existe uma melhora considerável da disponibilidade. Já para o MTTF, quanto maior seu valor, maior a disponibilidade.

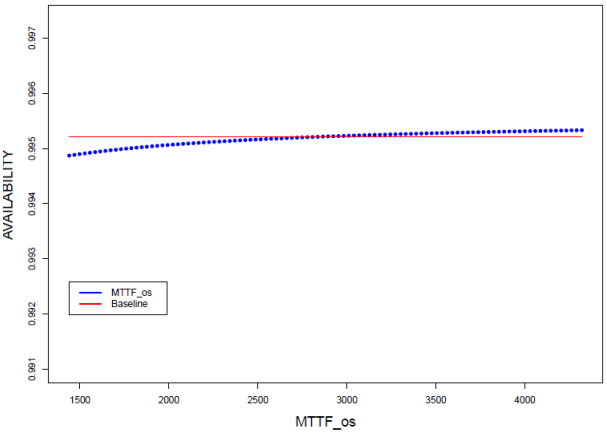
Nas Figuras 11(c), 11(d), 11(e) e 11(f), as variações são mínimas, por isso, pode-se considerar que as Aplicações têm o impacto mais significativo porque possuem um tempo entre falhas menor.



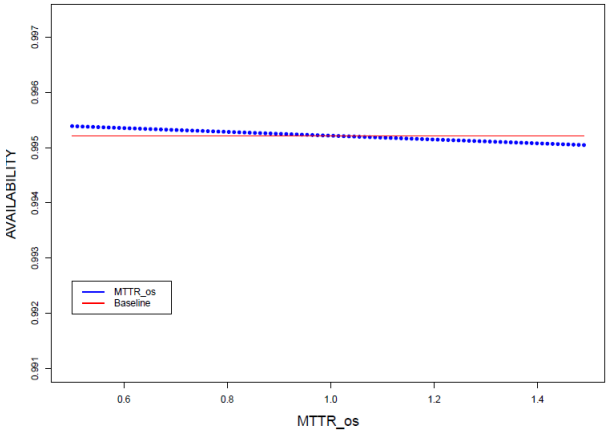
(a) Variação do MTTF da Aplicação na *Fog*



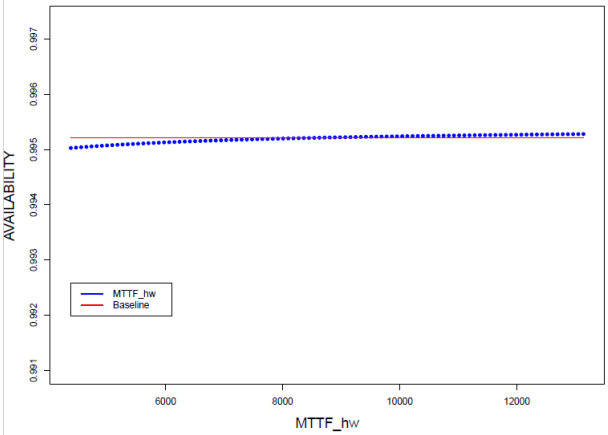
(b) Variação do MTTR da Aplicação na *Fog*



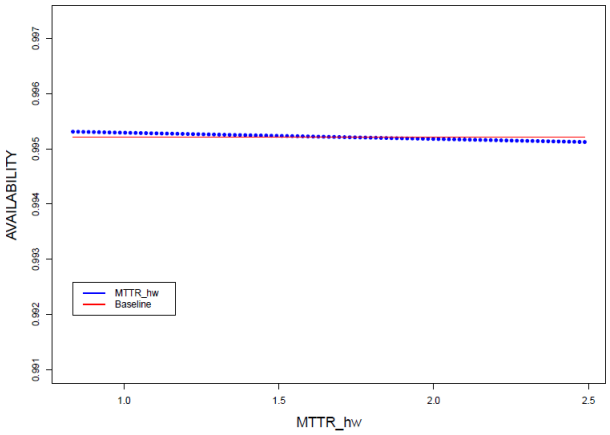
(c) Variação do MTTF do SO na *Fog*



(d) Variação do MTTR do SO na *Fog*



(e) Variação do MTTF do HARDWARE na *Fog*



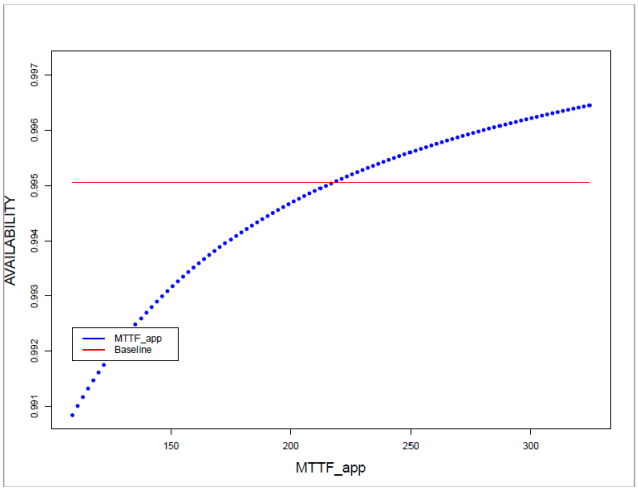
(f) Variação do MTTR do HARDWARE na *Fog*

Figura 11 – Variação da análise de sensibilidade dos componentes da *Fog Computing*

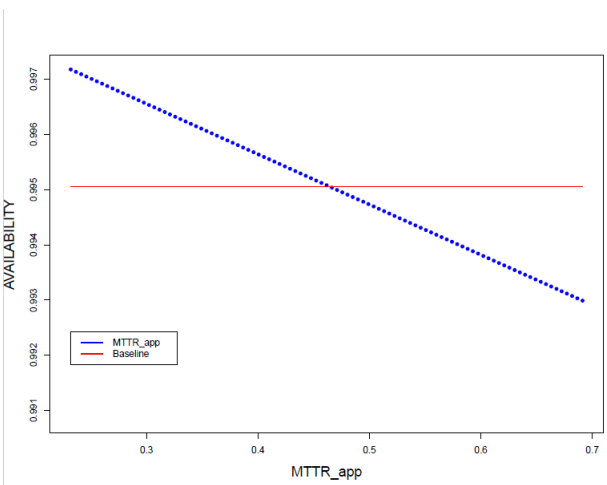
Na análise da sensibilidade para a *Edge*, para o componente Python, a disponibilidade aumenta à medida que o MTTF aumenta e diminui à medida

que o MTTR aumenta, como pode-se observar nas Figura 12(a) e 12(b).

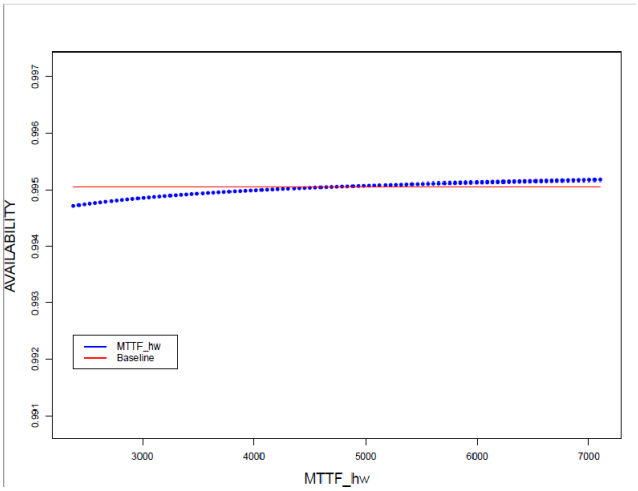
Nas Figuras 12(c), 12(d), 12(e) e 12(f), as variações são mínimas, por isso, pode-se considerar que as Aplicações têm o impacto mais significativo na *Edge* também.



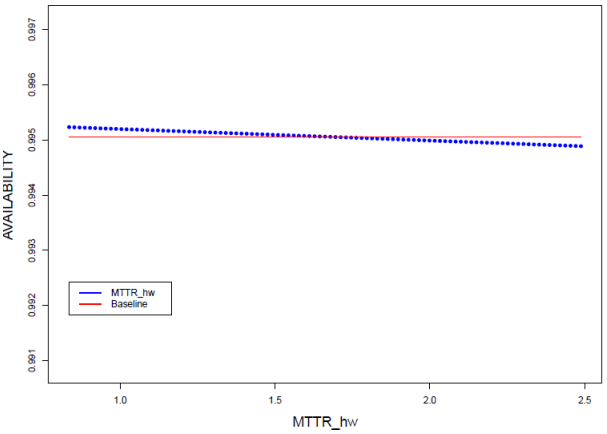
(a) Variação do MTTF da Aplicação na *Edge*



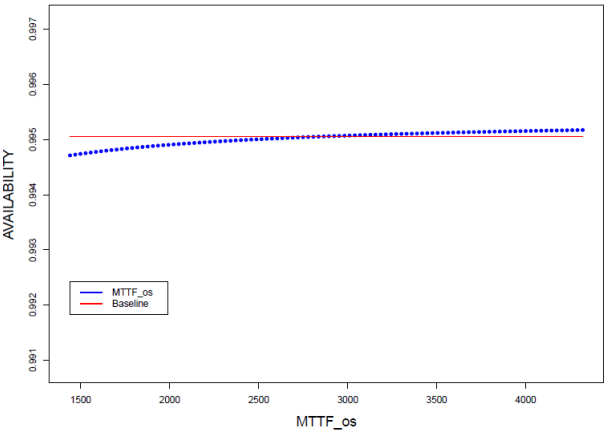
(b) Variação do MTTR da Aplicação na *Edge*



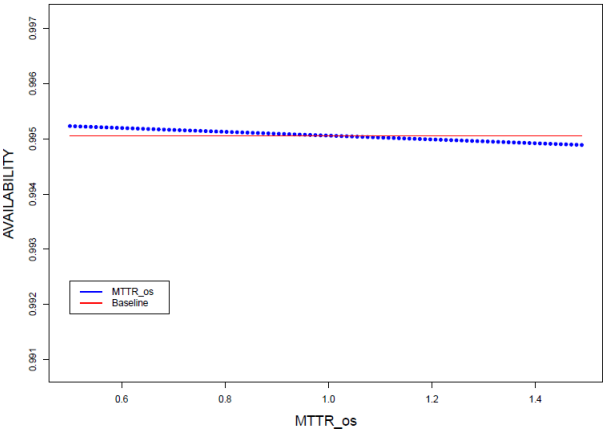
(c) Variação do MTTF da RASPBERRY na *Edge*



(d) Variação do MTTR do RASPBERRY na *Edge*



(e) Variação do MTTF do SO na *Edge*



(f) Variação do MTTR do SO na *Edge*

Figura 12 – Variação da análise de sensibilidade dos componentes da *Edge Computing*

6 Conclusão e Trabalhos Futuros

Os serviços em um ambiente de *Fog* e *Edge Computing* requer alta disponibilidade da aplicação. Nos estudos de casos notou-se que a aplicação rodando nesses dois ambientes tem impactos significativos na disponibilidade do sistema.

A principal contribuição deste trabalho foi a comparação da disponibilidade dos ambientes *Fog* e *Edge Computing*, usando o modelo RBD para checagem desse parâmetro.

Com objetivo de avaliar os modelos das infraestruturas propostas, foram realizados três estudos de caso. O primeiro estudo de caso validou os modelos de disponibilidade dos ambientes *Fog* e *Edge Computing*. No segundo estudo de caso foi realizada uma avaliação da disponibilidade nos ambientes *Fog* e *Edge*. Para isso, a representação desses ambientes foi realizada através da modelagem com RBD, que foi validada no primeiro estudo de caso. No terceiro estudo de caso uma análise de sensibilidade dos modelos usando a técnica de diferença percentual foi realizada nos dois ambientes propostos neste trabalho.

Os resultado obtidos mostram que a aplicação, tanto para o ambiente de *Fog* como para o de *Edge*, a aplicação tem o maior impacto sobre a disponibilidade.

Os próximos passos para enriquecer a pesquisa serão proposição e validação de modelos de desempenho e com isso realizar uma nova análise de sensibilidade.

Bibliografia

[1] *O que é Edge Computing?* Disponível em: https://www.hostmidia.com.br/blog/edge-computing/?utm_campaign=edge-computing&utm_medium=social&utm_source=facebook. Acessado em 10/09/2019.

[2] *Entenda aqui como o Fog Computing impacta a TI.* Disponível em: https://blog.teclogica.com.br/fog-computing-impacta-ti/. Acessado em 03/09/2019.

[3] *O que é cloud computing?* Disponível em: https://www.techtudo.com.br/artigos/noticia/2012/03/o-que-e-cloud-computing.html. Acessado em 10/09/2019.

[4] *As diferenças entre, Cloud, Fog e Edge Computing.* Disponível em: https://www.deal.com.br/blog/as-diferencas-entre-cloud-fog-e-edge-computing. Acessado em 10/09/2019.

[5] *Fog Computing é o novo paradigma para a Internet das Coisas, diz Cisco.* Disponível em: https://cio.com.br/fog-computing-e-o-novo-paradigma-para-a-internet-das-coisas-diz-cisco/. Acessado em 19/11/2019.

[6] Lukas Derner Grüdtner. *Segurança no Contexto de IOT e Fog Computing.* Disponível em: http://www.inf.ufsc.br/~carlos.westphall/Lukas.pdf. Acessado em 03/09/2019.

[7] Weisong Shi et al. "Edge Computing: Vision and Challenges". Em: *IEEE Internet of Things Journal* 3 (out. de 2016), pp. 1–1. DOI: 10.1109/JIOT.2016.2579198.

[8] *Cloud computing vs fog computing vs edge computing na era da internet das coisas industrial.* Disponível em: http://www.ccg.pt/cloud-computing-vs-fog-computing-vs-edge-computing-na-internet-das-coisas-industrial/. Acessado em 21/11/2019.

[9] Jamilson Ramalho Dantas. "Modelos para Análise de Dependabilidade de Arquiteturas de Computação em Nuvem". Diss. de mestrado. Universidade Federal de Pernambuco, 2013.

[10] Maria Clara dos Santos Bezerra. "Modelos para Análise de Disponibilidade de Arquiteturas de um Serviço de *VoD Streaming* na Nuvem". Diss. de mestrado. Universidade Federal de Pernambuco, 2015.

[11] José dos Santos Machado, Admilson de Ribamar Lima Ribeiro e Edward David Moreno. "Uma Revisão da Fog Computing e Seus Desafios de Pesquisas". Em: *Journal On Advances In Theoretical and Applied Informatics* 3.2 (2017), pp. 32–39.

[12] Ronghua Xu et al. "Real-Time Human Objects Tracking for Smart Surveillance at the Edge". Em: Dept. of Electrical Computer Engineering, Dept. of Mathematical Sciences, Binghamton University, School of Computing e Engineering, Univerity of Missouri-Kansas City, Dept. of Geography, New York State University Police, p. 6.

[13] Y.Navaneeth Krishnan, Chandan N Bhagwat e Aparajit P Utpat. "Fog Computing- Network Based Cloud Computing". Em: *IEEE SPONSORED 2ND INTERNATIONAL CONFERENCE ON ELECTRONICS AND COMMUNICATION - SYSTEM(ICECS 2015)*. Bangalore, 2015, p. 2.

[14] F.jalali et al. "Fog Computing May Help to Save Energy in Cloud Computing". Em: *IEEE Journal on Selected Areas in Communications* 34.5 (2016), pp. 1728–1739.

[15] Haymanot Gebre-Amlak. "Toward a Reliable Network Management Framework". Diss. de mestrado. University of Missouri - Kansas City, 2018.

[16] Jamilson Ramalho Dantas. "Planejamento de infraestrutura de nuvens computacionais para serviço de VoD streaming considerando desempenho, disponibilidade e custo". Tese de doutorado. Universidade Federal de Pernambuco, 2018.

[17] Matheus Felipe Ferreira da Silva Lisboa Tigre et al. "Modeling the availability of a n e-health system integrated with edge, fog and cloud infrastructures". Em: Natal, Brasil, p. 6.

[18] Carlos Julian Menezes Araújo. "Tomada de Decisão Multicritério em Infraestruturas como Serviço em Nuvem: Uma Abordagem baseada em Modelos de Dependabilidade, Perfomabilidade e Custo". Tese de doutorado. Universidade Federal de Pernambuco, 2019.

[19] Majid Hajibaba e Saeid Gorgin. "A Review on Modern Distributed Computing Paradigms: Cloud Computing, Jungle Computing and Fog Computing". Em: *Journal of Computing and Information Technology - CIT* 22.2 (2014), pp. 69–84.

[20] Hany F. Atlam, Robert J. Walters e Gary B. Wills. *Fog Computing and the Internet of Things: A Review*. 2018.

[21] José dos Santos Machado. "Implementação de uma Fog Computing para Fornecer Staas a Dispositivos IoT Utilizando Sistemas Embarcados". Diss. de mestrado. Universidade Federal de Sergipe, 2018.

[22] *Ainda não sabe o que é edge computing? A gente conta para você*. Disponível em: https://olhardigital.com.br/noticia/ainda-nao-sabe-o-que-e-edge-computing-a-gente-conta-para-voce/89132. Acessado em 21/11/2019.

[23] *Edge computing needs Edge AI*. Disponível em: https://www.imagimob.com/blog/edge-computing-needs-edge-ai. Acessado em 21/11/2019.

[24] Telium Networks. *Confidencialidade, integridade e disponibilidade: os três pilares da segurança da informação*. Disponível em: https://www.telium.com.br/blog/confidencialidade-integridade-e-disponibilidade-os-tres-pilares-da-seguranca-da-informacao. Acessado em 18/11/2019.

[25] Irene Eusgeld, Felix C. Freiling e Ralf Reussner, eds. *Dependability Metrics*. Springer, 2008.

[26] Ivan Luizio Magalhães e Walfrido Brito Pinheiro, eds. *Gerenciamento de Serviços de TI na Prática: Uma abordagem com base na ITIL*. Novatec, 2007.

[27] *O que é e como funciona a Alta Disponibilidade*. Disponível em: https://appunix.com.br/o-que-e-e-como-funciona-a-alta-disponibilidade.html. Acessado em 03/09/2019.

[28] *PMG Academy*. Disponível em: https://www.pmgacademy.com/pt/glossario-itil/147-disponibilidade-continua. Acessado em 19/11/2019.

[29] Erica Teixeira Gomes de Sousa. "Modelagem de Desempenho, Dependabilidade e Custo para o Planejamento de Infraestruturas de Nuvens Privadas". Tese de doutorado. Universidade Federal de Pernambuco, 2015.

[30] *Segurança da Informação - Confidencialidade, Integridade e Disponibilidade (CID)*. Disponível em: https://www.profissionaisti.com.br/2015/07/seguranca-da-informacao-confidencialidade-integridade-e-disponibilidade-cid/. Acessado em 19/11/2019.

[31] Gabrielly Ferreira Leonardo de Lima. *Estudo de mapeamento sistemático em dependabilidade e métodos ágeis: uma reprodução de estudo de 2012 a 2015*. Monografia (Bacharel em Ciência da Computação), UnB (Universidade de Brasília), Brasília, Brasil. 2016.

[32] Danielle Brito Marques. "Metodologia para Análise da Dependabilidade de Smart Grids". Diss. de mestrado. Universidade Federal do Rio Grande do Norte, 2015.

[33] Kishor S. Trivedi et al. *Reliability Analysis Techniques Explored Through a Communication Network Example*.

[34] Camila Gonzaga de Araujo. "Modelos para Avaliação de Disponibilidade em Ambiente de Internet das Coisas: Um Estudo Aplicado em Serviços MHealth Utilizando Dispositivo Wearable". Diss. de mestrado. Universidade Federal de Pernambuco, 2017.

[35] MoDCS Research Group. Disponível em: http://www.modcs.org/?page_id=2392. Acessado em 15/09/2019.