



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO CIÊNCIA DA COMPUTAÇÃO

ANA RAQUEL DE MORAIS ALVES DA SILVA

Offloading em Dispositivos Móveis Considerando Consumo Energético

Recife

2021

ANA RAQUEL DE MORAIS ALVES DA SILVA

Offloading em Dispositivos Móveis Considerando Consumo Energético

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: Rede de Computadores e Sistemas Distribuídos

Orientador (a): Prof. Dr. Eduardo Antonio Guimarães Tavares

Recife

2021

Catálogo na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

S586o Silva, Ana Raquel de Moraes Alves da
Offloading em dispositivos móveis considerando consumo energético / Ana Raquel de Moraes Alves da Silva. – 2021.
94 f.: il., fig., tab.

Orientador: Eduardo Antonio Guimarães Tavares.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2021.
Inclui referências e apêndices.

1. Rede de computadores e sistemas distribuídos. 2. Offloading. 3. Computação em nuvem. 4. Confiabilidade. 5. Disponibilidade. I. Tavares, Eduardo Antonio Guimarães (orientador). II. Título

004.6

CDD (23. ed.)

UFPE - CCEN 2022 – 18

Ana Raquel de Moraes Alves da Silva

**“Offloading em Dispositivos Móveis Considerando
Consumo Energético”**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 26/02/2021.

BANCA EXAMINADORA

Prof. Dr. Paulo Romero Martins Maciel
Centro de Informática/ UFPE

Prof. Dr. Bruno Costa e Silva Nogueira
Instituto de Computação / UFAL

Prof. Dr. Eduardo Antonio Guimaraes Tavares
Centro de Informática / UFPE
(Orientador)

Dedico este trabalho a Deus, minha rocha e fortaleza. Sem Ele, nada seria, nada existiria e nada faria sentido.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que me permitiu chegar até aqui, apesar das minhas limitações e que meu deus forças durante todo o caminho, a Ele toda honra e toda a glória.

Agradeço ao meu orientador Eduardo Tavares, que me ajudou a superar minhas dificuldades e acreditou em minha capacidade de obter êxito neste processo. Agradeço a Eric Borba que tão gentilmente disponibilizou seu tempo para me ajudar no processo de correção do trabalho, sem dúvida seu auxílio e contribuições foram essenciais.

Agradeço aos meus pais, Gilson e Gilva que sempre acreditaram em mim, a dedicação e incentivo dados durante toda minha vida foram essenciais para me fazer chegar até aqui. Agradeço também a minha irmã Anne Caroline que sempre torce por mim, vibra em cada vitória e sempre está disponível para ouvir e ajudar.

Agradeço ao meu esposo Paulo Mateus, com quem tenho o privilégio de compartilhar a vida, que esteve presente em cada momento dessa trajetória, me ajudando, ouvindo, incentivando e animando. Agradeço por estar presente em mais essa etapa de tantas que vivemos, sua presença e carinho foram essenciais para que eu chegasse até aqui.

Aos familiares e colegas que torceram por mim, em especial a minha tia Marluce que me acompanha durante toda a vida e se faz presente mesmo distante. Aos meus sogros Marcilene, José Paulo, meus cunhados Morgan e Morgana, e a Priscila, que são a família que ganhei de presente.

Aos amigos feitos na caminhada acadêmica mas que foram para além dela: Aline, Amanda, Isabelle, Naira, Vanessa, Marrone, João e Jefferson, obrigada pelos momentos de descontração e pelas inúmeras risadas.

Agradeço a FACEPE pelo apoio financeiro para o desenvolvimento deste trabalho e ao CIN/UFPE pelo apoio acadêmico.

Por fim, agradeço a cada um que fez parte dessa jornada, a todos os amigos, colegas, parentes que torceram e incentivaram. Muito obrigada.

RESUMO

A computação móvel tem avançado rapidamente nos últimos anos, permitindo que diversos usuários tenham acesso a dispositivos móveis cada vez mais potentes, que têm se tornado o meio principal de computação para muitos usuários. O paradigma de computação em nuvem também passou por um grande avanço nos últimos tempos e seu uso tem sido adotado nas mais diversas áreas por fornecer recursos computacionais através da internet de maneira escalável. Ao unir a computação móvel com a computação em nuvem, surgiu o conceito de computação em nuvem móvel, que estende os recursos da computação em nuvem para o ambiente dos dispositivos móveis, permitindo que recursos antes limitados não sejam um impedimento para que os usuários realizem tarefas mais pesadas em seus dispositivos. Uma das técnicas utilizadas por programadores dentro da computação em nuvem móvel é a técnica de *offloading* de aplicações móveis, que move o processamento de tarefas do dispositivo para uma nuvem, com o objetivo de gerar um aumento do poder de processamento e uma economia de energia. Entretanto, nem sempre esses objetivos conseguem ser alcançados, tornando necessária a análise do emprego dessa técnica para identificar os cenários em que ela é vantajosa e aqueles em que ela não apresenta benefícios. Este trabalho apresenta avaliação de disponibilidade e de confiabilidade realizadas através da modelagem do sistema utilizando modelos estocásticos Continuous Time Markov Chain (CTMC) e Reliability Block Diagram (RBD), que representam o dispositivo móvel e o descarregamento da bateria em função da rede móvel utilizada. Em seguida, são realizados experimentos para obtenção de dados do ambiente real para serem utilizados na validação e análise dos modelos concebidos. Os resultados obtidos mostram que o emprego do *offloading* oferece diferentes resultados de acordo com a configuração dos dispositivos utilizados e com a rede móvel utilizada. Nem sempre, porém, os resultados obtidos oferecem vantagens para o usuário, além de seu emprego gerar um impacto sobre a disponibilidade e confiabilidade do dispositivo móvel.

Palavras-chaves: offloading; computação em nuvem; confiabilidade; disponibilidade; CTMC; RBD.

ABSTRACT

Mobile computing has advanced rapidly in recent years, allowing diverse users to have access to mobile devices increasingly powerful since those have become the main computing source for many users. The cloud computing paradigm has also undergone a major advance in recent times and its use has been adopted in the most diverse areas as it provides computational resources over the internet and in a scalable way. By combining mobile computing with cloud computing, the concept of mobile cloud computing emerged, which extends the capabilities of cloud computing to the environment of mobile devices, allowing previously limited resources not to be an obstacle for users to perform heavy tasks on their devices. One of the techniques used by developers within mobile cloud computing is the offloading technique of mobile applications, which moves the task processing from the device to a cloud in order to generate an increase in processing power and an economy of energy. However, these objectives are not always achieved; thus it is necessary to consider the use of this technique to identify scenarios in which it is advantageous and those in which it has no benefits. This work presents an Availability and reliability analysis performed through system modeling using the stochastic models Continuous Time Markov Chain (CTMC) and Reliability Block Diagram (RBD), which represent the mobile device and battery discharge according to the mobile network used. Then, experiments are performed to obtain data from the real environment in order to be used in the validation and analysis of the designed models. The results show that the application of offloading offers different outcomes according to the configuration of the devices and the mobile network used. The results obtained do not always present advantages to the user, and its employment impacts the availability and reliability of the mobile device.

Keywords: offloading; cloud computing; reliability; availability; CTMC; RBD.

LISTA DE FIGURAS

Figura 1 – Arquitetura de MCC	29
Figura 2 – Entidades que afetam a decisão para o offloading	34
Figura 3 – Processo de Decisão para o Offloading	35
Figura 4 – Árvore de Dependabilidade	36
Figura 5 – RBD de Sistema Representado em Série	41
Figura 6 – RBD de Sistema Representado em Paralelo	41
Figura 7 – RBD de Sistema Representado com Mecanismo de Redundância	42
Figura 8 – Cadeia de Markov de Tempo Contínuo	46
Figura 9 – Elementos do Diagrama	49
Figura 10 – Metodologia Proposta	50
Figura 11 – Arquitetura do MpOS	55
Figura 12 – Tela Original do BenchImage	56
Figura 13 – Modelo RBD do Dispositivo Móvel	59
Figura 14 – Modelo Cold Standby do Servidor	60
Figura 15 – Modelo <i>Cold Standby</i>	62
Figura 16 – Modelo <i>Hot Standby</i>	62
Figura 17 – Modelo de Disponibilidade da Bateria	64
Figura 18 – Modelo de Disponibilidade da Bateria sem Uso de Redes Móveis	65
Figura 19 – Modelo de Confiabilidade da Bateria	67
Figura 20 – Modelo de Confiabilidade da Bateria sem Uso de Redes Móveis	67
Figura 21 – Imagem com Filtro Cartoonizer Aplicado	71

LISTA DE TABELAS

Tabela 1 – Comparação Entre Este Trabalho e os Trabalhos Relacionados	24
Tabela 2 – Trabalhos Relacionados ao MpOs	54
Tabela 3 – Configuração do Dispositivo Móvel	70
Tabela 4 – Fatores e Níveis do Experimento	70
Tabela 5 – Tratamentos do Experimento	72
Tabela 6 – Resultados das Replicações	73
Tabela 7 – Resultados da Validação	74
Tabela 8 – Parâmetros Utilizados para as CTMC	75
Tabela 9 – Resultado de Disponibilidade da Bateria do Dispositivo Móvel	76
Tabela 10 – Resultado de Confiabilidade da Bateria do Dispositivo Móvel	76
Tabela 11 – Parâmetros da Análise de Disponibilidade do Dispositivo Móvel	77
Tabela 12 – Disponibilidade do Dispositivo Móvel sem Redundância	77
Tabela 13 – Disponibilidade do Dispositivo Móvel com Redundância Cold Standby . . .	77
Tabela 14 – Disponibilidade do Dispositivo Móvel com Redundância Hot Standby . . .	78
Tabela 15 – Resultados das Replicações nos Tratamentos 1 e 2	91
Tabela 16 – Resultados das Replicações nos Tratamentos 3 e 4	92
Tabela 17 – Tempos Estimados de Descarregamento Total da Bateria	94

LISTA DE ABREVIATURAS E SIGLAS

CTMC	<i>Continuous Time Markov Chain</i>
IAAS	<i>Infrastructure as a Service</i>
mAh	<i>Miliampère</i>
MCC	<i>Mobile Cloud Computing</i>
MDAAS	<i>Mobile Data as a Service</i>
MIAAS	<i>Mobile Infrastructure as a Service</i>
MNAAS	<i>Mobile Network as a Service</i>
MPAAS	<i>Mobile Platform as a Service</i>
MSAAS	<i>Mobile Software as a Service</i>
MTTA	<i>Mean Time do Absorption</i>
MTTF	<i>Mean Time To Failure</i>
MTTR	<i>Mean Time do Repair</i>
PAAS	<i>Platform as a Service</i>
QOS	<i>Quality of Service</i>
RBD	<i>Reliability Block Diagram</i>
SAAS	<i>Software as a Service</i>
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CONTEXTUALIZAÇÃO	13
1.2	MOTIVAÇÃO	15
1.3	OBJETIVOS	17
1.4	ESTRUTURA DA DISSERTAÇÃO	18
2	TRABALHOS RELACIONADOS	19
2.1	AVALIAÇÃO E MODELAGEM DA TÉCNICA DE <i>OFFLOADING</i>	19
2.2	COMPARAÇÃO DOS TRABALHOS	23
2.3	CONSIDERAÇÕES FINAIS	25
3	REFERENCIAL TEÓRICO	26
3.1	OFFLOADING DE APLICAÇÕES MÓVEIS	26
3.1.1	Computação móvel	26
3.1.2	Mobile Cloud Computing	27
3.1.2.1	<i>Arquitetura de MCC</i>	28
3.1.2.2	<i>Modelos de Serviço</i>	29
3.1.3	Offloading	30
3.1.3.1	<i>Passos do Offloading</i>	32
3.1.3.2	<i>Tomada de decisão para o Offloading</i>	33
3.2	AVALIAÇÃO DE CONFIABILIDADE E DISPONIBILIDADE	36
3.2.1	Confiabilidade	37
3.2.2	Disponibilidade	38
3.3	TÉCNICAS DE MODELAGEM	39
3.3.1	Diagramas de Bloco de Confiabilidade	40
3.3.2	Cadeias de Markov de Tempo Contínuo	44
3.4	CONSIDERAÇÕES FINAIS	47
4	METODOLOGIA E ARQUITETURA	49
4.1	METODOLOGIA PROPOSTA	49
4.2	ARQUITETURA DO AMBIENTE DE EXPERIMENTAÇÃO	53
4.3	CONSIDERAÇÕES FINAIS	56
5	MODELAGEM	57

5.1	CONSIDERAÇÕES INICIAIS	57
5.2	MODELOS	58
5.2.1	Modelo de Disponibilidade do Dispositivo Móvel	58
5.2.2	Modelo de Disponibilidade do Dispositivo Móvel com Presença de Redundância	59
5.2.2.1	<i>Cold Standby</i>	60
5.2.2.2	<i>Hot Standby</i>	62
5.2.3	Modelos de Descarregamento da Bateria	63
5.2.3.1	<i>Modelos de Disponibilidade da Bateria</i>	64
5.2.3.2	<i>Modelos de Confiabilidade da Bateria</i>	66
5.3	CONSIDERAÇÕES FINAIS	68
6	RESULTADOS EXPERIMENTAIS	69
6.1	CONFIGURAÇÃO DOS EXPERIMENTOS	69
6.1.1	Cenários de Replicação do Experimento	72
6.2	VALIDAÇÃO	72
6.2.1	Validação do Modelo de Confiabilidade	72
6.3	ANÁLISE DE DEPENDABILIDADE DO DISPOSITIVO MÓVEL	74
6.3.1	Análise de Disponibilidade e Confiabilidade da Bateria do Dispositivo Móvel	75
6.3.2	Análise de Disponibilidade do Dispositivo Móvel	76
6.4	CONCLUSÕES	78
7	CONCLUSÕES	80
7.1	LIMITAÇÕES	81
7.2	TRABALHOS FUTUROS	82
	REFERÊNCIAS	83
	APÊNDICE A – RESULTADO DAS REPLICAÇÕES DO EXPERIMENTO	90
	APÊNDICE B – ESTIMATIVAS PARA O DESCARREGAMENTO TOTAL DA BATERIA	93

1 INTRODUÇÃO

O presente capítulo apresenta uma introdução ao tema da pesquisa e às motivações que fizeram com que esse trabalho fosse iniciado e realizado. São apresentados também os objetivos que foram estabelecidos para serem alcançados ao longo da realização do trabalho e a estruturação de toda a dissertação.

1.1 CONTEXTUALIZAÇÃO

Com o avanço da tecnologia de dispositivos móveis, eles têm se tornado a principal forma de computação para muitos usuários, já que características como a mobilidade e o acesso às redes móveis aumentam o tempo que os usuários passam conectados. Essa nova realidade acaba gerando uma grande busca por dispositivos cada vez mais potentes que consigam executar aplicações com uma alta demanda computacional. Neste contexto, além da necessidade de um bom processamento, surge uma grande procura por baterias que consigam acompanhar a demanda de energia gerada pelo uso do dispositivo móvel.

Entretanto, apesar de o *hardware* de dispositivos móveis continue a evoluir e melhorar, ele sempre terá poucos recursos se comparado ao *hardware* de dispositivos estáticos. Isso ocorre porque melhorar o tamanho, o peso e a vida útil da bateria são prioridades mais altas do que aumentar a capacidade computacional do *hardware* que as pessoas carregam ou usam por longos períodos de tempo (SILVA et al., 2017).

A execução de aplicativos complexos em dispositivos móveis é desafiadora devido às limitações de recursos como capacidade de memória, velocidade de processamento e energia da bateria (GU et al., 2004). Embora os novos dispositivos móveis sejam cada vez mais poderosos em termos de processamento, o alto consumo de energia ainda representa um obstáculo significativo, pois restringe os usuários de desfrutar plenamente de aplicativos com grande demanda de processamento (MACH; BECVAR, 2017). Sendo assim, recursos limitados impedem significativamente a melhoria das qualidades de serviço (DINH et al., 2013).

Os dispositivos móveis fornecem menor poder computacional se comparados a computadores de servidores e *desktops* regulares, além disto, as tarefas intensivas em computação consomem uma grande quantidade de energia da bateria (AKHERFI; GERNDT; HARROUD, 2016). Convencionalmente, o usuário móvel deve processar todas as tarefas em seu dispositivo mó-

vel, no entanto, esse processo pode consumir toda a energia da bateria do dispositivo e levar mais tempo para ser concluído. Entretanto, em um sistema de nuvem móvel, o usuário pode transferir essas tarefas computacionais para a nuvem (TRUONG-HUU; THAM; NIYATO, 2014).

A computação em nuvem móvel pode aumentar os recursos dos dispositivos móveis em situações em que aplicações demandam muitos recursos através da aplicação da técnica de *offloading* (CHEN, 2014). A técnica está se tornando uma abordagem promissora ao migrar tarefas de dispositivos com recursos limitados para um servidor de nuvem com mais recursos através de redes sem fio (WU, 2018). O *offloading* computacional é uma solução que tem o objetivo de melhorar o desempenho dos dispositivos móveis, permitindo a execução das mais diversas aplicações (FAN; LIU, 2016).

No entanto, existem desafios quando se trata do uso do *offloading*. Exemplos disso são os cenários em que o menor custo nem sempre pode ser alcançado devido aos possíveis altos custos de comunicação e execução remota ou questões relacionadas a indisponibilidade ou devido à instabilidade de conexões sem fio (ZHANG; NIYATO; WANG, 2015). Portanto, é de um alto nível de importância observar os diversos aspectos envolvidos na tomada de decisão para o *offloading*.

O processo de tomada de decisão pelo *offloading* envolve algumas questões importantes, como a necessidade de verificar se existe uma conexão entre o dispositivo e o servidor de nuvem e a análise de se vão haver ganhos ao migrar dados ou processamento (SANTOS; REGO; TRINTA, 2017). Por isso é de extrema relevância a realização de pesquisas e estudos que verifiquem situações em que a técnica realmente gera benefícios e assim obter cada vez mais informações que mostrem quando ela não é vantajosa, fazendo com que a aplicação da técnica seja aprimorada e gerando melhorias na utilização de dispositivos móveis.

Este trabalho apresenta uma avaliação do uso da técnica de *offloading* em dispositivos móveis, considerando o consumo energético gerado através da rede móvel em uso. A avaliação de confiabilidade e disponibilidade é feita através da modelagem do sistema utilizando *Reliability Block Diagram* (RBD) e *Continuous Time Markov Chain* (CTMC), cujos modelos propostos representam o dispositivo móvel e o descarregamento da bateria em função da rede móvel utilizada. Adicionalmente, o trabalho realiza experimentos no ambiente real, onde é possível observar o uso da técnica de *offloading* e obter dados que podem ser utilizados nas análises e na validação dos modelos. Os resultados obtidos demonstram que a técnica, apesar de promissora, nem sempre apresenta ganhos ao ser utilizada.

1.2 MOTIVAÇÃO

Nos últimos anos, as aplicações voltadas para dispositivos móveis cresceram nas mais diversas categorias, como entretenimento, saúde, jogos, negócios, redes sociais, viagens e notícias. A razão disso é que a computação móvel é capaz de fornecer uma ferramenta ao usuário quando e onde é necessário, e a mobilidade é uma das características de um ambiente de computação abrangente, em que o usuário pode continuar seu trabalho independente de sua localização (FERNANDO; LOKE; RAHAYU, 2013).

Na medida em que os aplicativos para dispositivos móveis aumentam sua complexidade, aumenta também a demanda por recursos computacionais. Porém, os avanços no *hardware* desses dispositivos e na capacidade de duração da bateria têm sido lentos para responder às demandas computacionais dos aplicativos, que só tem aumentado ao longo dos anos (OTHMAN et al., 2014).

A computação em nuvem móvel (*Mobile Cloud Computing* (MCC)) é um paradigma emergente que abrange a computação móvel, computação em nuvem, rede e virtualização (RAHMAN; GAO; TSAI, 2013). A MCC é empregada para mitigar os problemas críticos em dispositivos móveis, tais como a limitação da capacidade da bateria, a capacidade computacional, limitações de processamento e armazenamento (AL-JANABI et al., 2017). Esse objetivo pode ser atingido por meio do uso da técnica de *offloading*, que consiste em mover processos ou dados de um dispositivo móvel, com limitações de processamento, para um servidor mais potente (SANTOS; REGO; TRINTA, 2017).

Apesar da técnica de *offloading* ser utilizada com o intuito de economizar energia, um dos maiores desafios no uso de dispositivos móveis é a capacidade limitada da bateria (RAJU; SARITHA, 2018), esta abordagem se depara com o desafio de encontrar a condição ideal para realizar o *offloading* e encontrar os fatores que mostram quando é energeticamente viável transferir as tarefas para a nuvem ou quando é melhor manter a execução no próprio dispositivo móvel (RAHMAN; GAO; TSAI, 2013).

O mercado de desenvolvimento tem criado cada vez mais aplicações que necessitam de conectividade, comunicação sem fio de alta velocidade, multimídia de alta definição e que possuem uma grande interação com o usuário, porém, a eficiência energética sempre foi essencial para dispositivos móveis e a importância dada a esse aspecto parece estar aumentando, pois, o desenvolvimento da tecnologia de baterias não foi capaz de atender aos requisitos de energia da crescente demanda de recursos (MIETTINEN; NURMINEN, 2010).

Diversos estudos identificaram que o recurso mais desejado pelos usuários de dispositivos móveis é uma bateria com maior duração (KUMAR; LU, 2010). Além disso, os diversos usuários se conectam à rede através de diferentes tipos de conexão, portanto, alguns fatores como o tipo de conexão utilizada, o consumo de bateria gerado pela conexão e o local em que o usuário está utilizando seu dispositivo, precisam ser considerados.

O uso da MCC é totalmente dependente do uso de rede sem fio e isso implica que, quando a conectividade é limitada, ela pode se tornar inutilizável (KUMAR; LU, 2010). O potencial de *offloading* depende principalmente das tecnologias de rede móvel, pois elas determinam a sua viabilidade, já que algumas redes oferecem uma largura de banda maior que outras, enquanto umas gastam mais energia que outras (AKHERFI; GERNDT; HARROUD, 2016). O custo de transmissão de dados relativamente alto para a rede celular pode impedir que alguns usuários móveis usem os recursos da MCC e por isso, é necessário analisar as condições ideais para seu uso (LIU; KHOUKHI; HAFID, 2017).

Como o *offloading* migra a execução para um computador mais engenhoso, isso envolve a tomada de uma decisão sobre se é viável e o que é viável migrar, e para tomar essas decisões é importante observar e considerar alguns aspectos (KUMAR et al., 2013). Além disso, acredita-se que os pesquisadores irão estender seus trabalhos para integrar o *offloading* em outros cenários, como por exemplo, GPU's, tornando muito importante o estudo e melhoria da técnica.

Existem vários estudos que fazem uma análise de custo/benefício para decidir se o *offloading* deve ou não ser realizado (KUMAR; LU, 2010). A execução de aplicativos em um ambiente de computação em nuvem móvel requer gerenciamento de recursos eficiente (SILVA et al., 2015), reforçando ainda mais a importância de aprofundar os estudos nesse sentido.

O *offloading* tem seus componentes de ganho e perda de energia. Os ganhos de energia são provenientes da energia economizada da computação no dispositivo móvel, enquanto as perdas de energia incluem a transmissão e recepção de dados processados na nuvem. Portanto, é importante determinar qual quantidade de dados deve ser transferida de forma que o consumo de energia seja otimizado (CHOUHAN, 2019).

A técnica é promissora e tem sido utilizada positivamente em diversas aplicações, porém, é importante que ela seja cada vez mais aprimorada para que a tomada de decisão possa envolver mais aspectos que levem em consideração o avanço dos dispositivos móveis e as diversas redes móveis que estão disponíveis hoje. Pesquisadores e desenvolvedores que utilizam essa técnica podem se beneficiar de dados e informações que enriqueçam o processo de tomada de decisão que precisar ser realizado ao se aplicar a técnica.

Ao se utilizar dispositivos móveis, o uso de bateria e o uso de dados são aspectos que estão em evidência e requerem uma atenção maior, pois, aplicações que demandam um alto uso desses dois recursos, podem tornar a experiência do usuário negativa. O *offloading* vem sendo utilizado com o objetivo de melhorar o uso de dispositivos móveis e, portanto, é de extrema importância analisar essa técnica e suas particularidades e perceber até onde ela se faz realmente viável e onde ela passa a demandar mais recursos do que a execução normal no dispositivo, para que seja possível mitigar os problemas existentes, oferecendo um serviço de maior qualidade para o usuário.

1.3 OBJETIVOS

Com base nas motivações que foram anteriormente apresentadas, a presente dissertação de mestrado tem como o principal objetivo a concepção de modelos e a realização de experimento para possibilitar a análise da técnica de *offloading* em dispositivos móveis.

Para que esse objetivo seja alcançado, foram definidos objetivos específicos, sendo eles:

- A extensão do modelo de consumo energético em dispositivos móveis de (MATOS et al., 2014), baseado em CTMC, adicionando a rede 4G ao modelo, para que ele passe a abranger as três interfaces de rede móveis mais utilizadas atualmente;
- A extensão do modelo baseado em RBD do dispositivo móvel, proposto por (OLIVEIRA et al., 2013), incluindo o bloco representando o componente de bateria, com o objetivo de considerar o impacto do consumo energético na disponibilidade do dispositivo móvel;
- A confecção de modelos com mecanismo de redundância *Cold Standby* e *Hot Standby*;
- A avaliação de disponibilidade e confiabilidade da bateria do dispositivo móvel;
- A avaliação de disponibilidade do dispositivo móvel;
- Realização de experimento no sistema real da utilização da técnica de *offloading* em dispositivos móveis, focando no consumo energético gerado ao se utilizar diferentes conexões de rede móvel;
- Demonstração dos resultados experimentais utilizados para validação do modelo de confiabilidade da bateria e para análise nos modelos concebidos.

1.4 ESTRUTURA DA DISSERTAÇÃO

A dissertação está organizada em sete capítulos. O capítulo atual faz uma introdução ao tema seguido da motivação e dos objetivos que compõem este trabalho.

O Capítulo 2 são apresentados os resumos dos trabalhos relacionados que foram utilizados para auxiliar e basear o presente trabalho, além de elencar as principais diferenças com o estudo realizado.

O Capítulo 3 introduz os conceitos que compõem o referencial teórico do trabalho. Primeiramente trazendo conceitos relacionados ao tópico de *offloading* de aplicações móveis, em seguida são apresentados conceitos sobre dependabilidade, por fim são exibidos os conceitos a respeito de RBD e CTMC.

O Capítulo 4 apresenta a metodologia adotada para o desenvolvimento deste trabalho, detalhando as atividades realizadas em cada etapa e por fim é explicitada a arquitetura do sistema utilizado no experimento de medição realizado.

O Capítulo 5 apresenta a modelagem que foi realizada e detalhes sobre os modelos de disponibilidade e confiabilidade que foram propostos utilizando CTMC e RBD.

O Capítulo 6 descreve o ambiente de experimentação que foi configurado, as replicações dos experimentos e seus respectivos resultados, as análises que foram feitas com os modelos desenvolvidos e as validações que foram necessárias.

Finalmente, o Capítulo 7 conclui o trabalho e apresenta as limitações e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Este capítulo apresenta resumos dos trabalhos encontrados na literatura que se relacionam com o tema da dissertação. Alguns deles apresentam experimentos que foram realizados para avaliar a técnica de *offloading*. Outros apresentam modelos que foram concebidos para representar partes do mesmo sistema que está sendo analisado. No final deste capítulo será apresentado um comparativo que foi feito entre os trabalhos apresentados e esta dissertação.

2.1 AVALIAÇÃO E MODELAGEM DA TÉCNICA DE *OFFLOADING*

Para compor o estudo apresentado por (ALTAMIMI et al., 2015) foram desenvolvidos e validados modelos matemáticos para estimar a energia que *smartphones* consomem durante a utilização de diferentes interfaces de rede para realizar atividades que envolvem o *offloading*. A validação do trabalho foi feita utilizando 5 modelos diferentes de *smartphones*. Eles tiveram suas atividades monitoradas e o consumo energético medido em 13 diferentes cenários que foram definidos e cujos dados foram comparados com o modelo matemático que foi proposto. O experimento levou em consideração o tipo de conexão utilizada, a quantidade de dados a ser transferida e os protocolos da camada de aplicação a serem utilizados. Como resultado, foi possível desenvolver um modelo que estima a energia consumida em um *smartphone* para realizar o *offloading* de tarefas utilizando redes Wi-Fi, 4G e 3G. A estimativa é feita considerando a quantidade de dados que o *smartphone* precisa transferir para realizar o *offloading* e a rede a ser utilizada como parâmetros de entrada. Por fim, foi notado que em alguns casos, o *offloading* consome mais energia em alguns cenários do que quando a tarefa é realizada localmente, fazendo com que seja inviável em algumas situações.

O trabalho em questão fornece um modelo importante, porém, pelo constante avanço das tecnologias empregadas no desenvolvimento de dispositivos móveis, a cada dia surgem dispositivos mais avançados e hoje temos acesso a um poder de processamento e uma capacidade de bateria maior do que quando o trabalho foi desenvolvido. Os dispositivos utilizados para validar o modelo proposto não são comumente utilizados hoje. Além disto, o modelo só funciona em cenários em que é possível saber a quantidade de dados que vai ser transferida. Porém, quando não é possível ter acesso a essa informação, ou em cenários em que esse valor não é estável, o modelo não tem como estimar o gasto de energia gerado pelo uso do *offloading*.

No estudo de (SUH; KO; PACK, 2015) foram desenvolvidos modelos analíticos sobre a eficiência do *offloading* através de redes Wi-Fi. Nele são consideradas as técnicas de *offloading* através de Wi-Fi tardias e oportunistas. Para a validação dos modelos analíticos foram realizadas simulações extensivas no simulador desenvolvido. As simulações e os resultados analíticos demonstram que a técnica de *offloading* por meio de Wi-Fi tardio pode melhorar a eficiência de técnica de *offloading* através de Wi-Fi oportunista em algumas situações. Também é mostrado que a eficiência do *offloading* é amplamente afetada pela cobertura das redes de Wi-Fi. Este trabalho tem grande importância ao fazer uma análise do cenário de uso de redes Wi-Fi para a realização do *offloading*, porém, outras redes móveis não são consideradas, nem são consideradas questões relacionadas ao consumo energético.

O trabalho apresentado por (XU et al., 2015) apresenta uma análise do consumo de energia do *offloading* em aplicações móveis desenvolvidas em JavaScript. Uma de suas motivações é o fato de que apesar de existirem soluções de *offloading* para diversas linguagens de desenvolvimento e plataformas, as condições de consumo de energia e da performance divergem de acordo com o cenário e por conta disso é importante realizar análises e avaliações em diferentes contextos. Foram realizados experimentos tanto em dispositivos que utilizam o sistema Android, quanto nos que utilizam sistema IOS. Em todos eles foram realizadas tarefas primeiramente localmente e depois com *offloading*. Foram feitas medições do consumo energético, do tempo para realização da tarefa e do custo para transferência dos dados.

Através dos resultados foi possível notar que em alguns cenários, o *offloading* consome mais energia e tempo para executar a tarefa do que a execução no próprio dispositivo móvel, porém, em alguns casos, seu uso consegue trazer uma boa economia de tempo e energia, mostrando a importância de analisar esses cenários antes de decidir pelo uso da técnica. Apesar do trabalho fazer uma análise do consumo energético da técnica de *offloading*, ele só considera aplicações em JavaScript e, além disso, os dispositivos móveis utilizados para os experimentos possuem configurações inferiores aos mais utilizados atualmente, não sendo mais possível oferecer um resultado atual.

Em seu trabalho, (SANTOS; TRINTA; REGO, 2018) apresenta um *framework* que foi desenvolvido para além dos experimentos realizados para que fosse possível fazer uma avaliação dos diversos aspectos do *framework* e para avaliar o impacto do *offloading* nos tempos de execução e no consumo de energia em dispositivos móveis. O *framework* CAOS D2D (Device-to-Device) (SANTOS et al., 2017) surgiu da extensão do *framework* CAOS (GOMES et al., 2017) e fornece suporte para realização de *offloading* a nível de métodos entre dispositivos móveis e as tarefas

são movidas de um dispositivo móvel cliente para um dispositivo móvel servidor.

Para realização do experimento apresentado em (SANTOS; TRINTA; REGO, 2018), foram utilizados 3 dispositivos móveis, que tiveram seu tempo de execução e seu consumo de bateria monitorados enquanto cada um deles executava dois aplicativos selecionados para o experimento. O primeiro deles realiza operações de soma e multiplicação em matrizes e o segundo permite que o usuário tire fotos e aplique filtros nas imagens capturadas. Os dispositivos móveis estavam conectados na mesma rede Wi-Fi. Após a realização das replicações do experimento foram colhidos os tempos de execução e o consumo de energia e comparados com os tempos de execução e consumo de energia das mesmas tarefas realizadas localmente.

Os resultados constataram que houve situações em que o *offloading* tornou a execução mais rápida, porém, ocorreram casos em que não existiu nenhum ganho. O mesmo se deu com o consumo energético, haja vista que houve casos em que ocorreu uma economia significativa de energia e outros em que não houve ganho algum. Aspectos como a complexidade das tarefas e o tamanho dos dados que são enviados pela rede são de grande peso para definir quando o *offloading* será ou não vantajoso. O trabalho traz uma análise da técnica *offloading*, porém, os experimentos são feitos utilizando apenas rede Wi-Fi, e não consideram outras redes móveis que são bastante utilizadas.

No trabalho de (JOARI et al., 2017), é proposto um mecanismo de *offloading* com tomada de decisão, que considera as condições da rede e o histórico de execuções para fazer a seleção de quais arquivos devem ser migrados do dispositivo móvel para uma *cloudlet*. A decisão é tomada através de árvores de decisão e para a realização do experimento com este mecanismo foram utilizados dispositivos móveis e um servidor configurado para operar como uma *cloudlet*. Neles foram utilizadas duas aplicações, uma delas sendo executada no dispositivo móvel e a segunda na *cloudlet*. O aplicativo que fica no dispositivo móvel utiliza filtros de uma biblioteca que podem ser aplicados em imagens e, além disto, utiliza o *framework* MpOS (MPOS, 2020) para migrar as imagens para uma *cloudlet* a fim de que a aplicação de filtros seja realizada. As mesmas tarefas de aplicação de filtros foram realizadas no dispositivo móvel e depois na *cloudlet*, através do uso do mecanismo de *offloading*, ao mesmo tempo os tempos de execução das tarefas foram colhidos. Após algumas replicações, foi possível utilizar os dados obtidos com os experimentos para criar uma árvore de decisão com o intuito de definir quais arquivos devem ser ou não migrados na hora de executar as tarefas da aplicação.

Os resultados mostraram que houve uma economia de tempo de execução em alguns dos casos, porém houve ocorrências em que o uso de *offloading* tornou o processamento mais

lento (no caso desse experimento foi notado que quanto maior o tamanho dos arquivos a serem migrados, mais vantajoso é o uso do *offloading*.), além disto, foi percebida a importância de um tomador de decisão que considera o histórico das execuções, pois, permite otimizar o uso da técnica. Os experimentos realizados são de grande importância para mostrar a necessidade de um tomador de decisões, porém, no caso deste trabalho não foram consideradas questões relacionadas ao consumo energético gerado pelas atividades realizadas, como também foi utilizada apenas a conexão Wi-Fi para a realização dos experimentos, desconsiderando as demais redes móveis amplamente utilizadas.

O trabalho realizado por (OLIVEIRA et al., 2013) apresenta uma avaliação de disponibilidade e do consumo energético do ambiente de computação em nuvem móvel. Para tal, foram propostos modelos de disponibilidade que consideram diversos cenários na nuvem móvel e levaram em conta características do sistema que podem ter grande influência na disponibilidade, como a conexão sem fio, o descarregamento da bateria gerado pela rede sem fio em uso e a interação entre a nuvem e a *cloudlet*. Os modelos de rede consideram apenas as redes Wi-Fi e 3G. Através dos modelos desenvolvidos, foram realizados estudos de caso utilizando parâmetros obtidos na literatura que considerou cinco cenários diferentes para serem realizadas análises com os modelos.

Os resultados mostram o impacto do uso de redes sem fio e do consumo energético na disponibilidade do sistema, em que a maior disponibilidade é atingida quando os sinais de rede Wi-Fi e 3G estão em boas condições. Um sinal estável da rede Wi-Fi, gera uma maior disponibilidade do que um sinal estável da rede 3G. A bateria teve um tempo médio de funcionamento similar em todos os cenários, exceto no cenário em que apenas a rede 3G estava disponível. As conclusões mostram que quando apenas a rede 3G está disponível para uso há um maior consumo energético e portanto uma disponibilidade menor do sistema, porém, quando tanto Wi-Fi, quanto o 3G estão disponíveis, é possível obter uma maior disponibilidade e maior duração da bateria. As análises realizadas são muito relevantes, porém, o trabalho ainda não considera a rede móvel 4G que hoje é muito mais utilizada. Ademais, também vale considerar que não foram realizados experimentos no sistema real considerando este cenário de consumo energético.

O estudo de (SILVA; MACIEL; MATOS, 2015) apresenta a ferramenta SmartRank, que é um *framework* criado para fazer o particionamento de tarefas e realizar o *offloading* em dispositivos móveis com o objetivo de aumentar o desempenho em termos de tempo de resposta. Através da ferramenta é possível tirar uma foto no dispositivo móvel e solicitar o reconhecimento facial

através da nuvem móvel, o objetivo é enviar essa solicitação para as cloudlets próximas para utilizá-las para realizar a tarefa, esse processo envolve a realização de cálculos e análise para ver o caminho que vai ser mais rápido e menos custoso. Para realizar essa análise, foram realizados experimentos utilizando algumas máquinas virtuais para receber as solicitações dos dispositivos móveis, o experimento então fez a medição dos dados para notar se utilizar várias máquinas virtuais simultaneamente traria ganhos.

Os resultados mostraram que utilizar várias cloudlets é viável para reconhecimento facial, pois os experimentos mostraram que houve uma melhora no tempo de resposta do processo de reconhecimento, ou seja, ao invés de utilizar apenas um recurso, múltiplas máquinas podem resolver mais rapidamente uma tarefa de reconhecimento.

2.2 COMPARAÇÃO DOS TRABALHOS

A seguir, na Tabela 1, é apresentada uma comparação entre os trabalhos mencionados neste capítulo e o presente trabalho. Assim, é possível ressaltar os diferenciais do trabalho desenvolvido. Para a comparação foram levadas em consideração as avaliações que foram realizadas utilizando os modelos desenvolvidos e também comparações com o experimento que foi realizado.

A segunda coluna, mostra que apenas três dos estudos citados apresentam algum tipo de modelagem do sistema ou de parte dele. Em seguida, é possível perceber que apenas um dos estudos apresenta avaliação de disponibilidade e nenhum apresenta de confiabilidade. Em relação aos experimentos que foram realizados nos estudos, apenas um deles realiza os experimentos considerando os três tipos de rede móvel mais utilizados, os demais utilizam apenas uma ou duas delas. Dos estudos selecionados, quatro deles realizam experimentos com a técnica de *offloading* em dispositivos móveis e apenas dois deles utilizam dispositivos com configurações mais atuais.

É possível notar que cada um dos estudos fez contribuições importantes em diferentes aspectos que foram tratados em seus trabalhos, o trabalho atual se propõe a contribuir com o tema e fazer tanto a modelagem do sistema estudado como a avaliação de confiabilidade e disponibilidade do sistema e por fim, propor e realizar experimentos no sistema real, utilizando dispositivo móvel e as três principais redes móveis.

Tabela 1 – Comparação Entre Este Trabalho e os Trabalhos Relacionados

Estudo	Modelagem	Avaliação de Disponibilidade	Avaliação de Confiabilidade	Experimentos utilizando Wi-Fi, 4G e 3G	Experimentos utilizando dispositivos móveis	Experimentos utilizando dispositivos móveis atuais
Este trabalho	✓	✓	✓	✓	✓	✓
(ALTAMIMI et al., 2015)	✓			✓	✓	
(SUH; KO; PACK, 2015)	✓					
(XU et al., 2015)					✓	
(SANTOS; TRINTA; REGO, 2018)					✓	✓
(JOARI et al., 2017)					✓	✓
(OLIVEIRA et al., 2013)	✓	✓				
(SILVA; MACIEL; MATOS, 2015)	✓					✓

Fonte: A autora (2021)

2.3 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os resumos dos trabalhos relacionados com o tema da dissertação, que abordam o tema do *offloading* em dispositivos móveis e realizam experimentos e/ou a modelagem do sistema para analisar melhor a aplicação da técnica, assim, foram expostos os principais pontos e os resultados que foram obtidos. Alguns trabalhos realizaram experimentos utilizando a técnica de *offloading*, e mostrando um sistema que pudesse simular e medir alguns aspectos importantes durante a replicação dos experimentos, outros desenvolveram modelos que representam partes do mesmo sistema sendo analisado. Além disto, foram ressaltadas as principais diferenças entre os trabalhos e esta dissertação e uma tabela comparativa foi exibida ao final do capítulo.

3 REFERENCIAL TEÓRICO

O presente capítulo contém os conceitos básicos sobre os assuntos abordados nesta dissertação. Primeiramente são apresentados os conceitos que envolvem a técnica de *offloading*, como a computação móvel e *mobile cloud computing*, como também as questões que envolvem a realização e a tomada de decisão para o *offloading*. Em seguida, são dadas as definições dos métodos e conceitos utilizados na avaliação de confiabilidade e de disponibilidade. Por fim, são mostradas as técnicas de modelagem que foram usadas na elaboração do trabalho, suas respectivas definições e exemplos de representação.

3.1 OFFLOADING DE APLICAÇÕES MÓVEIS

O *offloading* de aplicações móveis é uma solução que vem sendo amplamente utilizada no desenvolvimento de aplicações móveis, e sua definição envolve diversos conceitos e paradigmas que embasam e fazem parte de sua composição, dentre eles a computação móvel e a computação em nuvem móvel. Os conceitos mais importantes são explanados a seguir.

3.1.1 Computação móvel

A computação móvel é baseada em um conjunto de três conceitos principais: *hardware*, *software* e comunicação. No *hardware* estão os dispositivos móveis, como *smartphone* e *laptop* ou seus componentes móveis. Já o *Software* de computação móvel são as inúmeras aplicações móveis nos dispositivos, como navegador, antivírus e jogos. Por sua vez, a comunicação inclui a infraestrutura de redes móveis, protocolos e tráfego de dados, que devem ser transparentes para os usuários finais (QI; GANI, 2012).

Dispositivos móveis estão se tornando cada vez mais uma parte essencial da vida humana como ferramentas de comunicação mais eficazes e convenientes, não limitadas por tempo e local (DINH et al., 2013). Os usuários hoje têm acesso a *smartphones*, *tablets*, *e-readers*, *smartwatches* entre outros dispositivos que podem se conectar à internet móvel e serem utilizados a qualquer momento e em qualquer lugar. O aumento significativo no desenvolvimento de dispositivos móveis permite que os usuários os transportem para fins de comunicação, orientações e acesso à Internet (HLATSHWAYO; ZUVA, 2016).

Juntamente com o avanço do poder de processamento destes dispositivos vem ocorrendo também um avanço na área de desenvolvimento de aplicações móveis que, com o objetivo de oferecer as mais diversas soluções, fazem uso de vários recursos que envolvem novas tecnologias e demandam um grande poder de processamento. Este avanço, combinado ao aumento da utilização de dispositivos móveis, fez surgir o desafio de tornar as aplicações mais rápidas e mais eficientes, além de fazer com que fosse necessário encontrar maneiras que permitissem que as aplicações aloquassem cada vez menos recursos do dispositivo.

À medida que os dispositivos móveis se tornam cada vez mais poderosos, as aplicações se tornam mais exigentes em recursos computacionais, ainda assim os usuários se deparam com algumas limitações em relação aos recursos disponíveis para processamento de tarefas mais pesadas. Para lidar com esse problema, os dispositivos móveis podem utilizar recursos externos para executar tarefas que demandam um maior poder computacional, assim surgiu a *Mobile Cloud Computing* (WU et al., 2015).

3.1.2 Mobile Cloud Computing

A computação em nuvem móvel é um paradigma para dispositivos móveis, por meio da qual a maior parte do processamento e armazenamento de dados associados aos aplicativos é movida do dispositivo móvel para plataformas poderosas e centralizadas de computação localizadas na nuvem (PRASAD; GYANI; MURTI, 2012). Esse processo pode ser considerado a integração da tecnologia de computação em nuvem com dispositivos móveis para torná-los completos em termos de potência computacional, memória, armazenamento, energia e observação de contexto (OTHMAN et al., 2014).

(GAO et al., 2013) define *mobile cloud computing* como:

"MCC é um paradigma emergente de nuvem móvel que utiliza a computação móvel, a rede e a computação em nuvem para estudar modelos de serviços móveis, desenvolver infraestruturas, plataformas e aplicativos de serviços para clientes móveis. Seu principal objetivo é fornecer serviços móveis com reconhecimento de local com mobilidade para os usuários com base em recursos escaláveis de nuvem móvel em redes, computadores, armazenamentos e dispositivos móveis. Seu objetivo é fornecê-los com recursos seguros de nuvem móvel, aplicativos de serviço e dados usando recursos de nuvem móvel com eficiência de energia em um modelo de

pagamento conforme o uso."

De acordo com as afirmações de (UMAIR et al., 2015), a MCC pode ser definida como uma estrutura em que o armazenamento e o processamento da computação ocorrem fora do dispositivo móvel, considerando que os recursos podem ser explorados no dispositivo. Ela estende o conceito de computação em nuvem para os dispositivos móveis, trazendo vantagens em relação à utilização de recursos e à execução de tarefas.

Ao utilizar serviços de armazenamento na nuvem para fornecer armazenamento *online* e processamento na nuvem, o resultado é um aumento da capacidade de processamento dos dispositivos móveis (SHIRAZ et al., 2012). Seu uso, proporciona um aumento da confiabilidade e da disponibilidade dos serviços e minimiza os requisitos de energia e computação em dispositivos móveis (LI et al., 2015).

3.1.2.1 Arquitetura de MCC

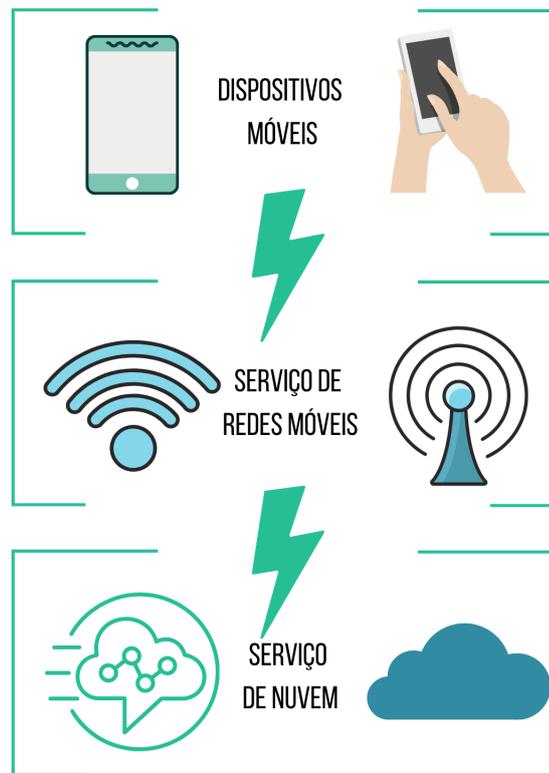
A arquitetura geral da *Mobile Cloud Computing* pode ser dividida em três camadas que são interligadas entre si, a Figura 11 fornece a visão geral dessas camadas que são:

Dispositivos Móveis: a primeira camada consiste em muitos usuários de serviços de nuvem móvel que acessam serviços em nuvem usando seus dispositivos móveis (como *smartphones* e *tablets*). Esses dispositivos se conectam à rede móvel usando pontos de acesso sem fio (NOOR et al., 2018).

Serviço de Redes Móveis: a segunda camada, conecta os usuários e seus dispositivos aos serviços disponíveis na nuvem e através da internet, eles podem ter acesso aos serviços em nuvem através de redes de telecomunicações 3G ou 4G ou conexões sem fio (FAN; LIU, 2016).

Serviço de Nuvem: consiste em vários provedores da computação em nuvem que fornecem todos os tipos de serviços na nuvem, incluindo *Infrastructure as a Service* (IAAS), *Platform as a Service* (PAAS) e *Software as a Service* (SAAS). Esses serviços podem ser aumentados ou reduzidos com base no que os usuários exigem. Tudo isso também é fornecido aos usuários de dispositivos móveis que podem acessar serviços em seus dispositivos através da internet (NOOR et al., 2018).

Figura 1 – Arquitetura de MCC



Fonte: A autora (2021)

3.1.2.2 Modelos de Serviço

Os modelos de serviço na nuvem móvel podem ser divididos de acordo com o recurso oferecido, semelhante aos modelos da computação em nuvem:

Mobile Software as a Service (MSAAS)- Nesse modelo, uma aplicação móvel é implantada e executada em uma nuvem, e os usuários móveis acessam os serviços por meio de Internet sem fio (GAO et al., 2013).

Ex.: Netflix (netflix.com), Google Docs (google.com/docs).

Mobile Platform as a Service (MPaaS)- Os provedores de PaaS oferecem plataformas hospedadas online para desenvolvimento, implantação e provisionamento de aplicativos. Sob essa plataforma, há uma infraestrutura de nuvem oculta do usuário, e ele pode ser fornecido como uma extensão do PaaS regular, com os provedores tradicionais de PaaS estendendo suas ofertas a recursos móveis (DIHAL et al., 2013).

Ex.: Microsoft Azure (azure.microsoft.com), Amazon EC2 (aws.amazon.com/pt/ec2) e Google App Engine (cloud.google.com/appengine).

Mobile Infrastructure as a Service (MIAAS)- Consiste em uma infraestrutura de nuvem habilitada para dispositivos móveis e seus recursos de computação e armazenamento, bem como para os componentes e dispositivos de rede, que são provisionados, gerenciados e retornados de acordo com as solicitações de clientes móveis sob demanda em um modelo de pagamento conforme o uso (GAO et al., 2013).

Ex.: Amazon EC2 (aws.amazon.com/pt/ec2), Amazon S3 (aws.amazon.com/pt/s3).

Mobile Network as a Service (MNAAS)- Nesse modelo de serviço, os provedores oferecem uma infraestrutura de rede para que os usuários possam criar suas próprias redes, controlar os tráfegos e conectar-se aos servidores (MOLLAH; AZAD; VASILAKOS, 2017).

Ex.: OpenStack (www.openstack.org).

Mobile Data as a Service (MDAAS)- Nesse modelo, bancos de dados (ou repositórios de dados) são disponibilizados para dispositivos móveis em larga escala e recursos de armazenamento são fornecidos aos clientes para oferecer suporte à transações, ao gerenciamento e aos acessos pela rede sem fio, alavancando a tecnologia emergente de banco de dados na nuvem (GAO et al., 2013).

Ex.: iCloud (icloud.com), Google Drive (drive.google.com), Dropbox (dropbox.com)

3.1.3 Offloading

Para auxiliar a execução de tarefas que demandam um uso intensivo de recursos, elas são migradas do dispositivo móvel para a nuvem ou para um servidor que possui os recursos e infraestrutura necessários, sendo tal processo chamado de *offloading* (ENZAI; TANG, 2014). A ideia básica envolve o aprimoramento dos recursos computacionais de dispositivos com recursos limitados, aproveitando as funcionalidades da computação em nuvem (KHANNA; KERO; KUMAR, 2016). Dessa maneira, é possível explorar os recursos de computação e armazenamento disponíveis na nuvem móvel, fazendo com que aplicativos que fazem um uso intensivo de processamento possam ser executados mesmo em dispositivos móveis de baixo recurso (FERNANDO; LOKE; RAHAYU, 2013).

De acordo com (KUMAR et al., 2013) o *offloading* é uma solução utilizada para aumentar os recursos de sistemas móveis, migrando a computação para computadores com mais recursos e se diferencia do modelo cliente-servidor pelo fato de utilizar a virtualização para migrar programas para servidores fora do ambiente de computação imediato dos usuários. A técnica pode ser definida como um conceito utilizado para resolver os problemas inerentes à computação móvel usando provedores de recursos diferentes do próprio dispositivo móvel para que se possa hospedar a execução de aplicativos móveis, a infraestrutura em que o armazenamento e o processamento de dados ocorrem fora do dispositivo móvel. Esta técnica pode ser chamada de "nuvem móvel"(FERNANDO; LOKE; RAHAYU, 2013).

A ideia básica por trás do conceito da técnica é particionar a aplicação em várias tarefas para que ela seja executada remotamente, pois assim os componentes de um aplicativo (tarefas) podem ser identificados como, por exemplo, aqueles que podem ser executados localmente, aqueles que devem ser executados na nuvem e aqueles que podem ser executados local ou remotamente (KHANNA; KERO; KUMAR, 2016). Existem diferentes *frameworks* que realizam *offloading*, e embora sejam diferentes entre si, todos eles têm o mesmo objetivo final: descarregar as tarefas pesadas de um dispositivo cliente para algum servidor remoto mais poderoso (SILVA et al., 2017).

O uso da técnica tem o objetivo de manter a *Quality of Service* (QOS) e diminuir o custo e o impacto dos serviços que demandam muito da rede móvel, por essa razão, é esperado que o *offloading* torne-se um tema chave para a indústria de dispositivos móveis, já que o tráfego de dados nas redes móveis só vem aumentando rapidamente (AIJAZ; AGHVAMI; AMANI, 2013), e a sua utilização melhora o desempenho dos aplicativos, reduz o consumo de energia da bateria, permitindo a execução de aplicativos que não poderiam ser executados devido à insuficiência de recursos do *smartphone* (OTHMAN et al., 2014).

Entretanto, a técnica só se torna vantajosa quando a execução local (no dispositivo móvel) consome mais tempo e energia do que o processo de enviar as tarefas para serem realizadas na nuvem, e deste modo muitos fatores podem afetar a decisão para o *offloading* e influenciar todo o processo. O seu potencial depende principalmente das tecnologias de redes móveis e Wi-Fi, pois, elas determinam a viabilidade da técnica. Hoje, a tecnologia Wi-Fi é capaz de fornecer conexões de alta largura de banda, no entanto, a transmissão de dados usando a rede móvel requer uma quantidade considerável de energia do dispositivo móvel quando comparada a uma rede Wi-Fi (AKHERFI; GERNDT; HARROUD, 2016).

O *offloading* computacional tem sido tema de vários estudos, contudo apenas uma parte

desse deles se concentra no efeito do *offloading* no consumo de energia do dispositivo móvel, haja vista que a maioria das pesquisas foca no tempo de resposta e no consumo de outros recursos (MIETTINEN; NURMINEN, 2010). Portanto, é de grande importância investir num processo robusto de tomada de decisão que possa priorizar e otimizar a economia de energia.

3.1.3.1 Passos do Offloading

O processo para realizar a técnica de *offloading* pode ser resumido em três passos principais (AKHERFI; GERNDT; HARROUD, 2016):

Particionamento da Aplicação: A primeira etapa consiste no particionamento da aplicação. Nesse processo os componentes da aplicação serão divididos entre aqueles componentes que são capazes de utilizar *offloading* e os que não são, e essa etapa define quais componentes ficarão retidos no dispositivo móvel e quais serão migrados para a nuvem. Essa decisão é definida pelo próprio programador e pode ser feita a nível de desenvolvimento ou em tempo real, valendo ressaltar que a maneira que essa divisão é feita influencia diretamente na eficiência da técnica.

Preparação: A etapa de preparação realiza todas as ações necessárias para a execução do *offloading*, isso inclui a seleção de um servidor remoto, a transferência e a instalação do código e dos dados para o lugar remoto em que as tarefas serão processadas.

Decisão para o Offloading: Por fim, é realizada a etapa de decisão, que é feita antes do início da execução remota para os componentes que vão ser migrados. Essa decisão pode ser previamente tomada ou feita em tempo de execução. Para decidir se o *offloading* é realmente viável, diversos aspectos devem ser considerados.

A etapa de Particionamento da Aplicação introduz o conceito de granularidade do *offloading*, que indica a maneira que a partição dos componentes das aplicações é feita, que podem ser classificadas em três (LIN et al., 2019):

Offloading Completo: nesse modo, toda a parte que vai ser processada é migrada, deixando o dispositivo móvel responsável apenas pela interface do usuário, entrada/saída e detecção de dados.

Tarefa/Componente: esse tipo de granularidade divide as aplicações em diferentes tarefas e componentes e, em seguida, migra tarefas/componentes que demandam um alto poder de processamento para infraestruturas remotas. Para fazer esse processo, os desenvolvedores analisam o fluxo de trabalho da execução da aplicação para obter uma visão profunda do comportamento do programa, existem diferentes estratégias de particionamento.

Método/Thread: esse modo é uma migração de computação de baixa granularidade. Em geral, uma aplicação contém uma quantidade considerável de métodos e, então, o *offloading* no nível do método precisa de mecanismos de particionamento para auxiliar. Por meio desses mecanismos é possível identificar os métodos que exigem um maior poder computacional.

O passo de decisão para o *offloading* é de grande importância para o uso da técnica e está diretamente ligado com o seu sucesso e eficiência. Por isso, a próxima subseção trata com mais detalhes sobre essa etapa, com o objetivo de ressaltar a importância de obter cada vez mais dados e informações que permitam que o processo de tomada de decisão seja cada vez mais aprimorado.

3.1.3.2 Tomada de decisão para o Offloading

O *offloading* de computação móvel pode ser feito de duas maneiras: através do particionamento de código estático ou dinâmico. Quando feito de maneira estática, envolve o particionamento explícito de uma aplicação por um desenvolvedor a fim de executá-lo remotamente, enquanto no particionamento dinâmico, a partição da aplicação é feita de maneira automatizada, sem qualquer intervenção humana, dependendo das variáveis do ambiente de execução (KHANNA; KERO; KUMAR, 2016).

Como o *offloading* migra a computação para um computador com mais recursos, esse processo envolve a tomada de decisões sobre "se" e "o que" migrar, por esta razão, existem diversas pesquisas sobre decisões de *offloading* para melhorar o desempenho e economizar energia (DE, 2016).

Algumas questões importantes são consideradas durante esse processo de decisão como, por exemplo, precisar definir qual a principal motivação do *offloading*: economizar energia, melhorar o desempenho da computação, ou ambos? Outra questão que deve ser analisada é o real ganho que o uso da técnica trará, pois não é viável recorrer às nuvens para um trabalho que possa ser executado com facilidade e eficiência localmente. Nos casos de trabalhos difíceis

de serem executados localmente, movê-lo para a nuvem pode resultar em um grande volume de transferência de dados que, embora possa não ser um problema grave para usuários com conexões de rede com fio de alta velocidade, pode contradizer amplamente os benefícios para usuários móveis com suas interfaces sem fio que consomem muita energia (MA et al., 2013).

A decisão para o *offloading* computacional é um processo extremamente complexo e é afetado por diferentes entidades, como usuário, conexão, *smartphone*, aplicação e serviço de nuvem (OTHMAN et al., 2014). Cada uma dessas entidades possui características que variam conforme o contexto em que cada uma delas se encontra e isto tem efeito direto na tomada de decisão para o *offloading*, a Figura 2 ilustra essas entidades e alguns dos aspectos relacionados a suas características.

Figura 2 – Entidades que afetam a decisão para o offloading

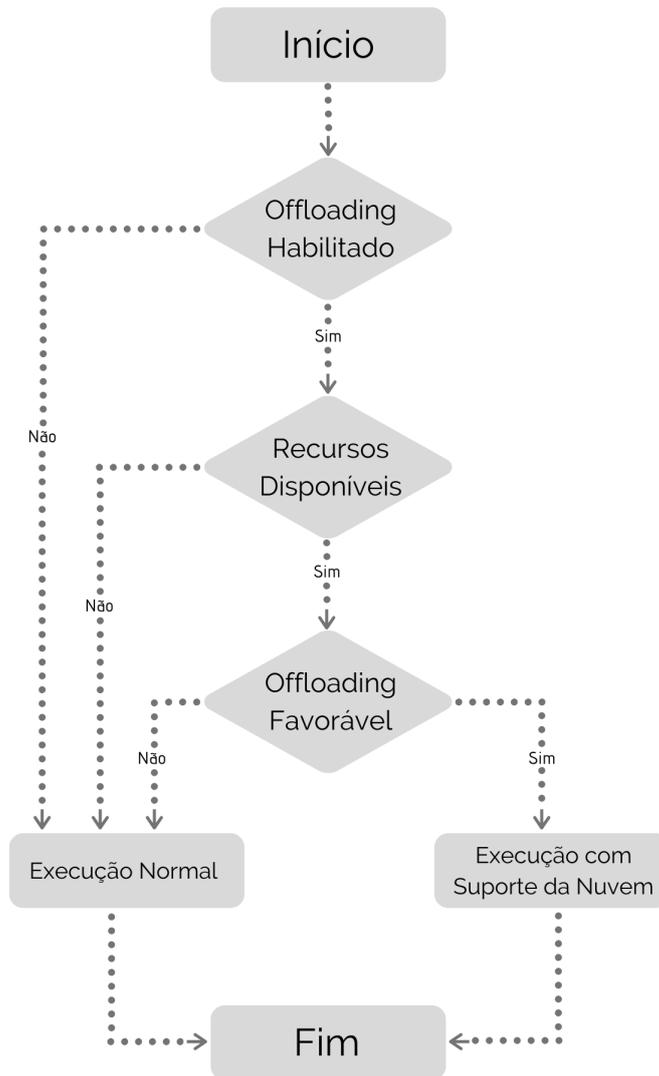


Fonte: Adaptado de (OTHMAN et al., 2014)

Levando em consideração as entidades acima citadas, para que o *offloading* seja enfim executado ou não, ele passa por um esquema de decisão que pode ser observado na Figura 3. Cada um dos passos representados nos losangos da figura, representam uma série de análises e decisões que, por sua vez, podem ser cada vez mais aperfeiçoadas à medida que mais estudos e informações vão sendo adquiridas para enriquecer o processo de tomada de decisão.

Como ilustrado na Figura 3, é possível dividir o processo de decisão para o *offloading* em três passos:

Figura 3 – Processo de Decisão para o Offloading



Fonte: A autora (2021)

Passo 1 - Offloading Habilitado: a primeira etapa consiste na execução do aplicativo em questão e a realização da verificação se o *offloading* está habilitado ou não pelo usuário;

Passo 2 - Recursos Disponíveis: em seguida, é feita a averiguação se todos os recursos necessários para realização dessa tarefa estão disponíveis, como se há conexão com a internet, se a nuvem a ser utilizada está disponível;

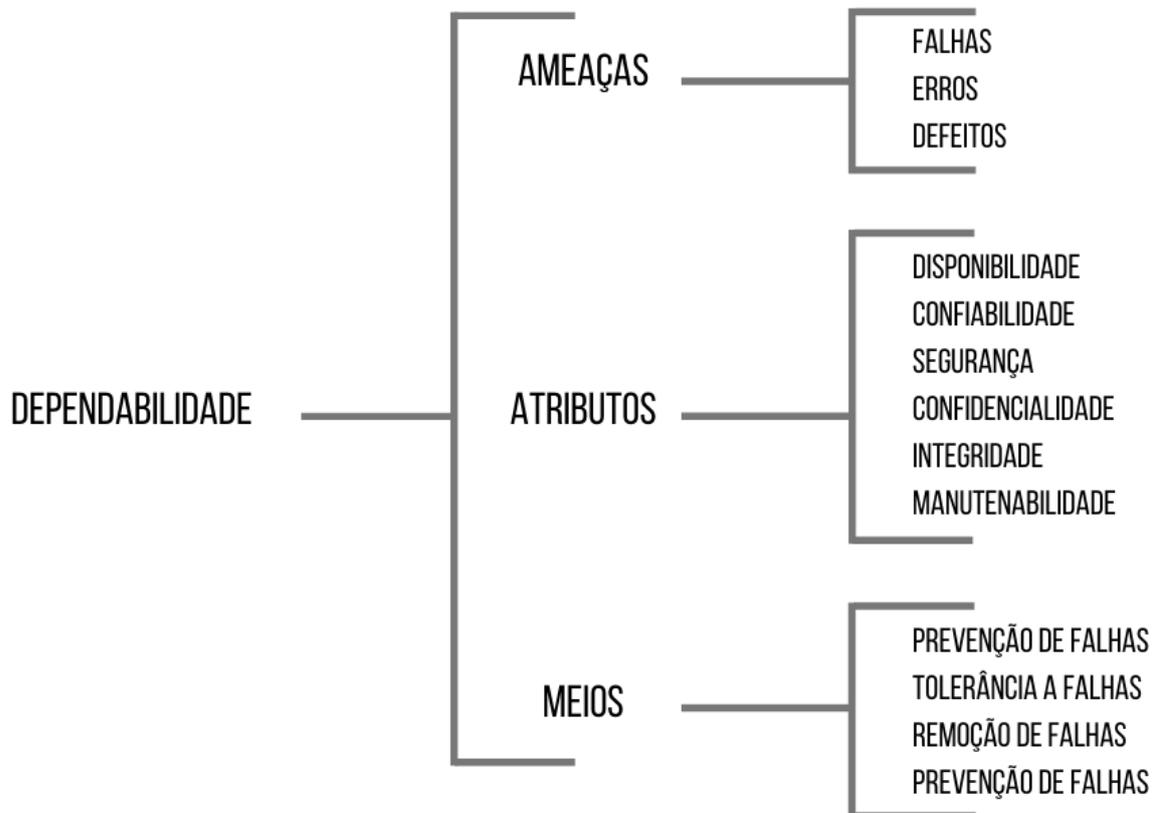
Passo 3 - Offloading Favorável: finalmente, é feita a análise para saber se a realização

da técnica é realmente favorável e se esta decisão está diretamente ligada com o objetivo do usuário (economia de bateria, menor tempo de execução, etc.). Esta etapa requer uma análise do contexto e de como os recursos vão ser utilizados para responder se o objetivo definido poderá ser alcançado ou se é melhor realizar a tarefa no próprio dispositivo móvel (UMAIR et al., 2015).

3.2 AVALIAÇÃO DE CONFIABILIDADE E DISPONIBILIDADE

A dependabilidade de um sistema de computação está relacionada com a sua capacidade de fornecer um serviço confiável (AVIZIENIS et al., 2001) e com a sua possibilidade de executar corretamente todas as funções do sistema, apesar de possíveis ameaças (ataques externos ou internos) e mau funcionamento do sistema (falhas de hardware, software e falhas humanas) (ZAMOJSKI; CABAN, 2006). A Figura 4 faz uma exposição sistemática dos três conceitos que estão envolvidos na definição de dependabilidade.

Figura 4 – Árvore de Dependabilidade



Fonte: A autora (2021)

Considerando os atributos básicos da dependabilidade, podemos defini-los como:

- **Disponibilidade:** a probabilidade de que um sistema pode ser usado para a finalidade a que se destina durante um intervalo de tempo especificado;
- **Confiabilidade:** a probabilidade de um sistema executar um serviço especificado durante um intervalo de tempo especificado;
- **Segurança:** é a probabilidade de um sistema não falhar de maneira a causar danos catastróficos durante um período especificado;
- **Confidencialidade:** ausência de divulgação não autorizada de informações;
- **Integridade:** ausência de alterações inadequadas do estado do sistema;
- **Manutenibilidade:** capacidade de realizar reparos e modificações (AVIZIENIS et al., 2001) (NICOL; SANDERS; TRIVEDI, 2004).

Em suas definições, disponibilidade e confiabilidade enfatizam a prevenção de falhas, elas podem ser agrupadas e definidas coletivamente como a prevenção ou minimização de interrupções de serviço (AVIZIENIS et al., 2001). Este trabalho foca nos conceitos relacionados à confiabilidade e disponibilidade, que são os aspectos do sistema que são analisados.

3.2.1 Confiabilidade

A confiabilidade é uma medida probabilística e é definida em termos de um dispositivo, que pode ser um componente de um sistema ou um sistema que consiste em muitos componentes, sendo o tempo um fator importante na definição de confiabilidade (KURO; ZUO, 2003), que pode ser definida como a probabilidade de um sistema funcionar (sem falhas) durante um intervalo de tempo (CLARK; PRADHAN, 1995).

Definição 1: A Confiabilidade pode ser definida como a probabilidade $R(t)$ de um item não falhar no intervalo de tempo $(0, t]$ ou, em outras palavras, a probabilidade de que o item sobreviva ao intervalo de tempo $(0, t]$ e ainda esteja funcionando no momento t . Dessa forma,

ela pode ser representada pela Equação 3.1, onde T é uma variável aleatória que representa o tempo de falha (RAUSAND; HØYLAND, 2003); (XIE; DAI; POH, 2004).

$$R(t) = P(T > t), t \geq 0 \quad (3.1)$$

A Equação 3.2 representa a probabilidade de ocorrência de falha.

$$F(t) = 1 - R(t) = P(T \leq t) \quad (3.2)$$

3.2.2 Disponibilidade

Disponibilidade é a probabilidade de um sistema estar operacional (não falhou) em um determinado instante no tempo (CLARK; PRADHAN, 1995). Ela está diretamente relacionada à proporção de tempo em que o sistema é considerado operacional, portanto, é importante alcançar alta disponibilidade para evitar as consequências nocivas da interrupção do serviço e até mesmo perdas financeiras (MATOS et al., 2014).

Definição 2: A Disponibilidade pode ser definida como a razão entre o tempo de funcionamento esperado do sistema e a soma dos tempos de funcionamento e de falha esperados (MACIEL et al., 2012), como está descrita na equação 3.3.

$$A = \frac{E[Uptime]}{E[Uptime] + E[Downtime]}, \quad (3.3)$$

onde:

- A é a disponibilidade estacionária do sistema;
- $E[Uptime]$ é o tempo de funcionamento esperado do sistema;
- $E[Downtime]$ é o tempo de falha esperado do sistema.

Quando não se conhece os tempos de funcionamento e de falha do sistema, é possível utilizar os valores médios entre os eventos de falha e reparo para calcular a disponibilidade, assim ela pode ser representada em função do *Mean Time do Repair* (MTTR) e do *Mean Time To Failure* (MTTF), como na Equação 3.4

$$A = \frac{MTTF}{MTTF + MTTR} \quad (3.4)$$

Assim, podemos calcular a indisponibilidade de um sistema através da Equação 3.5.

$$UA = 1 - A \quad (3.5)$$

Considerando a indisponibilidade de um sistema, podemos calcular o *downtime* num período de tempo (T) específico através da 3.6, o resultado vai indicar o tempo esperado que aquele sistema esteja indisponível.

$$Dt = UA \times T \quad (3.6)$$

3.3 TÉCNICAS DE MODELAGEM

A pesquisa em análise de dependabilidade levou a uma variedade de modelos, cada um focado em níveis específicos de abstração e/ou de características do sistema (NICOL; SANDERS; TRIVEDI, 2004). Por isso é importante conhecer bem o sistema e os modelos existentes para identificar quais modelos serão melhor empregados na análise de determinado aspecto do sistema.

Os tipos de modelo utilizados para dependabilidade estão divididos em duas categorias: modelos combinatoriais e modelos baseados na geração de espaço de estados. Os modelos combinatoriais incluem os diagramas de blocos de confiabilidade (*Reliability Block Diagram*), árvores de falhas (*Fault Tree*) e grafos de confiabilidade (*Reliability Graph*). Entre o segundo

tipo, estão as cadeias de Markov em tempo contínuo (*Continuous Time Markov Chain*) e modelos baseados em Redes de Petri (*Petri Nets*) (MALHOTRA; TRIVEDI, 1995).

O presente trabalho utilizou a técnica combinatorial de *Reliability Block Diagram* (RBD) e a técnica baseada em geração de espaços de estados *Continuous Time Markov Chain* (CTMC) para modelar o sistema em questão.

3.3.1 Diagramas de Bloco de Confiabilidade

Um diagrama de blocos de confiabilidade é frequentemente usado para representar a relação entre o funcionamento de um sistema e o funcionamento de seus componentes, em um RBD um retângulo ou um círculo é frequentemente usado para representar um componente. Ele não representa necessariamente como os componentes estão fisicamente conectados no sistema, mas ele representa a relação lógica entre o funcionamento do sistema e o funcionamento de seus componentes (KUO; ZUO, 2003).

Esta técnica é utilizada para análises de confiabilidade e consiste em um único bloco no diagrama para cada componente de hardware ou software. O RBD de um sistema representa o efeito da falha de um componente na performance do sistema (FORCHE, 1990). Ele é uma técnica de análise gráfica que expressa um determinado sistema por meio de blocos conectados de acordo com sua relação lógica de confiabilidade. As conexões em série representam a lógica “e” dos componentes, e as conexões paralelas representam a lógica “ou” (GUO; YANG, 2007). Sua representação gráfica é vantajosa por facilitar a sua compreensão.

O RBD aceita uma entrada E e uma saída S. Nele os elementos do sistema são conectados por arcos que refletem as relações entre os diferentes componentes e o sistema funciona se houver um caminho de sucesso entre E e S (BOUROUNI, 2013).

A abordagem utilizada pelo diagrama de blocos de confiabilidade é a de utilizar informações de disponibilidade de componentes e a de usar um modelo definido pelo usuário do equipamento ou sistema para aproximar a disponibilidade da saída do sistema. O modelo é baseado nas conexões físicas do equipamento e na maneira como certos componentes interagem (ROCZEN; ARNO; HALE, 2004).

Do ponto de vista da confiabilidade, um sistema pode ser considerado em série, em paralelo ou *standby* (BOUROUNI, 2013):

- **Sistema em série:** o sistema falha quando um ou mais de seus componentes falharem.

No caso de um sistema com vários elementos em série, a disponibilidade geral corresponde ao produto dos componentes. Um exemplo pode ser observado na Imagem 5.

Figura 5 – RBD de Sistema Representado em Série

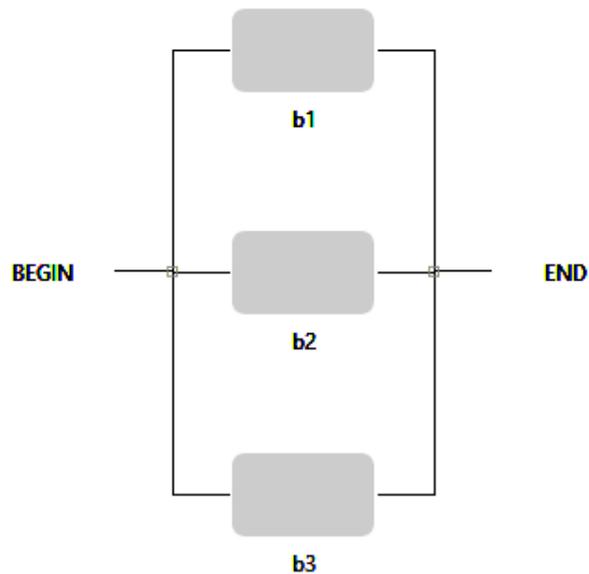
Fonte: A autora (2021)

udegraphics[width=.9]images/serie.PNG

Fonte: A autora (2021)

▪ **Sistema em paralelo:** quando um ou mais de seus componentes falham, o sistema não falha. A redundância ativa é alcançada pelos elementos paralelos, fornecendo as mesmas funções e trabalhando juntos. Um exemplo desse sistema, encontra-se na Imagem 6.

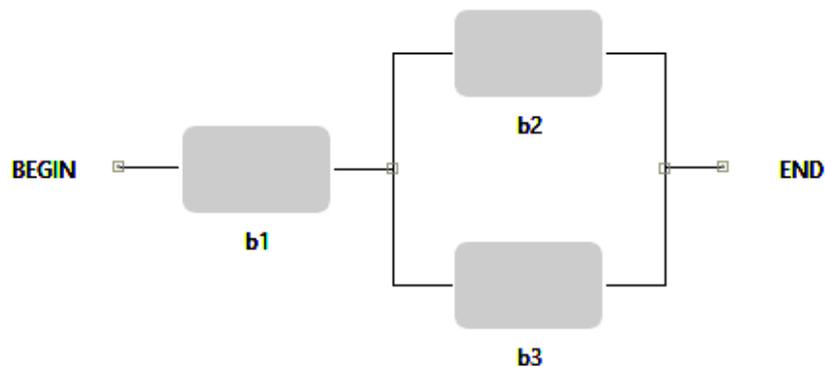
Figura 6 – RBD de Sistema Representado em Paralelo



Fonte: A autora (2021)

▪ **Sistema em *standby*:** quando vários componentes idênticos que executam a mesma função são usados, um componente está em operação enquanto os outros estão em reserva. A Imagem 7 traz um exemplo em que o sistema falha se o bloco b1 falhar, ou se ambos os blocos b2 e b3 falharem. Esse modelo introduz o conceito de redundância.

Figura 7 – RBD de Sistema Representado com Mecanismo de Redundância



Fonte: A autora (2021)

A redundância é utilizada para alcançar alta disponibilidade de serviço, já que ela oferece a oportunidade de recuperar rapidamente o serviço em uma unidade operacional, em vez de acumular uma interrupção de serviço mais longa, enquanto as falhas em um elemento autônomo são verificadas, reparadas e o serviço é restaurado. Uma maneira de fazer isso é através da duplicação de elementos críticos (BAUER; ADAMS; EUSTACE, 2011).

Quando um sistema apresenta alguma falha em um de seus componentes e possui algum mecanismo de redundância, o comportamento redundante pode ser apresentado das seguintes maneiras (RAM, 2019):

- **Cold Standby:** O elemento ou módulo de backup permanece desligado em espera e só será ativado se o elemento primário falhar. Um ponto positivo dessa técnica é que o elemento secundário possui baixo consumo de energia. Por outro lado, é necessário um tempo significativo para a ativação desse segundo módulo e as informações do primeiro módulo são perdidas sendo necessário refazer parte do trabalho.

- **Hot Standby:** Os módulos replicados são sincronizados com o módulo operacional, assim, tantos os elementos ativos quanto os que estão em espera são vistos pelo usuário final como um único recurso. Após a falha de um módulo, o módulo secundário é ativado automaticamente e os usuários que acessam o módulo primário agora acessarão o módulo secundário sem perceber a troca de equipamento.

- **Warm Standby:** Esta técnica tenta equilibrar os custos e o atraso do tempo de recupe-

ração das técnicas *Cold* e *Hot Standby*. O módulo secundário fica em espera, mas não completamente desligado, portanto pode ser ativado mais rápido do que na técnica *Cold Standby*. O módulo replicado é parcialmente sincronizado com o módulo operacional, de modo que os usuários que acessam informações no módulo operacional podem perder algumas informações que estavam sendo gravadas perto do momento em que o módulo primário falhou.

Assim, considerando uma estrutura em série composta por n componentes independentes, a disponibilidade A_S do sistema pode ser representada pela Equação 3.7 onde A_i representa a disponibilidade do componente i .

$$A_S = \prod_{i=1}^n A_i \quad (3.7)$$

Considerando esta mesma estrutura em série, a confiabilidade $R(t)_S$ do sistema pode ser representada pela Equação 3.8, onde $R_i(t)$ representa a confiabilidade do componente i (MACIEL et al., 2012).

$$R(t)_S = \prod_{i=1}^n R_i(t) \quad (3.8)$$

Considerando agora uma estrutura paralela composta de n componentes idênticos e independentes, a disponibilidade A_S do sistema pode ser representada pela Equação 3.9 onde A_i representa a disponibilidade do componente i .

$$A_S = \prod_{i=1}^n (1 - A_i) \quad (3.9)$$

Considerando esta mesma estrutura paralela, a confiabilidade R_S do sistema, pode ser representada pela Equação 3.10, onde $R_i(t)$ representa a confiabilidade do componente i (MACIEL

et al., 2012).

$$R_S = \prod_{i=1}^n 1 - R_i(t) \quad (3.10)$$

3.3.2 Cadeias de Markov de Tempo Contínuo

Introduzidas pela primeira vez por Andrey Markov em 1907, as cadeias de Markov são utilizadas na análise de desempenho desde cerca de 1950, e hoje elas são amplamente utilizadas na análise de desempenho e confiabilidade. Uma cadeia de Markov consiste em um conjunto de estados e em um conjunto de transições entre os estados. Um estado da cadeia de Markov pode modelar várias condições de interesse no sistema que está sendo estudado (BOLCH et al., 2006).

Para apresentar a definição de cadeia de Markov de tempo contínuo, primeiramente será feita a definição de processo estocástico:

Definição 3: Um processo estocástico é definido como uma família de variáveis aleatórias $\{X_t : t \in T\}$ onde cada variável aleatória X_t é indexada pelo parâmetro $t \in T$, que geralmente é chamado de parâmetro de tempo se $T \subseteq R_+ = [0, \infty)$. O conjunto de todos os valores possíveis de X_t (para cada $t \in T$) é conhecido como o espaço de estado S do processo estocástico.

As Cadeias de Markov são um tipo especial de processo estocástico que possui a propriedade de Markov, comumente utilizadas para modelar sistemas com vários estados e as transições entre esses estados.

Definição 4: Uma Cadeia de Markov é um processo estocástico $\{X(t), t \geq 0\}$ que possui a propriedade de Markov. A variável aleatória $X(t)$ denota o estado do processo no tempo t .

Em relação ao tempo, ele pode ser discreto, assumindo valores em $(0, 1, 2, \dots)$ ou contínuo. Quando o tempo é discreto, temos uma Cadeia de Markov de Tempo Discreto (DTMC) e quando o tempo é contínuo, temos uma Cadeia de Markov de Tempo Contínuo (CTMC)

(RAUSAND; HØYLAND, 2003).

Definição 5: Um processo estocástico $\{X_t : t \in T\}$ constitui uma CTMC se para todo $t_i \in R_0^+$, com $0 = t_0 < t_1 < \dots < t_n < t_n + 1, \forall n \in N$, e $\forall s_i \in S = N_0$ a seguinte relação presente na Equação 3.11 se mantém:

$$P(X_{t_{n+1}} = s_{n+1} | X_{t_n} = s_n, X_{t_{n-1}} = s_{n-1}, \dots, X_{t_0} = s_0) = P(X_{t_{n+1}} = s_{n+1} | X_{t_n} = s_n) \quad (3.11)$$

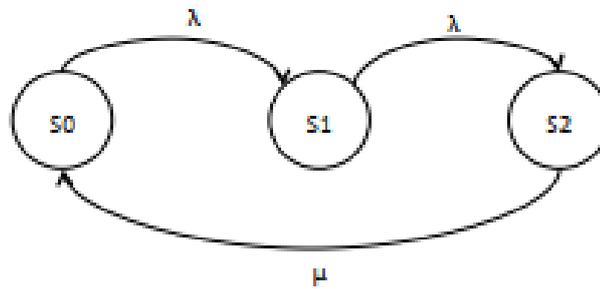
A propriedade markoviana acima define que a transição para o próximo estado do sistema depende apenas do seu estado atual, haja vista que os estados anteriores em que o sistema esteve não têm influência e são esquecidos após a transição de estados.

Para evitar a especificação de CTMCs diretamente no nível de estados, técnicas de especificação de modelo de alto nível foram desenvolvidas, exemplos disso são as redes de Petri estocásticas, redes de atividade estocástica e álgebras de processos estocásticos (BAIER et al., 2003).

É possível representar uma CTMC graficamente através de um diagrama de estados, como é mostrado na Figura 8, onde cada círculo representa um estado do sistema e cada arco representa uma transição de um estado para o outro. Cada arco possui taxas que representam algum aspecto do sistema. Assim, a CTMC representada na imagem possui três estados $S = \{S_0, S_1, S_2\}$ e duas taxas $T = \{\lambda, \mu\}$.

Uma CTMC pode ser representada em forma de uma matriz de taxa de transição Q , onde Q é uma matriz quadrada que possui dimensão igual ao número de estados no espaço de estados S . Os elementos da matriz representam as taxas de transição entre os estados da cadeia, os elementos q_{ij} representam a taxa partindo de i e chegando ao estado j . Elementos da diagonal q_{ii} são definidos como o somatório dos elementos da mesma linha, multiplicado por 1. A matriz que representa a cadeia da Figura 8, é:

Figura 8 – Cadeia de Markov de Tempo Contínuo



Fonte: A autora (2021)

$$\begin{pmatrix} -\lambda & \lambda & 0 \\ 0 & -\lambda & \lambda \\ \mu & 0 & -\mu \end{pmatrix}$$

Métricas não dependentes do tempo são obtidas por meio de análise estacionária, que consiste em encontrar a probabilidade do sistema ser localizado em um determinado estado considerando um longo tempo de execução. As probabilidades são representadas pelo vetor $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, onde π representa a probabilidade estacionária para o estado i . Essas probabilidades não dependem do estado inicial do sistema. Considerando que a cadeia de Markov em questão é uma cadeia ergódica (quando é possível alcançar qualquer estado a partir de qualquer outro estado, em um número n finito de passos), é possível encontrar as probabilidades estacionárias através do sistema linear formado pelas Equações 3.12 e 3.13 (BOLCH et al., 2006).

$$\pi Q = 0 \tag{3.12}$$

$$\sum_{i=1}^n \pi_i = 1 \tag{3.13}$$

Onde a Equação 3.12 produz as equações de balanço da cadeia de Markov e significa que a soma do fluxo de entrada em um determinado estado com o fluxo de saída deve ser igual a zero. A Equação 3.13 diz que as probabilidades estacionárias são mutuamente exclusivas e exaustivas, portanto, a soma de todas elas devem ser igual a 1.

Métricas dependentes do tempo são obtidas por meio de avaliação transiente, que permite descobrir o que acontece no sistema levando em consideração um determinado período de tempo ao como um sistema se comporta antes de atingir seu estado estacionário. O vetor $\pi(t) = \{\pi_1(t), \pi_2(t), \dots, \pi_n(t)\}$ representa a probabilidade de estado transiente da CTMC no tempo t . O comportamento da CTMC pode ser descrito pela Equação de Kolmogorov 3.14 dado o vetor de probabilidade inicial $\pi(0)$. A Equação 3.15 fornece o tempo total esperado que o CTMC passe no estado i no intervalo $[0, t)$ (BOLCH et al., 2006).

$$\frac{d\pi(t)}{dt} = \pi(t) \times Q \quad (3.14)$$

$$\sum_{i=1}^n \pi_i(t) = 1 \quad (3.15)$$

3.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou alguns dos conceitos que fundamentam nosso trabalho e que são necessários para seu entendimento. Primeiramente foram trazidos conceitos relacionados á computação em nuvem móvel para em seguida introduzir as definições referentes à técnica de *offloading* para que seja possível entender melhor seu funcionamento e sua importância, assim como pontos que podem influenciar no desempenho da técnica ou da disponibilidade e confiabilidade do sistema. Em seguida são apresentados conceitos associados à avaliação de dependabilidade, focando nos atributos de disponibilidade e confiabilidade para fundamentar

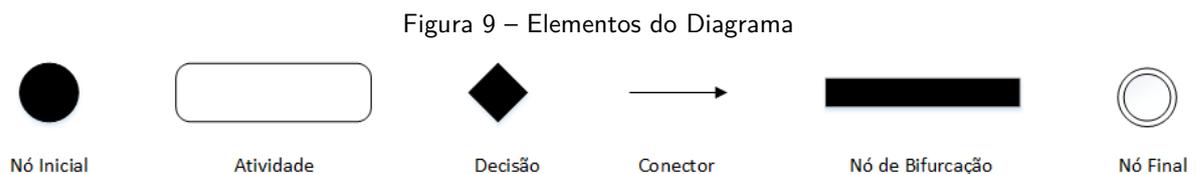
as avaliações que foram realizadas no trabalho. Por fim, foram definidos os formalismos de RBD e CTMC, que foram utilizados para realizar as modelagens que compõem o trabalho.

4 METODOLOGIA E ARQUITETURA

Este capítulo apresenta a metodologia utilizada para o desenvolvimento desta dissertação e a arquitetura dos sistemas utilizados nos experimentos realizados. Inicialmente, é apresentada a metodologia proposta, quando são detalhadas as etapas necessárias para atingir os objetivos do trabalho. Por fim, é apresentada a arquitetura do experimento que compõe esse trabalho.

4.1 METODOLOGIA PROPOSTA

Esta seção apresenta detalhes sobre a metodologia que foi proposta e adotada neste trabalho. A Figura 9 mostra os elementos utilizados no Diagrama de Atividades e o significado de cada um deles.



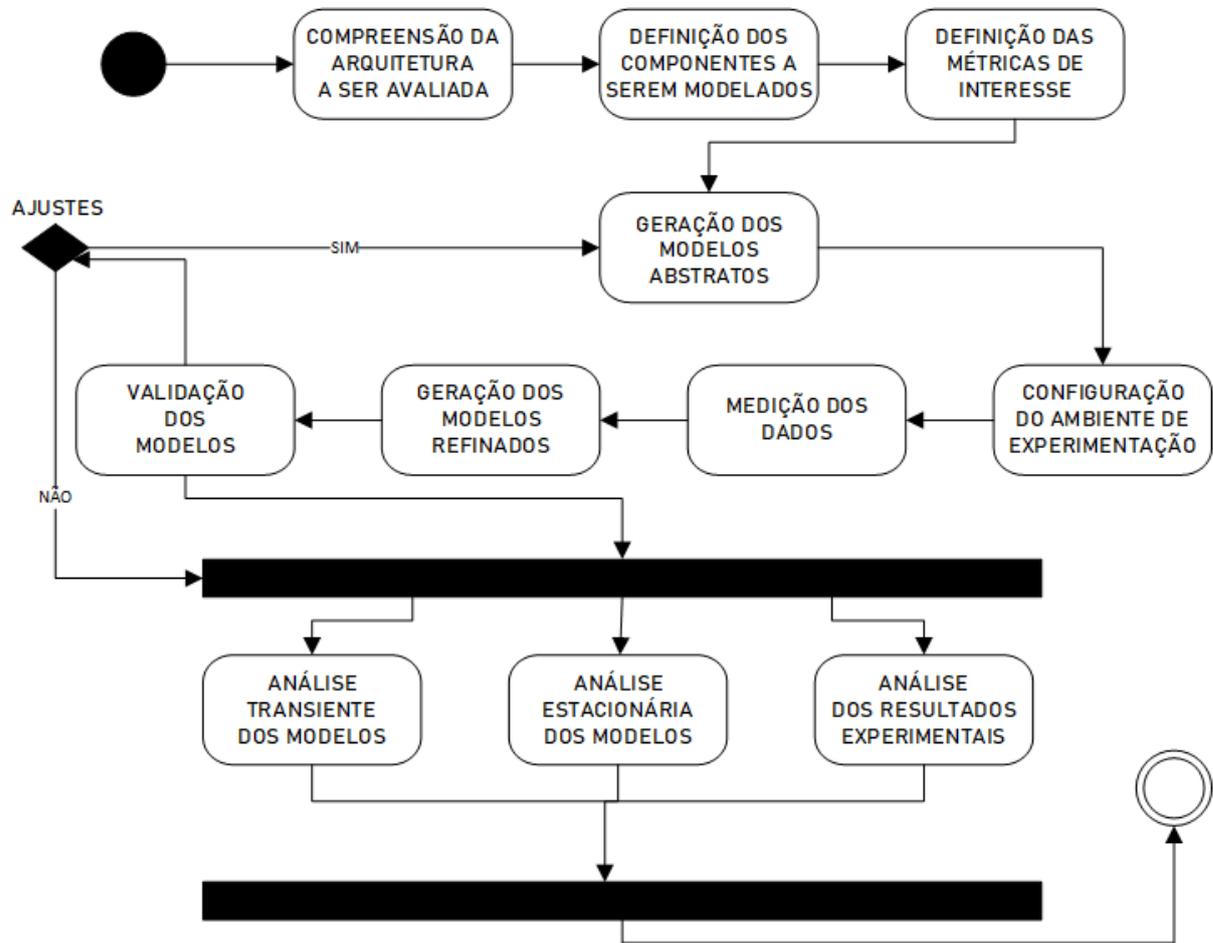
Fonte: A autora (2021)

A metodologia adotada para essa dissertação pode ser observada na Figura 10, que apresenta um Diagrama de Atividades na notação *Unified Modeling Language* (UML) (BOOCH; RUMBAUGH; JACOBSON, 2006). O diagrama exibe a ordem em que as atividades foram realizadas.

Cada elemento de atividade do diagrama representa uma etapa que serão melhor explanadas a seguir:

- **Compreensão da arquitetura a ser avaliada:** ao se decidir analisar o uso da técnica de *offloading* se fez necessário entender a arquitetura de computação em nuvem móvel, quais os elementos importantes nela e quais deles são críticos para seu funcionamento. O processo realizado nesta etapa diz respeito ao estudo e entendimento da arquitetura escolhida e ao funcionamento da técnica de *offloading* dentro desta arquitetura. Nesta etapa foi possível perceber que o dispositivo móvel compõe uma camada importante

Figura 10 – Metodologia Proposta



Fonte: A autora (2021)

da arquitetura, sua disponibilidade é necessária para que tudo funcione, já que sem ele não é possível se conectar as redes móveis e acessar os serviços disponíveis na nuvem. Já dentro do contexto do dispositivo móvel, notou-se que a bateria é um componente crítico que pode afetar diretamente a experiência do usuário durante o uso, portanto encontrar maneiras de otimizar seu uso e gerar uma economia no consumo de energia, pode trazer diversos benefícios aos usuários;

- **Definição dos componentes a serem modelados:** na etapa em questão são considerados os componentes que possuem maior relevância para a pesquisa. O dispositivo móvel é parte fundamental dentro da arquitetura de computação em nuvem, pois sem ele não é possível utilizar os serviços que o paradigma oferece, fato que motivou a modelagem do dispositivo móvel. Como o foco da pesquisa é voltado para o consumo energético gerado pelo uso de diferentes redes móveis, a modelagem do descarregamento da bateria do dispositivo móvel em função da rede móvel utilizada também possui grande relevância,

já que o consumo energético muda de acordo com a rede móvel que está sendo utilizada. Dentre as redes móveis existentes, foram selecionadas as mais utilizadas hoje em dia, sendo elas Wi-Fi, 4G e 3G, permitindo assim a obtenção de cenários em que o uso de determinada rede móvel pode acabar gerando um consumo energético maior ou fazer com que a troca de dados necessária para realizar o *offloading* ocorra mais lentamente ou mais rapidamente, gerando dados que tornem possível a comparação entre cenários;

- **Definição das métricas de interesse:** nesta etapa são definidas as métricas adotadas para validação e experimentação dos modelos. Confiabilidade, disponibilidade e *downtime* são as métricas adotadas de acordo com o atributo a ser avaliado, assim é possível entender os cenários que fornecem uma maior confiabilidade e disponibilidade da bateria ou do dispositivo móvel, ajudando a otimizar seu uso e identificar pontos críticos. Além disso também é levada em consideração a redundância do sistema, que é bastante utilizada em sistemas que utilizam o paradigma de computação em nuvem. As métricas escolhidas mostram pontos e componentes que são críticos durante a utilização do sistema, facilitando a identificação de gargalos e pontos que precisam de melhoria ou que podem ser aprimorados para melhorar a utilização do sistema e aplicação da técnica sendo estudada;
- **Geração dos modelos abstratos:** os processos dessa etapa unem informações e decisões que foram tomadas nas etapas anteriores, essa etapa está relacionada com a modelagem do sistema. Para gerar os modelos abstratos são consideradas a arquitetura da MCC, os componentes a serem modelados e as métricas a serem analisadas. Para os modelos de confiabilidade, foi definido que a modelagem seria feita através da adoção de CTMC, são gerados também os modelos de disponibilidade do sistema, para tal, é adotada a modelagem através de CTMCs e RBDs. A escolha de RBD se deu, dentre outros motivos, pela facilidade que ele fornece para realizar a análise da disponibilidade do sistema e por permitir o cálculo de disponibilidade através de fórmulas fechadas. A escolha de CTMC foi influenciada por sua característica de gerar modelos baseados em estados e por permitir modelar o comportamento dinâmico do sistema. Os detalhes sobre a concepção dos modelos, ferramentas e notações utilizadas e detalhes sobre a modelagem se encontram no Capítulo 5;
- **Configuração do ambiente de experimentação:** o experimento de medição com o

objetivo de obter dados do sistema real para serem utilizados na validação e na análise dos modelos é definido e organizado nesta etapa. Definir a configuração do ambiente é muito importante, pois, é necessário ajustar o sistema e verificar o efeito de cada configuração no desempenho ao avaliar o sistema real (PINHEIRO et al., 2019). Este processo envolve a definição das ferramentas a serem utilizadas, as redes móveis e os dados a serem obtidos do sistema real. O experimento de medição leva em consideração o consumo de energia no dispositivo móvel gerado pela rede móvel utilizada no processo de realização de *offloading*. A arquitetura do *framework* e do aplicativo utilizados no experimento se encontram nas Seções 4.2 e 4.3, respectivamente. A configuração do ambiente é detalhada no Capítulo 6;

- **Medição dos dados:** depois de configurar o ambiente de experimentação, são realizados os experimentos, porém, para que seja possível obter os dados do sistema real, é necessário realizar a medição destes dados. Nesta etapa são definidas e configuradas as ferramentas de medição a serem utilizadas durante a realização do experimento; com tudo pronto, o experimento pode ser realizado e os dados medidos para serem utilizados posteriormente. Foi definido que seriam medidos o tempo que leva para a tarefa ser realizada e o consumo energético gerado durante esse tempo, o tempo é informado pelo aplicativo de edição de imagem utilizado, para a medição do consumo energético foi necessário buscar e testar várias aplicações que realizam esse trabalho para identificar a melhor para o experimento;
- **Geração dos Modelos Refinados:** com todas as etapas anteriores realizadas, foi possível então gerar os modelos refinados que já foram previamente definidos, para isso foi escolhida a ferramenta Mercury (MACIEL et al., 2017) para ser utilizada tanto na geração dos modelos como na análise dos mesmos. Essa fase envolve também os ajustes necessários para que os modelos funcionem corretamente;
- **Validação dos modelos:** nesta etapa estão envolvidos os processos necessários para utilização dos dados obtidos no experimento na validação dos modelos concebidos, para tal, é necessário tratar os dados obtidos através de conversão, cálculo de taxas, cálculo de desvio padrão e intervalo de confiança. Em seguida é analisado e definido o método para validação que seja mais adequado para o cenário. Por fim, os dados são utilizados para validar os modelos concebidos;

- **Análise transiente dos modelos:** esta atividade consiste em utilizar os dados obtidos no experimento de medição para realizar a análise transiente dos modelos de confiabilidade que foram gerados anteriormente, esse processo envolve a obtenção dos valores das métricas de interesse que também foram definidas anteriormente;
- **Análise estacionária dos modelos:** semelhantemente a etapa anterior, é realizada a etapa atual, que consiste em utilizar os dados obtidos no experimento para realização da análise estacionária dos modelos de disponibilidade que foram concebidos anteriormente, obtendo também as métricas de interesse que foram previamente definidas;
- **Análise dos resultados experimentais:** em paralelo com as duas etapas descritas anteriormente, é realizada esta última etapa do fluxograma, que consiste em analisar os resultados obtidos com o experimento de medição e em levantar pontos importantes em relação ao comportamento do sistema real durante os experimentos, podendo, através disto, constatar pontos a serem melhorados e pontos de sucesso. Através desta etapa é possível fazer inferências e ressaltar pontos críticos e gargalos que porventura sejam identificados ao realizar os experimentos.

4.2 ARQUITETURA DO AMBIENTE DE EXPERIMENTAÇÃO

Esta seção apresenta a arquitetura do sistema que foi utilizado para realização do experimento de medição neste trabalho. Com o objetivo de obter dados do sistema real, o experimento foi configurado utilizando ferramentas que realizam o processo de *offloading* em dispositivos móveis, sendo elas um *framework* que realiza *offloading*, e um aplicativo móvel de edição de imagens que trabalha juntamente com o *framework*.

O *framework* para *offloading* adotado foi o MpOS (Multiplatform Offloading System) (MPOS, 2020), desenvolvido por um grupo de pesquisa da Universidade Federal do Ceará, e que tem como principal objetivo fornecer suporte às operações de *offloading* em múltiplas plataformas. Ele foi desenvolvido em 2015 e permite operações de *offloading* em dispositivos móveis que utilizam o sistema *Android* e *Windows Phone*.

Existem diferentes abordagens que são utilizadas em vários *frameworks* que foram desenvolvidos. A maioria das soluções diferem principalmente no que diz respeito a granularidade do *offloading*, plataforma móvel suportada, linguagem de programação e tomada de decisão para o *offloading* (REGO et al., 2017). Os autores do MpOs fizeram um levantamento dos trabalhos

relacionados com o desenvolvimento de *frameworks* para *offloading* ((MARCH et al., 2011), (KOVACHEV; YU; KLAMMA, 2012), (MA; LAM; WANG, 2011), (KAKADIA; SARIPALLI; VARMA, 2013), (CHUN et al., 2011), (GORDON et al., 2012), (VERBELEN et al., 2012), (KRISTENSEN; BOUVIN, 2010), (CUERVO et al., 2010), (KOSTA et al., 2012), (KEMP et al., 2010), (FERRARI; GIORDANO; PUCCINELLI, 2016), (COSTA et al., 2015)). A partir deste levantamento foi possível fazer uma comparação entre os *frameworks* levando em consideração os três pontos citados anteriormente. A Tabela 2 apresenta essa comparação.

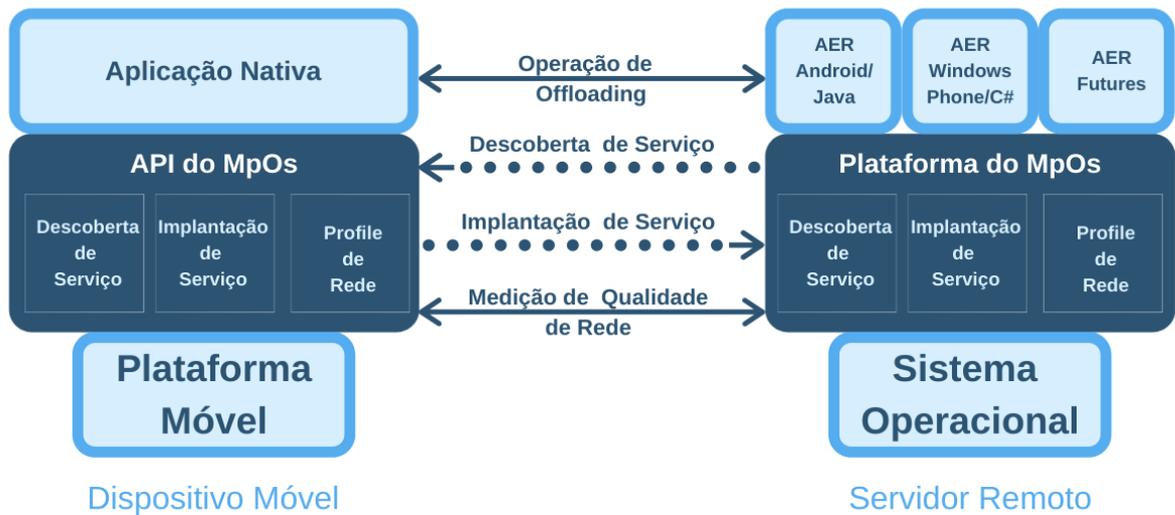
Tabela 2 – Trabalhos Relacionados ao MpOs

Abordagem	Decisão pelo Offloading	Granularidade	Plataforma Suportada/ Linguagem de Programação
<i>μCloud</i>	Estática	Componente	Android
MACS	Dinâmica	Componente	Android
eXCloud	Dinâmica	Aplicativo Inteiro	JamVM / Iphone 4
MECCA	Dinâmica	Aplicativo Inteiro	Android
CloneCloud	Dinâmica	Thread	Android
COMET	Dinâmica	Thread	Android
AIOLOS	Dinâmica	Método	Android
Scavenger	Dinâmica	Método	Python
MAUI	Dinâmica	Método	Windows Phone
ThinkAir	Dinâmica	Método	Android
Cuckoo	Dinâmica	Método	Android
ARC	Dinâmica	Método	Android
MpOs	Dinâmica e Estática	Método	Android e Windows Phone

Fonte: Adaptado de (REGO et al., 2017)

A arquitetura do *framework* MpOS segue o modelo cliente/servidor padrão como vários outros *frameworks* de computação em nuvem móvel. Através da Figura 11 é possível observar uma visão geral dessa arquitetura e das interações que ocorrem entre suas camadas. Também observamos que do lado do cliente (dispositivo móvel), está localizado o componente API MpOS, que será executado no dispositivo móvel e fará a conexão com o lado servidor onde está localizada a Plataforma MpOS. Os dois lados possuem as mesmas três camadas (COSTA et al., 2015).

Figura 11 – Arquitetura do MpOS



Fonte: Adaptado de (COSTA et al., 2015)

Juntamente com o MpOS, foi utilizado um aplicativo de edição de imagem para executar o lado cliente do *framework*. Vale considerar que sua escolha se deu pelo fato que as aplicações que fazem manipulação de imagens geram uma carga maior de trabalho para o dispositivo móvel, o que demanda um poder maior de processamento, tornando possível observar com mais facilidade os efeitos e resultados do uso do *offloading*.

BenchImage é um aplicativo móvel que usa algoritmos de processamento de imagem para aplicar efeitos (filtros) em fotos. O aplicativo, que pode ser observado na Figura 12, disponibiliza diversas fotos no formato JPEG com diferentes resoluções, desde 0,3 MP a 8 MP. É possível que o usuário escolha uma dessas fotos armazenadas e aplique um dos filtros disponíveis nelas, filtros esses que são o Cartoonizer, Sharpen e Red Ton (COSTA et al., 2015).

Desenvolvido pela mesma equipe que desenvolveu o MpOS, o aplicativo possui a opção de ser utilizado juntamente com o *framework* MpOs, permitindo que o usuário escolha entre realizar as tarefas de edição de imagem no próprio dispositivo ou faça uso da técnica de *offloading* ao enviar o processamento das tarefas para uma nuvem. Ao final da execução, o aplicativo apresenta informações sobre a tarefa realizada como o tempo total levado para executar as tarefas, tamanho da imagem, filtro aplicado e local de processamento da tarefa.

Para executar o lado servidor do MpOS, foi utilizada a plataforma de *cloud computing*

Figura 12 – Tela Original do BenchImage



Fonte: Adaptado de (COSTA et al., 2015)

Amazon Elastic Compute Cloud (Amazon EC2) (AMAZON, 2020). A plataforma foi utilizada para configurar uma máquina virtual, onde o servidor de aplicação do MpOS poderia ser executado e ficar pronto para receber as solicitações feitas pelo aplicativo BenchImage.

4.3 CONSIDERAÇÕES FINAIS

Este capítulo apresentou a metodologia proposta para a modelagem e análise do uso da técnica de *offloading* em dispositivos móveis. Inicialmente foi apresentada a metodologia proposta para realização do trabalho, detalhando as atividades realizadas em cada uma das etapas. Em seguida, foi apresentado o planejamento de experimento e a avaliação dos efeitos de cada fator do experimento. Por fim foi apresentada a arquitetura do ambiente de experimentação e dos sistemas que a compõe.

5 MODELAGEM

O capítulo atual apresenta os modelos que foram concebidos para o desenvolvimento deste trabalho, sendo eles os modelos de disponibilidade e confiabilidade que foram elaborados para representar o dispositivo móvel e a bateria do dispositivo móvel, além de modelos contendo componentes redundantes. O desenvolvimento destes modelos torna possível a realização de análises transientes e estacionárias do sistema.

A princípio são feitas algumas considerações sobre a arquitetura do sistema que os modelos representam, as métricas que foram adotadas e as ferramentas utilizadas para desenvolvimento e análise dos modelos. Em seguida são apresentados os modelos que foram confeccionados, assim como o detalhamento de seus componentes, os parâmetros utilizados e as equações utilizadas para obtenção das métricas definidas.

5.1 CONSIDERAÇÕES INICIAIS

Os modelos propostos são baseados na arquitetura de MCC, que foi exibida na Figura 11, sendo ela composta por três camadas interligadas entre si, que são: dispositivos móveis, serviços de redes móveis e serviço de nuvem, cada uma dessas camadas são detalhadas na Seção 3.1.2.1. Levando em consideração essa arquitetura, foi feita a modelagem do dispositivo móvel, que é uma das camadas necessárias para que o sistema funcione.

Dispositivos móveis são compostos por diversos elementos que se comunicam entre si e trabalham em conjunto para que todo o sistema possa funcionar. Neste trabalho a modelagem do dispositivo móvel é feita em alto nível, utilizando RBD, com blocos representando seus componentes. Dentre os componentes do dispositivo móvel, foi confeccionado um modelo mais detalhado da bateria, que é um componente crítico e fundamental para a disponibilidade do dispositivo.

O Diagrama de Bloco de Confiabilidade é utilizado para representar o dispositivo móvel e estimar sua disponibilidade e o seu *downtime*. A disponibilidade é calculada através da Equação 3.4 e o *downtime* através da Equação 3.6. Além disso, RBD é utilizado na confecção do modelo contendo mecanismo de redundância em *hot standby*, e um modelo hierárquico é utilizado para fazer a representação da redundância em *cold standby*, sendo assim possível estimar o aumento na disponibilidade do sistema com a presença da redundância.

Cadeia de Markov de Tempo Contínuo é utilizada para representar o descarregamento da bateria, tanto considerando o descarregamento de acordo com a rede móvel utilizada como o descarregamento quando não há utilização de nenhuma rede móvel. Esses modelos possibilitam que sejam estimados a confiabilidade, disponibilidade e *downtime* da bateria do dispositivo móvel. Sendo a confiabilidade calculada através da Equação 3.1, a disponibilidade através da Equação 3.4 e o *downtime* através da Equação 3.6. Nas figuras que mostram os modelos de RBD que foram concebidos, os blocos que estão na cor cinza possuem subsistemas que são representados com mais detalhes em um outro modelo.

Os modelos presentes neste trabalho foram todos desenvolvidos e analisados através da ferramenta Mercury (OLIVEIRA et al., 2017), portanto, as equações apresentadas neste capítulo para cálculo da disponibilidade, confiabilidade e *downtime* utilizam a sintaxe adotada pelo Mercury.

5.2 MODELOS

Esta seção apresenta os modelos que foram implementados neste trabalho, a descrição de seus componentes e elementos, as métricas aplicadas em cada um deles e as equações utilizadas em suas análises.

5.2.1 Modelo de Disponibilidade do Dispositivo Móvel

O dispositivo móvel compõe toda uma camada da arquitetura de MCC, o que significa que para que ela funcione, ele precisa estar disponível, sendo então um elemento fundamental no sistema de nuvem móvel. Levando em consideração o emprego da técnica de *offloading*, ele só é possível se houver um dispositivo móvel disponível e funcionando sem interrupções. Tendo em conta sua importância, foi desenvolvido um modelo em alto nível para representá-lo.

O modelo RBD, que representa a disponibilidade do dispositivo móvel, foi baseado no modelo proposto por (OLIVEIRA et al., 2013); a mudança realizada é que o bloco que representa a aplicação do dispositivo, presente no modelo original, nesta primeira representação foi omitido e foi então inserido um bloco representando a bateria do dispositivo, já que no caso do presente estudo a disponibilidade da bateria tem grande importância e influência na disponibilidade geral do dispositivo. O bloco de aplicação é utilizado apenas no modelo em que é inserida a redundância através de um modelo hierárquico, que será apresentado na Seção 5.2.2.1.

O modelo proposto se encontra na Figura 13, e aborda os principais componentes que compõem um dispositivo móvel. Como a representação é feita em um alto nível, os blocos abrangem diversos componentes de *software* e *hardware* juntos. Assim, o modelo consiste em três componentes: o primeiro (*mobile_hw*) representando os componentes de *hardware* do dispositivo exceto a bateria, o segundo (*mobile_os*) representando o sistema operacional do dispositivo móvel e, por fim, o terceiro bloco (*mobile_battery*) que representa a bateria do dispositivo.

Figura 13 – Modelo RBD do Dispositivo Móvel



Fonte: A autora (2021)

Os valores de MTTF e MTTR dos blocos *mobile_hw* e *mobile_os* são obtidos na literatura do trabalho de (OLIVEIRA et al., 2013), e o bloco *mobile_battery* utiliza os valores obtidos através das análises realizadas com o modelo CTMC, apresentada na Seção 5.2.3, que representa com mais detalhes o componente da bateria. A disponibilidade do dispositivo móvel é calculada através da Equação 5.1, seu resultado é obtido através da multiplicação dos valores de disponibilidade de cada um dos blocos.

$$A = A_{mobile_hw} \times A_{mobile_os} \times A_{mobile_battery} \quad (5.1)$$

5.2.2 Modelo de Disponibilidade do Dispositivo Móvel com Presença de Redundância

Esta seção apresenta os modelos que foram desenvolvidos com o objetivo de representar o dispositivo móvel com a presença de mecanismos de redundância. Para isso foram desenvolvidos

dois modelos baseados em RBD e um modelo baseado em CTMC. Para este trabalho, foram considerados os métodos de redundância *Cold Standby* e *Hot Standby*.

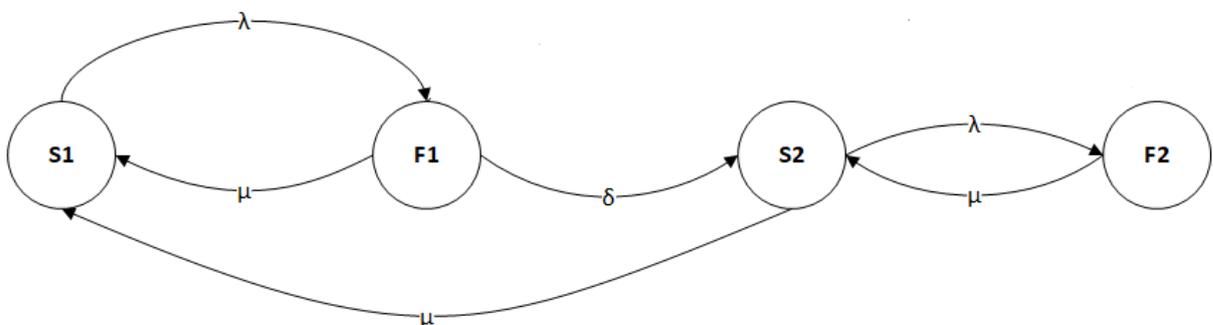
O uso de redundância permite a continuidade das operações no caso de falhas de componentes, isso é feito através de um sistema de *backup* que pode ser uma peça duplicada, um sistema ou um local alternativo, além disso, pode haver apenas um componente duplicado ou vários componentes (SCHMIDT, 2006). Considerando a importância da redundância no aumento da disponibilidade, os modelos incluindo redundância foram desenvolvidos para fornecer uma análise do dispositivo móvel em situações em que existem um *backup* para o servidor que realiza o *offloading* em casos de falha, prática comum utilizada em servidores no geral.

5.2.2.1 Cold Standby

Para realizar o *offloading* é necessário utilizar um servidor que irá realizar o processamento das tarefas que foram enviadas de um dispositivo móvel, porém para que isso ocorra é necessário que o servidor esteja disponível durante todo o processo. Uma estratégia utilizada por servidores é a de utilizar redundância para aumentar a disponibilidade do sistema.

O modelo de disponibilidade baseado em CTMC, que representa um servidor com um sistema de redundância foi baseado no modelo existente em (SOUSA et al., 2017), pode ser observado na Figura 14, nela está representado a redundância Cold Standby. Neste tipo de redundância, é assumido que o sistema principal está em funcionamento enquanto o de *backup* está em espera, sendo ativado apenas quando ocorre falha no sistema principal.

Figura 14 – Modelo Cold Standby do Servidor



Fonte: A autora (2021)

Os estados que compõem o modelo são: S1, F1, S2 e F2 e as transições entre os estados

possuem as taxas λ , μ e δ , que recebem respectivamente os seguintes valores: $\frac{1}{MTTF}$, $\frac{1}{MTTR}$ e $\frac{1}{MTTA}$, onde os valores de MTTF, MTTR e MTTA do servidor são herdados de (SOUSA, 2015).

O estado S1 representa o sistema principal em funcionamento, e caso ocorra falha neste sistema, a transição λ será ativada. Ao estar no estado F1, existem duas possibilidades: a primeira é a ocorrência do reparo do sistema principal e a ativação da transição μ ; a segunda é a ocorrência do chaveamento para que o sistema de *backup* seja ativado e a ativação da transição δ . O estado S2, representa o sistema de *backup* em funcionamento, na ocorrência de falha a transição λ é ativada, se o sistema principal for reparado a transição μ será ativada. Finalmente, estando no estado F2, se houver a ocorrência de reparo do sistema de *backup* a transição μ será ativada.

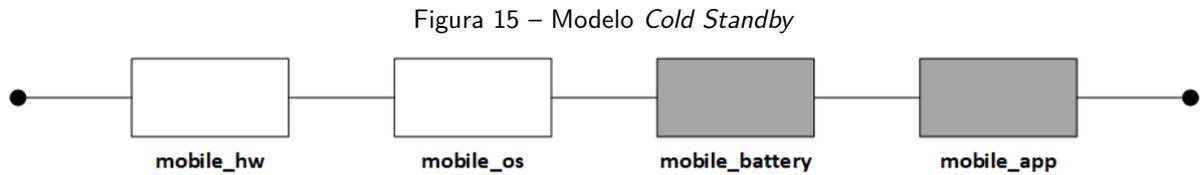
O modelo é utilizado para obter valor da disponibilidade através da Equação 5.2, que é escrita de acordo com a notação utilizada pela ferramenta Mercury, onde $P\{S1\}$ representa a probabilidade do sistema principal estar em funcionamento, e $P\{S2\}$ a probabilidade do sistema de *backup* estar em funcionamento. O resultado dessa disponibilidade será utilizado como parâmetro no modelo que será introduzido a seguir. Além disso, é possível a obtenção do valor do *downtime* diário através da Equação 5.3, onde $P\{F1\}$ representa a probabilidade do sistema principal estar em estado de falha e $P\{F2\}$ representa a probabilidade do sistema de *backup* estar em estado de falha, o resultado obtido fornece o número de minutos no dia em que o sistema estará indisponível.

$$A = (P\{S1\} + P\{S2\}) \quad (5.2)$$

$$Dt/d = (P\{F1\} + P\{F2\}) \times 1440min \quad (5.3)$$

A Figura 15 apresenta o modelo RBD do dispositivo móvel com um novo bloco de componente em série, o *mobile_app* representa a aplicação de *offloading* com redundância Cold Standby. Esse novo bloco recebe como parâmetro o valor da disponibilidade do modelo de

CTMC, apresentado anteriormente. Sua disponibilidade é calculada através da Equação 5.4, que é obtida através da multiplicação dos valores de disponibilidade de cada um dos blocos.

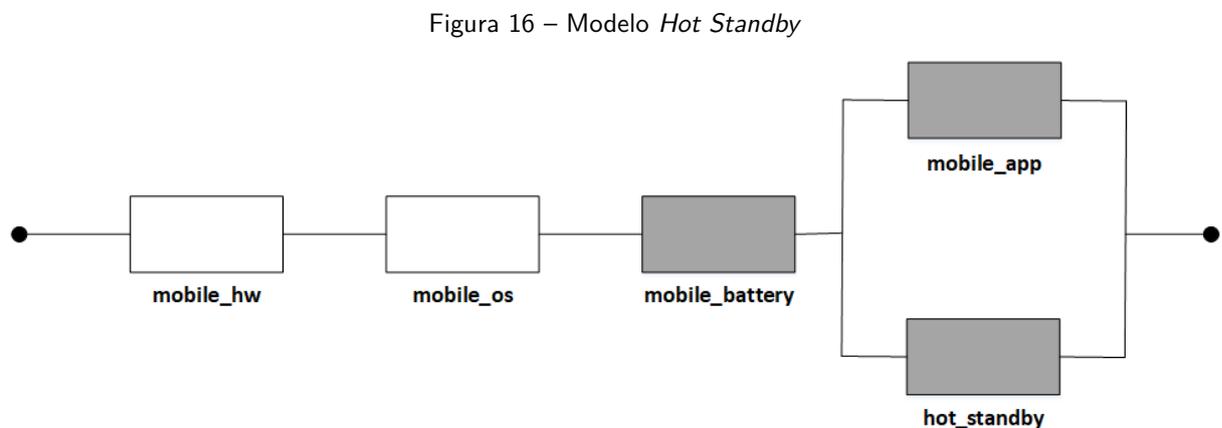


Fonte: A autora (2021)

$$A = A_{mobile_hw} \times A_{mobile_os} \times A_{mobile_battery} \times A_{mobile_app} \quad (5.4)$$

5.2.2.2 *Hot Standby*

A redundância *hot standby* também foi considerada para o modelo do dispositivo móvel, e a Figura 16 apresenta o modelo RBD do dispositivo móvel contendo essa redundância. O modelo teve a adição mais um bloco de componente, este em paralelo representando a redundância do sistema em de *Hot Standby*. Este novo bloco recebe os parâmetros de MTTF e MTTR de servidores que foi adotado da literatura (SOUSA, 2015).



Fonte: A autora (2021)

O modelo é utilizado para estimar a disponibilidade do sistema, que é calculada através da Equação 5.5.

$$A = A_{mobile_hw} \times A_{mobile_os} \times A_{mobile_battery} \times (1 - (1 - A_{mobile_app}) \times (1 - A_{hot_standby})) \quad (5.5)$$

5.2.3 Modelos de Descarregamento da Bateria

A bateria do dispositivo móvel é um componente muito importante, pois sua disponibilidade implica diretamente na disponibilidade do dispositivo. O processo de descarregamento ocorre de maneiras distintas, sendo influenciado pelo uso dos recursos disponíveis no dispositivo. Um destes recursos é a rede móvel, já que as diferentes possibilidades de conexão influenciam diretamente no tempo que a bateria levará para descarregar. Considerando que a técnica de *offloading* depende diretamente das redes móveis para ser realizada, foram desenvolvidos modelos representando o descarregamento da bateria de acordo com a rede móvel utilizada, bem como sem o uso de nenhuma rede.

As redes móveis existentes possuem diferentes larguras de banda, e a velocidade disponível influencia no tempo que leva para a transmissão de dados em cada uma delas, pois o tempo de uso e a maneira que ela utiliza os recursos do dispositivo vão gerar diferentes ritmos de consumo da energia disponível na bateria.

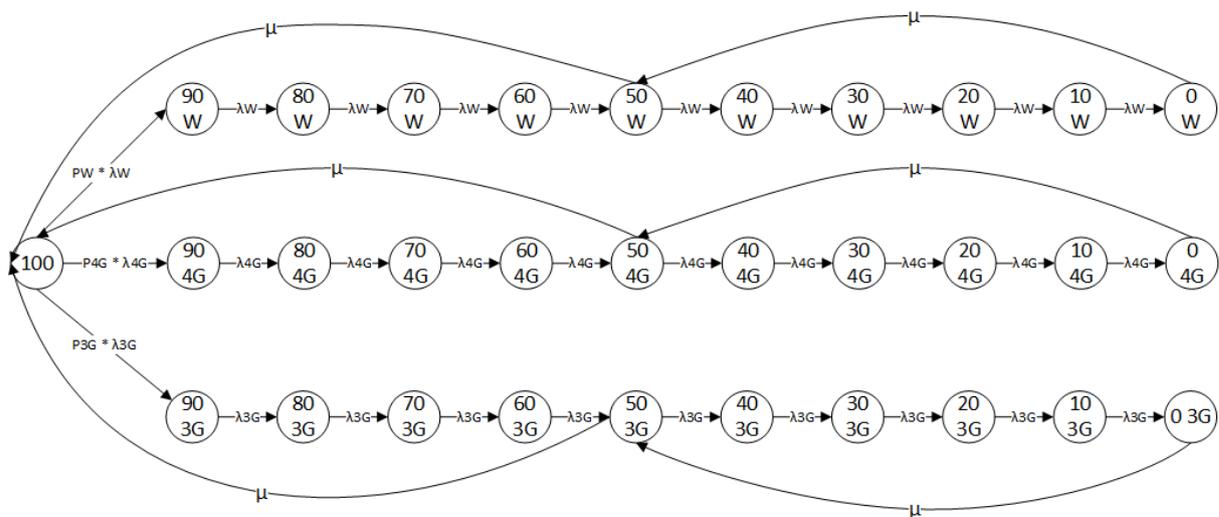
Essa seção apresenta os quatro modelos baseados em CTMC desenvolvidos que representam o descarregamento da bateria do dispositivo móvel, o primeiro deles sendo um modelo de disponibilidade considerando três interfaces de rede, o segundo o modelo de disponibilidade considerando o descarregamento sem o uso de nenhuma rede, o terceiro o modelo de confiabilidade considerando as mesmas três interfaces de rede e por fim o modelo de confiabilidade considerando o descarregamento sem uso de redes.

5.2.3.1 Modelos de Disponibilidade da Bateria

O modelo baseado em CTMC que representa o descarregamento da bateria do dispositivo móvel foi baseado no modelo de (MATOS et al., 2014) que representa o descarregamento em função da rede que está sendo utilizada. O modelo original contempla apenas as interfaces Wi-Fi e 3G e também considera o carregamento instantâneo do sistema através da troca da bateria descarregada por outra bateria reserva completamente carregada.

A Figura 17 apresenta o modelo que foi aprimorado, e é possível observar que foi adicionado um novo caminho que representa o descarregamento da bateria gerado pelo uso da interface de rede 4G, além disso, o carregamento ocorre através da conexão de um carregador e não mais pela troca total da bateria.

Figura 17 – Modelo de Disponibilidade da Bateria



Fonte: A autora (2021)

No estado 100 a bateria está totalmente carregada e vai mudando de estado a cada 10% de descarregamento. As taxas PW , $P4G$ e $P3G$ representam as probabilidades de cada uma das interfaces de rede móvel estarem disponíveis. Como o descarregamento é dado através da quantidade de energia ao utilizar cada uma das redes e por fim μ representa a taxa de carregamento em termos de 50%.

Através deste modelo é possível obter a disponibilidade da bateria, que é calculada através da Equação 5.6 que se encontra na notação do Mercury, onde $P\{0W\}$, $P\{04G\}$ e $P\{03G\}$ re-

presentam as probabilidades do modelo estar em qualquer um desses estados que representam o descarregamento total da bateria. Também é possível estimar o *downtime* diário através da Equação 5.7 que também se encontra na notação do Mercury e seu resultado estima a quantidade de minutos num dia que o sistema estará indisponível.

$$A = 1 - (P\{0W\} + P\{04G\} + P\{03G\}) \quad (5.6)$$

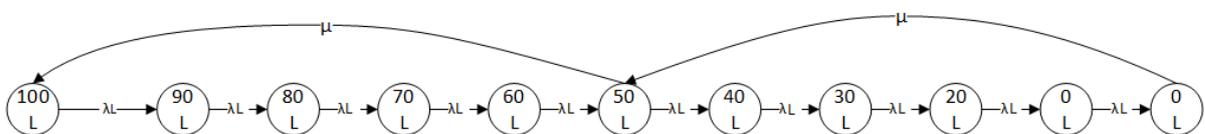
$$Dt/d = (1 - (1 - (P\{0W\} + P\{04G\} + P\{03G\}))) \times 1440min \quad (5.7)$$

Para este modelo não foi considerada a troca de redes durante a utilização do dispositivo para realização do *offloading*, já que na prática ao se desconectar de uma rede para se conectar em outra a troca de dados necessária durante o uso da técnica é interrompido e isso faz com que o processo seja reiniciado.

Uma versão simplificada do modelo de disponibilidade apresentado anteriormente foi desenvolvida para considerar o descarregamento da bateria sem uso de nenhuma rede móvel. Nesse contexto, todas as redes ficam desativadas e é possível estimar, por exemplo, o consumo de energia quando o processamento de tarefas é feito localmente no próprio dispositivo, sem o uso do *offloading*.

A Figura 18 mostra o modelo de disponibilidade da bateria sem uso de redes, onde λL é a taxa de descarregamento de 10% da bateria quando nenhuma rede está ativa e μ é a taxa de recarga da bateria a cada 50%.

Figura 18 – Modelo de Disponibilidade da Bateria sem Uso de Redes Móveis



Fonte: A autora (2021)

A disponibilidade deste modelo é calculada através da Equação 5.8, onde $P\{0L\}$ representa a probabilidade do sistema chegar ao estado 0L que representa o descarregamento total da bateria. O *downtime* diário é calculado através da Equação 5.9 que resulta na quantidade de minutos que o sistema ficará indisponível no período de um dia. Ambas as equações estão na notação do Mercury.

$$A = 1 - (P\{0L\}) \quad (5.8)$$

$$Dt/d = (1 - (1 - (P\{0L\}))) \times 1440min \quad (5.9)$$

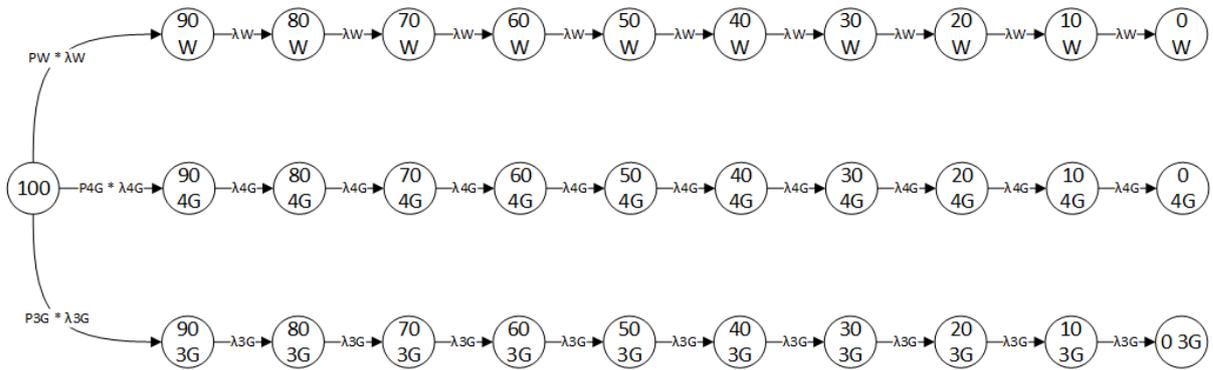
5.2.3.2 Modelos de Confiabilidade da Bateria

Para que fosse possível considerar também a confiabilidade da bateria, tomando como base o modelo anterior, o modelo que pode ser visto na Figura 19 foi desenvolvido. Neste modelo, semelhante ao de disponibilidade, o descarregamento é considerado em termos de 10%, e ao atingir um dos estados, a bateria descarrega e se torna indisponível. Neste cenário foi considerado o recarregamento da bateria, representando situações em que o dispositivo móvel não está conectado a um carregador. As taxas das transições são as mesmas dos modelos de disponibilidade, exceto pela taxa de carregamento da bateria, que não é considerada neste modelo.

Através deste modelo é possível estimar a confiabilidade da bateria, que é calculada através da Equação 5.10, onde $P\{0W\}$, $P\{04G\}$ e $P\{03G\}$ representam as probabilidades do modelo estar em algum desses estados que representam o descarregamento total da bateria. e o *downtime* diário que é calculado através da Equação 5.11.

$$R = 1 - (P\{0W\} + P\{04G\} + P\{03G\}) \quad (5.10)$$

Figura 19 – Modelo de Confiabilidade da Bateria

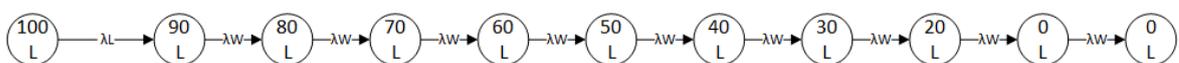


Fonte: A autora (2021)

$$Dt/d = (1 - (1 - (P\{0W\} + P\{04G\} + P\{03G\}))) \times 1440min \quad (5.11)$$

Semelhante ao que foi feito nos modelos de disponibilidade, também foi desenvolvida uma versão simplificada do modelo de confiabilidade apresentado anteriormente para considerar o descarregamento da bateria sem uso de nenhuma rede móvel, a Figura 20 apresenta este modelo que possui as mesmas taxas do modelo simplificado de disponibilidade, excetuando a taxa de carregamento que não é considerada neste modelo.

Figura 20 – Modelo de Confiabilidade da Bateria sem Uso de Redes Móveis



Fonte: A autora (2021)

Através deste modelo é possível estimar a confiabilidade da bateria quando não há uso de nenhuma rede móvel. Este cálculo é feito através da Equação 5.12, onde $P\{0L\}$ representa a probabilidade do modelo estar no estado 0L que representa o descarregamento total da bateria. É possível calcular o *downtime* diário que através da Equação 5.13, resultando no número de minutos que o sistema estará indisponível num dia.

$$R = 1 - (P\{0L\}) \quad (5.12)$$

$$Dt/d = (1 - (1 - (P\{0L\}))) \times 1440min \quad (5.13)$$

5.3 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os modelos propostos RBD e CTMC que compõem esse trabalho. Primeiramente foi apresentada a metodologia utilizada para basear o desenvolvimento do trabalho e a arquitetura que baseou os modelos concebidos. Em seguida foram apresentados os modelos e feita a descrição de seus elementos, assim como de suas métricas e das equações utilizadas para obter valores de confiabilidade, disponibilidade e *downtime*.

6 RESULTADOS EXPERIMENTAIS

O Capítulo atual apresenta os resultados do experimento que foi realizado com o objetivo de obter dados do sistema real. Os dados obtidos são utilizados tanto para validar os modelos, como para alimentar os modelos concebidos e assim realizar análises em cada um deles. São apresentados também os resultados das avaliações realizadas utilizando os modelos. Inicialmente, é apresentada a configuração que foi utilizada no experimento, onde é definido o número de replicações e as atividades adotadas, os dispositivos e redes utilizados. Em seguida, são descritos os experimentos efetuados para obtenção de dados para validação do modelo concebido e para avaliação da técnica de *offloading*. Por fim, são exibidos os resultados experimentais obtidos e as análises realizadas nos modelos desenvolvidos.

6.1 CONFIGURAÇÃO DOS EXPERIMENTOS

Esta seção apresenta a configuração adotada no ambiente de experimentação para obtenção de dados para validação e experimentação dos modelos concebidos com o propósito de prover a avaliação de disponibilidade do dispositivo móvel.

Os experimentos são baseados na realização da técnica de *offloading* no sistema real, utilizando um dispositivo móvel que utiliza um aplicativo de aplicação de filtros em imagens e que possui a opção de realizar a tarefa localmente, ou de migrar os dados através de um *framework* de *offloading* para que as tarefas sejam realizadas numa nuvem.

Para a realização do experimento é utilizado o *framework* de *offloading* MpOs, que tem um módulo executado juntamente com o aplicativo de edição de imagem BenchImage e outro módulo executado na nuvem, permitindo que as tarefas de aplicação de filtro realizadas pelo aplicativo sejam processadas no próprio dispositivo móvel ou sejam enviadas para serem processadas na nuvem. A arquitetura desse sistema que envolve o *framework* e a conexão do aplicativo de edição e a nuvem é detalhada na Seção 4.2.

O dispositivo móvel utilizado tem suas especificações técnicas detalhadas na Tabela 3.

A Tabela 4, reúne informações sobre cada fator do experimento e seus respectivos níveis.

Dentre as imagens disponíveis no BenchImage, foi selecionada a primeira imagem, FAB Show, para ser manipulada. Considerando os tamanhos de imagem disponíveis, foi escolhido

Tabela 3 – Configuração do Dispositivo Móvel

Componente	Configuração
Processador	4x 1.5 GHz Cortex-A53 + 4x 1.0 GHz
Memória RAM	3GB
Capacidade da Bateria	2880 mAh
Sistema Operacional	Android (7.1.2 Nougat)

Fonte: A autora (2021)

Tabela 4 – Fatores e Níveis do Experimento

Fator	Níveis
Imagem	FAB Show
Tamanho da Imagem	8MP
Filtro	Cartoonizer
Local de Processamento	Local, Nuvem
Rede Móvel Utilizada	Wi-Fi, 4G, 3G

Fonte: A autora (2021)

o de 8MP por ser o maior tamanho à disposição. Dentre as opções de filtro disponíveis, foi escolhido filtro Cartoonizer, por ser o filtro que requer um processamento mais intenso entre os disponíveis. Seu processamento envolve a aplicação de quatro outros filtros, seu processo de execução ocorre na seguinte sequência:

- (1) transforma a foto original em tons de cinza;
- (2) clona a foto e inverte suas cores;
- (3) aplica o filtro Sharpen para desfocar a imagem gerada em (1);
- (4) combina a foto gerada em (2) e (3).

Como resultado, uma nova foto que se parece com um esboço a lápis é criada e mostrada ao usuário na tela do aplicativo (COSTA et al., 2015). A Figura 21 mostra o resultado da aplicação do Cartoonizer em umas nas imagens disponíveis na aplicação.

Figura 21 – Imagem com Filtro Cartoonizer Aplicado



Fonte: Adaptado de (COSTA et al., 2015)

Como o objetivo do experimento envolve observar o consumo de energia gerado pelo uso de diferentes redes móveis, as redes Wi-Fi, 4G e 3G foram selecionadas, por serem as redes móveis mais utilizadas atualmente.

O monitoramento da bateria e a medição do consumo energético durante a realização de cada tarefa foram realizados através do aplicativo *AccuBattery* (DIGIBITES, 2020). Ele funciona monitorando o consumo energético real de cada aplicativo utilizado pelo celular, permitindo saber dados do descarregamento gerado por cada aplicação específica. Porém, o aplicativo em questão informa os valores de descarga em *Miliampère* (mAh), mas para este trabalho, será considerada a energia, e para tal, todos os resultados obtidos serão convertidos através da Equação 6.1.

$$E = V \times I \times T \quad (6.1)$$

onde:

- E representa a medida de energia
- V representa a tensão
- I se refere a corrente elétrica
- T representa o tempo

6.1.1 Cenários de Replicação do Experimento

Utilizando os fatores e níveis do experimento, foram definidos 4 tratamentos para que as replicações do experimento fossem realizadas no dispositivo e os dados obtidos para posterior utilização e análise. Esta seção apresenta as configurações de cada um desses tratamentos.

A Tabela 5 apresenta os tratamentos e os valores de cada um dos seus parâmetros. Cada um dos tratamentos foi configurado no dispositivo móvel e o filtro é aplicado 30 vezes para cada uma das 30 repetições do experimento de medição.

Tabela 5 – Tratamentos do Experimento

Parâmetros	Tratamento 1	Tratamento 2	Tratamento 3	Tratamento 4
Imagem	FAB Show	FAB Show	FAB Show	FAB Show
Tamanho da Imagem	8MP	8MP	8MP	8MP
Filtro	Cartoonizer	Cartoonizer	Cartoonizer	Cartoonizer
Aplicação do Filtro	30 vezes	30 vezes	30 vezes	30 vezes
Local de Processamento	Local	Nuvem	Nuvem	Nuvem
Rede Utilizada	Nenhuma	Wi-Fi	4G	3G

Fonte: A autora (2021)

6.2 VALIDAÇÃO

Esta seção apresenta a validação do modelo de confiabilidade da bateria do dispositivo móvel, que foi realizada utilizando os resultados obtidos do experimento de medição. A validação do modelo CTMC é apresentada juntamente com os resultados do experimento de medição.

6.2.1 Validação do Modelo de Confiabilidade

Para validar o modelo CTMC de confiabilidade da bateria do dispositivo móvel, que pode ser observado na Figura 19, utilizamos o Teste de Hipótese, com o qual estamos tomando

Tabela 6 – Resultados das Replicações

	Tratamento 1	Tratamento 2	Tratamento 3	Tratamento 4
Média de Energia Consumida	7420,62 J	320,52 J	942,82 J	1282,91 J
Média de Tempo Decorrido	3660,59 s	158,32 s	326,21 s	541,29 s
Tempo para Consumir 10% da Bateria	32,39 min	32,43 min	22,71 min	27,70 min
Taxa para Análise	$0,03min^{-1}$	$0,03min^{-1}$	$0,044min^{-1}$	$0,036 min^{-1}$

Fonte: A autora (2021)

como hipótese nula que o modelo se comporta igual ao sistema real.

Para obter os valores das taxas a serem inseridas nas transições do modelo, foram coletados dados do sistema real, onde foi realizado *offloading* através de três diferentes redes móveis. No experimento são medidos: o tempo levado pra aplicação de filtro numa imagem por 30 vezes e a taxa de descarga da bateria durante este tempo. Os valores colhidos são utilizados para estimar o tempo de descarga da bateria. Em seguida, os valores obtidos em cada um dos tratamentos são convertidos em taxas, que são obtidas ao se calcular o inverso do valor do tempo para consumo de 10% da bateria, esse valor é utilizado pois os modelos representam o descarregamento em termos de 10%. Assim, os parâmetros λ_W , λ_{4G} e λ_{3G} são referentes as atividades nas redes Wi-Fi, 4G e 3G, respectivamente.

As replicações do experimento em cada um dos quatro tratamentos configurados fornecem os resultados que se encontram no Apêndice A. As médias das replicações de cada tratamento, o tempo médio para consumo de 10% da bateria e a taxa utilizada para se inserida nos modelos, se encontram na Tabela 6.

Utilizando os valores obtidos nas replicações do experimento, foram calculadas as estimativas de tempo para descarregamento total da bateria, esses valores se encontram no Apêndice B. Em seguida, foi calculado o intervalo de confiança desses dados, para isso primeiramente foram calculados: média, mediana e desvio padrão. Em seguida foi definido um nível de confiança de 95%, sendo então possível obter o valor da variável aleatória e assim calcular os

limites inferior e superior que compõem o intervalo de confiança.

As taxas apresentadas na Tabela 6 foram inseridas no modelo de confiabilidade da bateria e foi realizada a análise transiente do modelo através do Mercury. A análise foi realizada considerando a utilização de uma rede de cada vez, para isso os valores da probabilidade de cada rede estar ativa recebiam valor de 1 para a rede de interesse enquanto as demais recebem valor de probabilidade igual a 0. Na análise calculado o MTTA para cada uma das situações para saber o tempo que levaria para o descarregamento total da bateria ao se utilizar cada uma das redes.

A Tabela 7 apresenta os dados referentes a validação do modelo, que considera o tempo para descarregamento total da bateria, onde na segunda coluna se encontram os intervalos de confiança para cada uma das interfaces de rede e na terceira coluna os valores de MTTA obtidos com as análises do modelo no Mercury.

Tabela 7 – Resultados da Validação

Tempo para descarregamento total (min.)		
Tratamento	Intervalo de Confiança 95%	CTMC
2	(315,68; 339,76)	324,34
3	(221,99; 237,91)	227,19
4	(271,68; 290,75)	277,05

Fonte: A autora (2021)

Os resultados obtidos no modelo se encontram todos dentro do intervalo de confiança previamente calculado, desta forma não é possível rejeitar a hipótese nula, pois não há evidências estatísticas que comprovem o comportamento distinto do modelo com o ambiente real.

6.3 ANÁLISE DE DEPENDABILIDADE DO DISPOSITIVO MÓVEL

Nesta seção serão apresentadas as análises do sistema que foram realizadas nos modelos desenvolvidos, assim como seus respectivos resultados. Com os dados obtidos no sistema é possível realizar uma análise da disponibilidade e da confiabilidade do dispositivo móvel.

6.3.1 Análise de Disponibilidade e Confiabilidade da Bateria do Dispositivo Móvel

Para a realização das análises utilizando os modelos CTMC da bateria do dispositivo móvel foram utilizados os valores dos parâmetros que se encontram na Tabela 8. Os parâmetros PW, P4G e P3G representam as probabilidades de cada uma das redes estarem ativadas e foram mantidas iguais, considerando um cenário em que o usuário tem todas as redes disponíveis para utilização imediata. Os parâmetros λL , λW , $\lambda 4g$ e $\lambda 3G$ representam a taxa de descarregamento de 10% da bateria. Ao ser utilizada cada uma das redes, esses valores foram obtidos através da média dos valores dos experimentos realizados. Por fim, μ representa a taxa de carregamento da bateria, que foi obtida em (MOTOROLA, 2020) que representa uma média da capacidade de recarga dos carregadores atuais.

Tabela 8 – Parâmetros Utilizados para as CTMC

Parâmetro	Valor
PW	0,33
P4G	0,33
P3G	0,33
λL	$0,03min^{-1}$
λW	$0,03min^{-1}$
$\lambda 4G$	$0,044min^{-1}$
$\lambda 3G$	$0,036min^{-1}$
μ	$0,04min^{-1}$

Fonte: A autora (2021)

Os resultados das análises estacionárias realizadas podem ser observados na Tabela 9. É possível perceber que ao utilizar o *offloading* a rede Wi-Fi oferece os melhores resultados, enquanto a rede 4G os piores, isso ocorre pois ao realizar mesmo trabalho em cada uma das redes, a 4G leva um tempo maior para completar todo o trabalho, além de gerar um consumo energético maior. Também é possível afirmar que ao utilizar 4G houve uma diminuição na disponibilidade quando comparado com o cenário em que o *offloading* não é utilizado, o resultado de disponibilidade da rede 3G quase não apresentou diferença e apenas a rede Wi-Fi que aumentou a disponibilidade.

A Tabela 10 exhibe os resultados das análises transientes realizadas, os tempos de 120 min e

Tabela 9 – Resultado de Disponibilidade da Bateria do Dispositivo Móvel

	Disponibilidade	Dt.dia (min.)
Offloading (todas as redes disponíveis)	0,9169	119,6
Offloading através do Wi-Fi	0,9371	90,5
Offloading através da 4G	0,8966	148,8
Offloading através da 3G	0,9211	113,5
Sem Offloading	0,9283	103,1

Fonte: A autora (2021)

300 min foram escolhidos pois a bateria não consegue ter duração suficiente para valores muito maiores. Os resultados obtidos mostram que o uso do *offloading* gerou um ganho quando se trata do tempo de execução, fazendo com que a execução das tarefas ocorresse num tempo muito menor, porém, quando o aspecto analisado é o consumo energético, o uso da técnica não trouxe ganhos.

Tabela 10 – Resultado de Confiabilidade da Bateria do Dispositivo Móvel

	Tempo (min.)	Confiabilidade	Dt/dia (min.)
Com Offloading	120	0,9866	19,2
	300	0,3592	922,7
Sem Offloading	120	0,9952	6,8
	300	0,5530	643,6

Fonte: A autora (2021)

6.3.2 Análise de Disponibilidade do Dispositivo Móvel

A análise estacionária efetuada utilizando o modelo RBD de Disponibilidade do Dispositivo Móvel foi feita considerando a realização do *offloading* e utilizando primeiramente a abordagem sem redundância e em seguida considerando com redundância, todas as análises foram feitas considerando a disponibilidade do sistema em 1440 minutos (um dia), os valores de *uptime*, representam o tempo esperado de funcionamento do sistema e os valores de *downtime* o tempo de falha esperado do sistema.

A Tabela 11 exibe os parâmetros que foram utilizados em cada um dos componentes do modelo. Os valores de MTTF e MTTR dos componentes *mobile_hw* e *mobile_os* foram her-

dados do trabalho de (OLIVEIRA et al., 2013), o bloco *mobile_battery* recebeu como entrada o valor da disponibilidade obtido através da análise estacionária do Modelo de Descarregamento da Bateria, os blocos *mobile_app* e *hot_standby* tiveram os valores de MTTF, MTTR e *Mean Time do Absorption* (MTTA) herdados da literatura (SOUSA, 2015)

Tabela 11 – Parâmetros da Análise de Disponibilidade do Dispositivo Móvel

Componente	MTTF	MTTR	MTTA	Disponibilidade
mobile_hw	13483146 min	720 min	x	x
mobile_os	86455,32 min	1,9998 min	x	x
mobile_battery	x	x	x	0,9169
mobile_app	86040 min	480 min	4,8 min	x
hot_standby	86040 min	480 min	4,8 min	x

Fonte: A autora (2021)

A Tabela 12 exibe os resultados obtidos com a análise realizada através da bordagem sem redundância, com os valores de disponibilidade, *uptime* e *downtime*.

Tabela 12 – Disponibilidade do Dispositivo Móvel sem Redundância

Disponibilidade	Uptime (min.)	Downtime (min.)
0,9168	482205,52	43743,25

Fonte: A autora (2021)

Realizando a análise do modelo de *cold standby* do servidor, foi obtido o valor de disponibilidade de 0,9999, este valor foi inserido no bloco *mobile_app* do RDB. Em seguida foi realizada a análise do modelo RBD do dispositivo móvel, levando em consideração a adição do mecanismo de redundância em *cold standby*, os resultados obtidos são exibidos na Tabela 14.

Tabela 13 – Disponibilidade do Dispositivo Móvel com Redundância Cold Standby

Disponibilidade	Uptime (min.)	Downtime (min.)
0,96406	507047,3	18901,4

Fonte: A autora (2021)

Por fim, foi realizada a análises do modelo RBD adicionando o componente de redundância em *Hot Standby*. Os resultados podem ser observados na Tabela 14.

Tabela 14 – Disponibilidade do Dispositivo Móvel com Redundância Hot Standby

Disponibilidade	Uptime (min.)	Downtime (min.)
0,96414	507090,7	18858,1

Fonte: A autora (2021)

É possível observar o aumento da disponibilidade do sistema ao se adicionar cada um dos mecanismos de redundância.

6.4 CONCLUSÕES

A partir dos modelos desenvolvidos e dos parâmetros obtidos na literatura e através do experimento, foi possível realizar diversas análises que envolvem o uso da técnica de *offloading* em dispositivos móveis. Ao fornecer um olhar sobre a importância de analisar diversos cenários, tivemos o objetivo de obter dados e informações que permitissem melhorar os mecanismos de decisão utilizados nos *frameworks* de *offloading* existentes, fornecendo informações que permitam focar no aspecto que é de interesse em cada aplicação da técnica.

Com os experimentos e análises que foram realizados, é possível perceber que a técnica de *offloading* apesar de muito promissora, tem seu uso afetado de acordo com a configuração do dispositivo móvel e rede móvel utilizados, esses fatores são extremamente importantes para definir o sucesso ou não da técnica. Além disso, é possível perceber que apesar de nem sempre gerar uma redução do consumo energético, a técnica pode aumentar a velocidade do processamento de uma tarefa. Por isso, é essencial entender qual a finalidade da aplicação da técnica, quais recursos se deseja aumentar ou otimizar e assim poder aplicar a técnica de maneira eficiente.

Um exemplo da utilização dos dados é numa situação em que se deseja obter o melhor tempo de execução das tarefas, nos cenários analisados é possível perceber que o uso do Wi-Fi oferece os melhores resultados, ao se tratar de um cenário onde há um menor consumo energético, é possível fazer o processamento localmente ou utilizar a rede Wi-Fi para realizar o *offloading* e os resultados serão semelhantes.

Este capítulo apresentou os resultados obtidos com o experimento realizado e forneceu as seguintes análises realizadas nos modelos desenvolvidos:

- Análise de Disponibilidade da Bateria do Dispositivo Móvel;

- Análise de Confiabilidade da Bateria do Dispositivo Móvel;
- Análise de Disponibilidade do Dispositivo Móvel;
- Análise de Disponibilidade do Dispositivo Móvel com Redundância em *Cold Standby*;
- Análise de Disponibilidade do Dispositivo Móvel com Redundância em *Hot Standby*;

7 CONCLUSÕES

A computação móvel vem se desenvolvendo ao longo dos anos e temos assistido ao grande crescimento do uso de dispositivos móveis em suas mais diversas formas e configurações. Para muitos usuários, eles têm sido o principal meio de acesso aos mais diversos sistemas, às aplicações e soluções que visam facilitar a vida dos milhões de usuários.

Diante de um crescimento tão notório da computação móvel, cresceu também as possibilidades do que se pode fazer utilizando um dispositivo móvel, partindo de tarefas simples até as mais complexas aplicações envolvendo processamento de imagem, inteligência artificial, manipulação de mídias, processamento de dados. Isso faz com que recursos como o processador, a memória e a bateria dos dispositivos tenham dificuldade em acompanhar todo esse avanço, e em algumas situações eles não sejam suficientes ou não forneçam um uso satisfatório ao usuário.

Ao mesmo tempo que tudo isso ocorre, podemos ver também um grande avanço da tecnologia de computação em nuvem, que permite que os usuários tenham acesso aos mais diversos recursos computacionais através da rede de maneira escalável. Ao unir esses dois cenários, temos a computação em nuvem móvel que surgiu com o objetivo de fornecer todos os recursos da computação em nuvem também para os usuários da computação móvel.

Dentro do universo de MCC existe a técnica de *offloading*, que permite que o processamento de tarefas mais pesadas sejam migradas para um servidor de nuvem, aumentando o poder de processamento dos dispositivos móveis e trazendo vantagens como a economia de energia. Porém, com o avanço dos dispositivos móveis, é necessário também que os métodos em que essa técnica é aplicada avancem também, para que seu uso se torne cada vez mais viável e vantajoso.

Os principais problemas e desafios relatados na literatura, mostram a grande importância de investir em estudos que possam melhorar a tomada de decisão para o *offloading*. Mecanismos que possuem informações suficientes para analisar o cenário do dispositivo móvel, da nuvem e das redes, podem tomar decisões cada vez mais robustas que vão tornar a técnica cada vez mais bem treinada para suprir as necessidades do usuário nos mais diversos cenários.

Levando em consideração esses pontos, foi desenvolvido este trabalho que faz uma análise do uso da técnica em diversos cenários, gerando mais informações e dados que podem auxiliar e corroborar na criação de mecanismos de tomada de decisão ainda mais robustos.

Este trabalho teve como contribuições:

- A realização do experimentos de medição no sistema real, considerando o uso da técnica de *offloading* em diferentes cenários e utilizando diferentes redes móveis;
- O desenvolvimento de um Modelo de Disponibilidade do Dispositivo Móvel;
- O desenvolvimento de um Modelo de Disponibilidade do Dispositivo Móvel com mecanismo de redundância em *Cold Standby*;
- O desenvolvimento de um Modelo de Disponibilidade do Dispositivo Móvel com mecanismo de redundância em *Hot Standby*;
- O desenvolvimento de Modelos de Descarregamento da Bateria do Dispositivo Móvel, considerando o descarregamento durante a execução da técnica de *offloading* em diferentes redes móveis;
- A realização da análise de disponibilidade da bateria do dispositivo móvel;
- A realização da análise de confiabilidade da bateria do dispositivo móvel;
- A realização da análise de disponibilidade do dispositivo móvel;
- A realização da análise de disponibilidade do dispositivo móvel com redundância em *cold dtandy*;
- A realização da análise de disponibilidade do dispositivo móvel com redundância em *hot standy*.

7.1 LIMITAÇÕES

Os modelos que representam o descarregamento da bateria não representam a troca entre os tipos de rede e a inatividade das redes, que são situações que podem ocorrer com certa

frequência durante o uso de redes móveis.

O experimento realizado, apesar de apresentar aspectos importantes, não mostra o uso de aplicações que realizam outros tipos de tarefas com grande demanda de processamento e nem consegue abordar o uso da técnica de *offloading* em dispositivos móveis com configurações diferentes.

7.2 TRABALHOS FUTUROS

Os resultados obtidos com esse trabalho abrem caminho para diversas possibilidades de análises que envolvem a técnica de *offloading* e o uso de dispositivos móveis. Algumas possibilidades de trabalhos futuros são:

- Replicar os experimentos com mais dispositivos móveis com diferentes configurações;
- Replicar os experimentos com o uso de redes móveis 5G;
- Desenvolver novos experimentos com outros tipos de aplicações que demandam um poder ainda maior de processamento;
- Estender os modelos da bateria do dispositivo móvel considerando redes 5G.

REFERÊNCIAS

- AIJAZ, A.; AGHVAMI, H.; AMANI, M. A survey on mobile data offloading: technical and business perspectives. *IEEE Wireless Communications*, IEEE, v. 20, n. 2, p. 104–112, 2013.
- AKHERFI, K.; GERNDT, M.; HARROUD, H. Mobile cloud computing for computation offloading: Issues and challenges. *Applied computing and informatics*, Elsevier, v. 14, n. 1, p. 1–16, 2016.
- AL-JANABI, S.; AL-SHOURBAJI, I.; SHOJAFAR, M.; ABDELHAG, M. Mobile cloud computing: challenges and future research directions. In: IEEE. *2017 10th International Conference on Developments in eSystems Engineering (DeSE)*. [S.l.], 2017. p. 62–67.
- ALTAMIMI, M.; ABDRABOU, A.; NAIK, K.; NAYAK, A. Energy cost models of smartphones for task offloading to the cloud. *IEEE Transactions on Emerging Topics in Computing*, IEEE, v. 3, n. 3, p. 384–398, 2015.
- AMAZON. *Amazon EC2*. 2020. Disponível em: <<https://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html>>. Acesso em: 20 setembro 2020.
- AVIZIENIS, A.; LAPRIE, J.-C.; RANDELL, B. et al. *Fundamental concepts of dependability*. [S.l.]: University of Newcastle upon Tyne, Computing Science, 2001.
- BAIER, C.; HAVERKORT, B.; HERMANN, H.; KATOEN, J.-P. Model-checking algorithms for continuous-time markov chains. *IEEE Transactions on software engineering*, IEEE, v. 29, n. 6, p. 524–541, 2003.
- BAUER, E.; ADAMS, R.; EUSTACE, D. *Beyond redundancy: how geographic redundancy can improve service availability and reliability of computer-based systems*. [S.l.]: John Wiley & Sons, 2011.
- BOLCH, G.; GREINER, S.; MEER, H. D.; TRIVEDI, K. S. *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. [S.l.]: John Wiley & Sons, 2006.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. [S.l.]: Elsevier Brasil, 2006.
- BOUROUNI, K. Availability assessment of a reverse osmosis plant: comparison between reliability block diagram and fault tree analysis methods. *Desalination*, Elsevier, v. 313, p. 66–76, 2013.
- CHEN, X. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 26, n. 4, p. 974–983, 2014.
- CHOUHAN, S. Energy optimal partial computation offloading framework for mobile devices in multi-access edge computing. In: IEEE. *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. [S.l.], 2019. p. 1–6.
- CHUN, B.-G.; IHM, S.; MANIATIS, P.; NAIK, M.; PATTI, A. Clonecloud: elastic execution between mobile device and cloud. In: *Proceedings of the sixth conference on Computer systems*. [S.l.: s.n.], 2011. p. 301–314.

- CLARK, J. A.; PRADHAN, D. K. Fault injection: A method for validating computer-system dependability. *Computer*, IEEE, v. 28, n. 6, p. 47–56, 1995.
- COSTA, P. B.; REGO, P. A.; ROCHA, L. S.; TRINTA, F. A.; SOUZA, J. N. de. Mpos: a multiplatform offloading system. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. [S.l.: s.n.], 2015. p. 577–584.
- CUERVO, E.; BALASUBRAMANIAN, A.; CHO, D.-k.; WOLMAN, A.; SAROIU, S.; CHANDRA, R.; BAHL, P. Maui: making smartphones last longer with code offload. In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. [S.l.: s.n.], 2010. p. 49–62.
- DE, D. *Mobile cloud computing: architectures, algorithms and applications*. [S.l.]: CRC Press, 2016.
- DIGIBITES. *AccuBattery*. 2020. Disponível em: <<https://accubatteryapp.com>>. Acesso em: 21 abril 2020.
- DIHAL, S.; BOUWMAN, H.; REUVER, M. de; WARNIER, M.; CARLSSON, C. Mobile cloud computing: state of the art and outlook. *info*, Emerald Group Publishing Limited, v. 15, n. 1, p. 4–16, 2013.
- DINH, H. T.; LEE, C.; NIYATO, D.; WANG, P. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, Wiley Online Library, v. 13, n. 18, p. 1587–1611, 2013.
- ENZAI, N. I. M.; TANG, M. A taxonomy of computation offloading in mobile cloud computing. In: IEEE. *2014 2nd IEEE international conference on mobile cloud computing, services, and engineering*. [S.l.], 2014. p. 19–28.
- FAN, Q.; LIU, L. A survey of challenging issues and approaches in mobile cloud computing. In: IEEE. *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. [S.l.], 2016. p. 87–90.
- FERNANDO, N.; LOKE, S. W.; RAHAYU, W. Mobile cloud computing: A survey. *Future generation computer systems*, Elsevier, v. 29, n. 1, p. 84–106, 2013.
- FERRARI, A.; GIORDANO, S.; PUCCINELLI, D. Reducing your local footprint with anyrun computing. *Computer Communications*, Elsevier, v. 81, p. 1–11, 2016.
- FORCHE, R. Analysis of reliability block diagrams with multiple blocks per component. In: IEEE. *Annual Proceedings on Reliability and Maintainability Symposium*. [S.l.], 1990. p. 145–148.
- GAO, J.; GRUHN, V.; HE, J.; ROUSSOS, G.; TSAI, W.-T. et al. Mobile cloud computing research-issues, challenges and needs. In: IEEE. *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*. [S.l.], 2013. p. 442–453.
- GOMES, F. A.; REGO, P. A.; ROCHA, L.; SOUZA, J. N. de; TRINTA, F. Chaos: A context acquisition and offloading system. In: IEEE. *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*. [S.l.], 2017. v. 1, p. 957–966.

- GORDON, M. S.; JAMSHIDI, D. A.; MAHLKE, S.; MAO, Z. M.; CHEN, X. {COMET}: Code offload by migrating execution transparently. In: *Presented as part of the 10th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 12)*. [S.l.: s.n.], 2012. p. 93–106.
- GU, X.; NAHRSTEDT, K.; MESSER, A.; GREENBERG, I.; MILOJICIC, D. Adaptive offloading for pervasive computing. *IEEE Pervasive Computing*, IEEE, v. 3, n. 3, p. 66–73, 2004.
- GUO, H.; YANG, X. A simple reliability block diagram method for safety integrity verification. *Reliability Engineering & System Safety*, Elsevier, v. 92, n. 9, p. 1267–1273, 2007.
- HLATSHWAYO, C. M.; ZUVA, T. Mobile public cloud computing, merits and open issues. In: IEEE. *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*. [S.l.], 2016. p. 128–132.
- JOARI, S.; NETO, M. G. d. S.; REGO, P. A.; GOMES, D. G. Um mecanismo de offloading de dados com tomada de decisao. In: SBC. *Anais do XVI Workshop em Desempenho de Sistemas Computacionais e de Comunicaao*. [S.l.], 2017.
- KAKADIA, D.; SARIPALLI, P.; VARMA, V. Mecca: mobile, efficient cloud computing workload adoption framework using scheduler customization and workload migration decisions. In: *Proceedings of the first international workshop on Mobile cloud computing & networking*. [S.l.: s.n.], 2013. p. 41–46.
- KEMP, R.; PALMER, N.; KIELMANN, T.; BAL, H. Cuckoo: a computation offloading framework for smartphones. In: SPRINGER. *International Conference on Mobile Computing, Applications, and Services*. [S.l.], 2010. p. 59–79.
- KHANNA, A.; KERO, A.; KUMAR, D. Mobile cloud computing architecture for computation offloading. In: IEEE. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*. [S.l.], 2016. p. 639–643.
- KOSTA, S.; AUCINAS, A.; HUI, P.; MORTIER, R.; ZHANG, X. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In: IEEE. *2012 Proceedings IEEE Infocom*. [S.l.], 2012. p. 945–953.
- KOVACHEV, D.; YU, T.; KLAMMA, R. Adaptive computation offloading from mobile devices into the cloud. In: IEEE. *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*. [S.l.], 2012. p. 784–791.
- KRISTENSEN, M. D.; BOUVIN, N. O. Scheduling and development support in the scavenger cyber foraging system. *Pervasive and Mobile Computing*, Elsevier, v. 6, n. 6, p. 677–692, 2010.
- KUMAR, K.; LIU, J.; LU, Y.-H.; BHARGAVA, B. A survey of computation offloading for mobile systems. *Mobile Networks and Applications*, Springer, v. 18, n. 1, p. 129–140, 2013.
- KUMAR, K.; LU, Y.-H. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, IEEE, n. 4, p. 51–56, 2010.
- KUO, W.; ZUO, M. J. *Optimal reliability modeling: principles and applications*. [S.l.]: John Wiley & Sons, 2003.

- LI, W.; ZHAO, Y.; LU, S.; CHEN, D. Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing. *IEEE Communications Magazine*, IEEE, v. 53, n. 3, p. 89–97, 2015.
- LIN, L.; LIAO, X.; JIN, H.; LI, P. Computation offloading toward edge computing. *Proceedings of the IEEE*, IEEE, v. 107, n. 8, p. 1584–1607, 2019.
- LIU, D.; KHOUKHI, L.; HAFID, A. Data offloading in mobile cloud computing: A markov decision process approach. In: IEEE. *2017 IEEE International Conference on Communications (ICC)*. [S.l.], 2017. p. 1–6.
- MA, R. K.; LAM, K. T.; WANG, C.-L. excloud: Transparent runtime support for scaling mobile applications in cloud. In: IEEE. *2011 International Conference on Cloud and Service Computing*. [S.l.], 2011. p. 103–110.
- MA, X.; ZHAO, Y.; ZHANG, L.; WANG, H.; PENG, L. When mobile terminals meet the cloud: computation offloading as the bridge. *IEEE Network*, IEEE, v. 27, n. 5, p. 28–33, 2013.
- MACH, P.; BECVAR, Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, IEEE, v. 19, n. 3, p. 1628–1656, 2017.
- MACIEL, P.; MATOS, R.; SILVA, B.; FIGUEIREDO, J.; OLIVEIRA, D.; FÉ, I.; MACIEL, R.; DANTAS, J. Mercury: Performance and dependability evaluation of systems with exponential, expolynomial, and general distributions. In: IEEE. *2017 IEEE 22nd Pacific Rim international symposium on dependable computing (PRDC)*. [S.l.], 2017. p. 50–57.
- MACIEL, P.; TRIVEDI, K.; MATIAS, R.; KIM, D. Performance and dependability in service computing: concepts, techniques and research directions. *SIJ: Igi Global*, 2012.
- MALHOTRA, M.; TRIVEDI, K. S. Dependability modeling using petri-nets. *IEEE Transactions on reliability*, IEEE, v. 44, n. 3, p. 428–440, 1995.
- MARCH, V.; GU, Y.; LEONARDI, E.; GOH, G.; KIRCHBERG, M.; LEE, B. S. μ cloud: towards a new paradigm of rich mobile applications. *Procedia Computer Science*, Elsevier, v. 5, p. 618–624, 2011.
- MATOS, R.; ARAUJO, J.; OLIVEIRA, D.; MACIEL, P.; TRIVEDI, K. Sensitivity analysis of a hierarchical model of mobile cloud computing. *Simulation Modelling Practice and Theory*, Elsevier, v. 50, p. 151–164, 2014.
- MIETTINEN, A. P.; NURMINEN, J. K. Energy efficiency of mobile clients in cloud computing. *HotCloud*, v. 10, n. 4-4, p. 19, 2010.
- MOLLAH, M. B.; AZAD, M. A. K.; VASILAKOS, A. Security and privacy challenges in mobile cloud computing: Survey and way ahead. *Journal of Network and Computer Applications*, Elsevier, v. 84, p. 38–54, 2017.
- MOTOROLA. *Motorola*). 2020. Disponível em: <<https://www.motorola.com.br/>>. Acesso em: 20 setembro 2020.
- MPOS. *Multiplatform Offloading System*). 2020. Disponível em: <<http://mpos.great.ufc.br/>>. Acesso em: 24 agosto 2020.

- NICOL, D. M.; SANDERS, W. H.; TRIVEDI, K. S. Model-based evaluation: from dependability to security. *IEEE Transactions on dependable and secure computing*, IEEE, v. 1, n. 1, p. 48–65, 2004.
- NOOR, T. H.; ZEDADALLY, S.; ALFAZI, A.; SHENG, Q. Z. Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications*, Elsevier, v. 115, p. 70–85, 2018.
- OLIVEIRA, D.; ARAUJO, J.; MATOS, R.; MACIEL, P. Availability and energy consumption analysis of mobile cloud environments. In: IEEE. *2013 IEEE International Conference on Systems, Man, and Cybernetics*. [S.l.], 2013. p. 4086–4091.
- OLIVEIRA, D.; MATOS, R.; DANTAS, J.; FERREIRA, J.; SILVA, B.; CALLOU, G.; MACIEL, P.; BRINKMANN, A. Advanced stochastic petri net modeling with the mercury scripting language. In: *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools*. [S.l.: s.n.], 2017. p. 192–197.
- OTHMAN, M.; MADANI, S. A.; KHAN, S. U. et al. A survey of mobile cloud computing application models. *IEEE Communications Surveys & Tutorials*, iee, v. 16, n. 1, p. 393–413, 2014.
- PINHEIRO, T.; SILVA, F. A.; FÉ, I.; OLIVEIRA, D.; MACIEL, P. Performance and resource consumption analysis of elastic systems on public clouds. In: IEEE. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. [S.l.], 2019. p. 2115–2120.
- PRASAD, M. R.; GYANI, J.; MURTI, P. Mobile cloud computing: Implications and challenges. *Journal of Information Engineering and Applications*, v. 2, n. 7, p. 7–15, 2012.
- QI, H.; GANI, A. Research on mobile cloud computing: Review, trend and perspectives. In: IEEE. *2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP)*. [S.l.], 2012. p. 195–202.
- RAHMAN, M.; GAO, J.; TSAI, W.-T. Energy saving in mobile cloud computing. In: IEEE. *2013 IEEE International Conference on Cloud Engineering (IC2E)*. [S.l.], 2013. p. 285–291.
- RAJU, D. N.; SARITHA, V. A survey on communication issues in mobile cloud computing. *Walailak Journal of Science and Technology (WJST)*, v. 15, n. 1, p. 1–17, 2018.
- RAM, M. *Reliability Engineering: Methods and Applications*. [S.l.]: CRC Press, 2019.
- RAUSAND, M.; HØYLAND, A. *System reliability theory: models, statistical methods, and applications*. [S.l.]: John Wiley & Sons, 2003. v. 396.
- REGO, P. A.; COSTA, P. B.; COUTINHO, E. F.; ROCHA, L. S.; TRINTA, F. A.; SOUZA, J. N. de. Performing computation offloading on multiple platforms. *Computer Communications*, Elsevier, v. 105, p. 1–13, 2017.
- ROCZEN, B.; ARNO, R. G.; HALE, P. S. Reliability block diagram methodology applied to gold book standard network. In: IEEE. *Conference, 2004 IEEE Industrial and Commercial Power Systems Technical*. [S.l.], 2004. p. 116–126.
- SANTOS, G. B. dos; REGO, P. A.; GOMES, F. A.; TRINTA, F. Caos d2d: Uma solução para offloading de métodos entre dispositivos móveis. In: SBC. *Anais Estendidos do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web*. [S.l.], 2017. p. 164–168.

- SANTOS, G. B. dos; REGO, P. A.; TRINTA, F. Uma proposta de solução para offloading de métodos entre dispositivos móveis. In: SBC. *Anais Estendidos do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web*. [S.l.], 2017. p. 76–81.
- SANTOS, G. B. dos; TRINTA, F. A.; REGO, P. A. Impactos do offloading de processamento no tempo de execução e consumo energético de dispositivos móveis. In: SBC. *Anais Principais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. [S.l.], 2018.
- SCHMIDT, K. *High availability and disaster recovery: concepts, design, implementation*. [S.l.]: Springer Science & Business Media, 2006. v. 22.
- SHIRAZ, M.; GANI, A.; KHOKHAR, R. H.; BUYYA, R. A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. *IEEE Communications Surveys & Tutorials*, IEEE, v. 15, n. 3, p. 1294–1313, 2012.
- SILVA, F. A.; KOSTA, S.; RODRIGUES, M.; OLIVEIRA, D.; MACIEL, T.; MEI, A.; MACIEL, P. Mobile cloud performance evaluation using stochastic models. *IEEE Transactions on Mobile Computing*, IEEE, v. 17, n. 5, p. 1134–1147, 2017.
- SILVA, F. A.; MACIEL, P.; FILHO, G.; MATOS, R. A scheduler for mobile cloud based on weighted metrics and dynamic context evaluation. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. [S.l.: s.n.], 2015. p. 569–576.
- SILVA, F. A.; MACIEL, P.; MATOS, R. Smartrank: a smart scheduling tool for mobile cloud computing. *The Journal of Supercomputing*, Springer, v. 71, n. 8, p. 2985–3008, 2015.
- SILVA, F. A.; MACIEL, P.; SANTANA, E.; MATOS, R.; DANTAS, J. Mobile cloud face recognition based on smart cloud ranking. *Computing*, Springer, v. 99, n. 3, p. 287–311, 2017.
- SOUSA, E.; LINS, F.; TAVARES, E.; MACIEL, P. Cloud infrastructure planning considering different redundancy mechanisms. *Computing*, Springer, v. 99, n. 9, p. 841–864, 2017.
- SOUSA, E. T. G. d. Modelagem de desempenho, dependabilidade e custo para o planejamento de infraestruturas de nuvens privadas. UNIVERSIDADE FEDERAL DE PERNAMBUCO, 2015.
- SUH, D.; KO, H.; PACK, S. Efficiency analysis of wifi offloading techniques. *IEEE Transactions on Vehicular Technology*, IEEE, v. 65, n. 5, p. 3813–3817, 2015.
- TRUONG-HUU, T.; THAM, C.-K.; NIYATO, D. To offload or to wait: An opportunistic offloading algorithm for parallel tasks in a mobile cloud. In: IEEE. *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*. [S.l.], 2014. p. 182–189.
- UMAIR, S.; MUNEEER, U.; ZAHOOR, M. N.; MALIK, A. W. Mobile computing: issues and challenges. In: IEEE. *2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET)*. [S.l.], 2015. p. 1–5.
- VERBELEN, T.; SIMOENS, P.; TURCK, F. D.; DHOEDT, B. Aiolos: Middleware for improving mobile application performance through cyber foraging. *Journal of Systems and Software*, Elsevier, v. 85, n. 11, p. 2629–2639, 2012.

WU, H. Performance modeling of delayed offloading in mobile wireless environments with failures. *IEEE Communications Letters*, IEEE, v. 22, n. 11, p. 2334–2337, 2018.

WU, L.; DU, X.; ZHANG, H.; YU, W.; WANG, C. Effective task scheduling in proximate mobile device based communication systems. In: IEEE. *2015 IEEE International Conference on Communications (ICC)*. [S.l.], 2015. p. 3503–3508.

XIE, M.; DAI, Y.-S.; POH, K.-L. *Computing system reliability: models and analysis*. [S.l.]: Springer Science & Business Media, 2004.

XU, C.; QIAO, Y.; LEE, B.; MURRAY, N. Energy consumption of mobile offloading for javascript applications. In: IEEE. *2015 26th Irish Signals and Systems Conference (ISSC)*. [S.l.], 2015. p. 1–6.

ZAMOJSKI, W.; CABAN, D. Introduction to the dependability modeling of computer systems. In: IEEE. *2006 International Conference on Dependability of Computer Systems*. [S.l.], 2006. p. 100–109.

ZHANG, Y.; NIYATO, D.; WANG, P. Offloading in mobile cloudlet systems with intermittent connectivity. *IEEE Transactions on Mobile Computing*, IEEE, v. 14, n. 12, p. 2516–2529, 2015.

APÊNDICE A – RESULTADO DAS REPLICAÇÕES DO EXPERIMENTO

A Tabela 15 exibe os resultados obtidos com as 30 replicações feitas nos Tratamentos 1 e 2. A Tabela 16 exibe os resultados obtidos com as 30 replicações feitas nos Tratamentos 3 e 4.

Tabela 15 – Resultados das Replicações nos Tratamentos 1 e 2

TRATAMENTO 1		TRATAMENTO 2	
TEMPO GASTO	CONSUMO	TEMPO GASTO	CONSUMO
3515,717 s	493,9 mAh	156,057 s	23,3mAh
3652,174 s	571,1 mAh	154,378 s	22,5mAh
3616,971 s	487,8 mAh	158,109 s	23,2mAh
3625,74 s	510,4 mAh	157,228 s	29,9mAh
3628,771 s	544,9 mAh	158,201 s	22,7mAh
3600,24 s	514 mAh	157,746 s	23,0mAh
3592,727 s	502 mAh	155,81 s	22,1mAh
3585,181 s	524,6 mAh	155,824 s	23,5mAh
3582,408 s	550,9 mAh	161,191 s	23,8mAh
3578,356 s	532,7 mAh	163,364 s	25,0mAh
3658,256 s	548,3 mAh	156,892 s	23,7mAh
3578,954 s	565,3 mAh	155,706 s	23,0mAh
3652,986 s	572,3 mAh	155,82 s	22,6mAh
3587,965 s	547,5 mAh	153,689 s	23,1mAh
3522,745 s	520,5 mAh	155,232 s	31,2mAh
3578,659 s	555,6 mAh	152,758 s	23,2mAh
3521,563 s	532,2 mAh	149,774 s	21,9mAh
3625,656 s	570,3 mAh	157,233 s	23,2mAh
3601,595 s	545,9 mAh	152,799 s	22,6mAh
3556,987 s	532,8 mAh	160,12 s	25,7mAh
3598,658 s	512,3 mAh	156,514 s	22,9mAh
3528,657 s	542,3 mAh	160,542 s	20,2mAh
3610,325 s	569,8 mAh	159,337 s	24,3mAh
3510,255 s	580,3 mAh	168,427 s	21,9mAh
3535,659 s	564,3 mAh	158,056 s	19,7mAh
3564,265 s	559,3 mAh	161,27 s	28,7 mAh
3536,235 s	532,2 mAh	156,112 s	21,1mAh
3548,236 s	546,5 mAh	169,39 s	22,3mAh
3625,365 s	569,5 mAh	160,311 s	21,7mAh
5896,635 s	574,3 mAh	171,462 s	27,2mAh

Fonte: A autora (2021)

Tabela 16 – Resultados das Replicações nos Tratamentos 3 e 4

TRATAMENTO 1		TRATAMENTO 2	
TEMPO GASTO	CONSUMO	TEMPO GASTO	CONSUMO
3515,717 s	493,9 mAh	156,057 s	23,3mAh
3652,174 s	571,1 mAh	154,378 s	22,5mAh
3616,971 s	487,8 mAh	158,109 s	23,2mAh
3625,74 s	510,4 mAh	157,228 s	29,9mAh
3628,771 s	544,9 mAh	158,201 s	22,7mAh
3600,24 s	514 mAh	157,746 s	23,0mAh
3592,727 s	502 mAh	155,81 s	22,1mAh
3585,181 s	524,6 mAh	155,824 s	23,5mAh
3582,408 s	550,9 mAh	161,191 s	23,8mAh
3578,356 s	532,7 mAh	163,364 s	25,0mAh
3658,256 s	548,3 mAh	156,892 s	23,7mAh
3578,954 s	565,3 mAh	155,706 s	23,0mAh
3652,986 s	572,3 mAh	155,82 s	22,6mAh
3587,965 s	547,5 mAh	153,689 s	23,1mAh
3522,745 s	520,5 mAh	155,232 s	31,2mAh
3578,659 s	555,6 mAh	152,758 s	23,2mAh
3521,563 s	532,2 mAh	149,774 s	21,9mAh
3625,656 s	570,3 mAh	157,233 s	23,2mAh
3601,595 s	545,9 mAh	152,799 s	22,6mAh
3556,987 s	532,8 mAh	160,12 s	25,7mAh
3598,658 s	512,3 mAh	156,514 s	22,9mAh
3528,657 s	542,3 mAh	160,542 s	20,2mAh
3610,325 s	569,8 mAh	159,337 s	24,3mAh
3510,255 s	580,3 mAh	168,427 s	21,9mAh
3535,659 s	564,3 mAh	158,056 s	19,7mAh
3564,265 s	559,3 mAh	161,27 s	28,7 mAh
3536,235 s	532,2 mAh	156,112 s	21,1mAh
3548,236 s	546,5 mAh	169,39 s	22,3mAh
3625,365 s	569,5 mAh	160,311 s	21,7mAh
5896,635 s	574,3 mAh	171,462 s	27,2mAh

Fonte: A autora (2021)

APÊNDICE B – ESTIMATIVAS PARA O DESCARREGAMENTO TOTAL DA BATERIA

A Tabela 15 exibe os resultados do cálculo da estimativa de descarregamento total da bateria, obtidos através dos valores das replicações do experimento. A Tabela 16 exibe os resultados obtidos com as 30 replicações feitas nos Tratamentos 3 e 4.

Tabela 17 – Tempos Estimados de Descarregamento Total da Bateria

Tratamento 1	Tratamento 2	Tratamento 3	Tratamento 4
20500,6377	19289,44893	13408,4129	17944,04571
18417,54705	19760,384	13566,03268	17577,14917
21354,81033	19627,32414	15001,81714	16933,27248
20458,721	15144,36923	11679,98604	16928,88511
19179,40995	20071,3163	11868,71889	16440,43543
20172,55097	19752,54261	13278,97278	16832,61046
20611,66088	20304,65158	14258,10091	17106,13443
19682,27465	19096,72851	14639,82703	16733,21045
18728,14493	19505,46555	15164,7532	14984,35988
19346,09589	18819,5328	15957,11378	17897,26512
19215,3516	19065,35696	16241,64324	17119,66154
18233,48226	19497,09913	14524,83014	15159,13563
18383,01534	19856,70796	13771,79234	17782,28168
18873,6789	19161,22597	15711,64007	18699,648
19491,84553	14329,10769	14312,1974	18385,05827
18550,28423	18963,06207	14714,70102	18014,41557
19056,93619	19696,30685	14292,40922	17324,90895
18309,46744	19518,57931	12962,51803	17643,38035
19000,90419	19471,73097	14054,4	17627,37572
19226,95676	17943,40856	12470,28213	16815,57818
20230,59738	19683,85677	13081,55178	17044,90804
18739,68682	22889,15644	12327,86139	16604,30542
18248,04493	18884,38519	12972,65455	16876,26912
17421,22075	22149,30411	13719,71545	9665,229621
18044,83062	23106,66396	11302,74207	18114,45773
18353,44752	16183,19164	12744,43636	16765,76028
19139,93009	21308,1782	14326,30088	17170,07735
18705,69226	21876,37668	14871,45	16024,38095
18340,15668	21276,29862	15281,23871	17698,49263
29570,44889	23665,19426	14759,69	16277,9215

Fonte: A autora (2021)