



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA**

**LUIS SÉRGIO DA SILVA JÚNIOR**

**UM MECANISMO PARA TOMADA DE DECISÃO DE OFFLOADING**  
**RELACIONANDO CUSTO ENERGÉTICO E COMPLEXIDADE ALGORÍTMICA**

**FORTALEZA**

**2016**

LUIS SÉRGIO DA SILVA JÚNIOR

UM MECANISMO PARA TOMADA DE DECISÃO DE OFFLOADING RELACIONANDO  
CUSTO ENERGÉTICO E COMPLEXIDADE ALGORÍTMICA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Orientador: Prof. Dr. Danielo Gonçalves Gomes

FORTALEZA

2016

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- S581m Silva Junior, Luis Sérgio da.  
Um mecanismo para tomada de decisão de offloading relacionando custo energético e complexidade algorítmica / Luis Sérgio da Silva Junior. – 2016.  
55 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2016.  
Orientação: Prof. Dr. Danielo Gonçalves Gomes.
1. Mobile cloud computing. 2. Offloading. 3. Simulação. 4. Green computing. 5. Complexidade algorítmica. I. Título.

CDD 621.38

---

LUIS SÉRGIO DA SILVA JÚNIOR

UM MECANISMO PARA TOMADA DE DECISÃO DE OFFLOADING RELACIONANDO  
CUSTO ENERGÉTICO E COMPLEXIDADE ALGORÍTMICA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Aprovada em: 28/07/2016.

BANCA EXAMINADORA

---

Prof. Dr. Danielo Gonçalves Gomes (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. José Neuman de Souza  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Fernando Antonio Mota Trinta  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Paulo Romero Martins Maciel  
Universidade Federal de Pernambuco (UFPE)

Dedico este trabalho a todos que estiveram comigo e me apoiaram em meus projetos em todas as etapas de minha vida. Também o dedico às pessoas que, mesmo nos momentos de dificuldades, sonham em contribuir para o avanço, em qualquer área, de nossa sociedade!

## AGRADECIMENTOS

Aos meus pais, Ozeni Faustino e Luis Sérgio, e irmã, Simone Faustino, por sempre terem me encorajado aos estudos, não desistir dos meus objetivos e apoio em meus projetos pessoais durante várias etapas da minha vida.

À minha noiva e futura esposa, minha doce Gisele, pelo carinho, sugestões e enorme compreensão durante todas as fases do mestrado. Mesmo em momentos de dificuldade e desmotivação, pude sempre contar com seu apoio para superar os problemas e também traçar planos para alcançarmos nossos objetivos. Ainda, menciono neste manuscrito que Gisele contribuiu, sempre que possível, em outras esferas da minha vida.

Ao meu orientador, Prof Danielo, pela enorme paciência, dedicação com orientação, confiança, compreensão e apoio durante a realização deste trabalho.

À Deborah, pelas incontáveis conversas, trocas de experiências, revisões e contribuições neste e em todos os artefatos produzidos durante a conclusão de todo este trabalho.

Aos professores Paulo Romero Maciel, Fernando Mota Trinta e José Neuman de Souza por terem aceito o convite em participar da minha banca e pelas excelentes sugestões para melhoria do trabalho.

Ao Philipp, pelas dicas e ajudas em implementações necessárias no *framework* MpOS.

À Carol Sampaio, pela criação de algumas figuras contidas neste trabalho e no artigo escrito durante o mestrado.

À coordenadora do GREat, Profa Rossana Andrade, pelo apoio e calorosa recepção quando ingressei nesse grupo de pesquisa.

Aos professores Danielo, Guilherme e Marques, pelas excelentes disciplinas ministradas durante meu primeiro ano de mestrado.

À Darilu e profissionais do setor administrativo do GREat, por fornecerem equipamentos e recursos necessários para apresentações de artigos e experimentos implementados neste trabalho.

Aos meus gerentes enquanto colaborador do GREat, Rute e Bruno, pelo apoio e ajuda para tornar a realização deste mestrado concomitante ao meu emprego possível.

Aos professores Danielo, Rossana Andrade e Fernando Mota Trinta, pelos vários *feedbacks* concedidos durante as apresentações nos seminários internos WTDG.

Aos meus amigos que conheci no GREat, Henrique, Wairton, Ericson, Wendell, Roney, Paulo Ivo, Eduardo, Vladimir, Cleiton, Lenderson, Wellington, Ananda, Marcelo e demais, pelo apoio e palavras de estímulo conferidas.

Aos meus amigos de graduação em Engenharia de Computação do IFCE, Rodrigo, Lailson, Italo (meu chapa), Danilo, Vanessa e demais pelo apoio durante meu mestrado.

Ao Paulo Artur pela paciência em revisar, opinar e sugerir melhorias pertinentes na elaboração dessa dissertação.

Aos meus amigos, Maurício e Samuel, que conheci no Mestrado em Engenharia de Teleinformática, pelas reuniões e apoio durante a etapa de conclusão de disciplinas vinculadas ao PPGETI.

Aos meus amigos do ensino médio do antigo CEFET-CE, Nélio, Alana, Thiago, Fernando, Marcelo, Pedro e demais pelo apoio e compreensão nas ausências constantes das reuniões do grupo.

À Nuziana, pelo apoio e por sempre acreditar que consigo transformar meus objetivos em realidade.

Aos primos de Gisele, Carlos e Marcos, e tia Raimunda, por me acolherem em Recife para apresentação do artigo válido como qualificação de mestrado no VII SBCUP.

À mãe de Gisele, dona Walda, por ter me ajudado a custear parte da viagem ao Recife para a mesma apresentação.

Aos meus amigos que conheci no Instituto Atlântico, Rodrigo, Aline, Rubens, Vitor, Pedro, Sandra, Patrícia, Albert e outros mais, pelo apoio e força nas etapas que antecederam a finalização deste manuscrito.

À todos que contribuíram direta ou indiretamente em todo o percurso realizado por mim neste curso de mestrado.

“Research is what I am doing when I do not know  
what I am doing.”

(Wernher von Braun)



## RESUMO

Computação Móvel em Nuvem (em inglês, *Mobile Cloud Computing* - MCC) é um paradigma computacional recente que utiliza serviços remotos a fim de estender recursos computacionais dos dispositivos móveis. Entretanto, se por um lado o uso da MCC permite a extensão das capacidades energéticas e computacionais de tais dispositivos, também traz desafios consideráveis acerca do local mais adequado à execução dos procedimentos. Nesta dissertação, propomos um mecanismo de tomada de decisão de *offloading* para redução do consumo energético levando-se em conta a complexidade algorítmica do procedimento a ser executado e o tipo de rede sem fio (3G, WiFi). O mecanismo proposto foi testado e avaliado via simulação e implementação real com um *framework* MCC. Para tal, utilizamos três algoritmos clássicos de busca e ordenação (Busca binária  $O(\log n)$ , Busca sequencial  $O(n)$ , Bubble Sort  $O(n^2)$ ) e um algoritmo de produto de matrizes quadradas  $O(n^3)$ . Os resultados indicam que para procedimentos cuja complexidade é inferior a  $O(n^2)$ , independente do tamanho de  $n$  e do tipo de rede sem fio, a execução “in loco” consome menos energia. Já para os procedimentos com complexidade maior ou igual a  $O(n^2)$ , dependendo do tamanho de  $n$ , a execução remota é mais vantajosa, do ponto de vista energético. Também observamos uma redução de consumo energético de até 8% em comparação com a literatura recente.

**Palavras-chave:** Mobile Cloud Computing, Offloading, Simulação, Energia, Green computing, Complexidade Algorítmica.

## ABSTRACT

Mobile Cloud Computing is a novel paradigm that uses remote infrastructure to improve resources of mobile devices. However, if MCC turns possible extend energy and computational capabilities of such devices, it also brings new challenges about where a task should be executed. Here we propose a *offloading* decision mechanism to reduce energy and decrease execution time considering algorithm complexity from a given procedure and the type of wireless connection such as 3G or WiFi. The proposal was validated and test by simulation and real implementation with a framework MCC. We implemented four existing algorithm (Binary Search  $O(\log n)$ , Linear Search  $O(n)$ , Bubble Sort  $O(n^2)$  and Matrix Multiplication  $O(n^3)$ ). The results show that procedures with complexity below  $O(n^2)$  with any I/O byte size and any network type do not consume energy enough to be executed remotely. Otherwise it is necessary to send information to execute a procedure remotely. We also observed that our proposed mechanism provides an economy of battery energy until 8% comparing with other *offloading* decision technique.

**Keywords:** Mobile Cloud Computing. Offloading. Energy. Simulation. Algorithm complexity

## LISTA DE ILUSTRAÇÕES

Figura 1 – Taxonomia de melhorias em MCC sob a perspectiva dos dispositivos móveis. <i>Offloading</i> destacado no fluxo em vermelho. Adaptada de : (ABOLFAZLI <i>et al.</i> , 2012) . . . . .	19
Figura 2 – Visão geral MCC. Adaptada de (QI; GANI, 2012). . . . .	19
Figura 3 – Diagrama ilustrativo de tomada de decisão de <i>offloading</i> proposta. . . . .	25
Figura 4 – Consumo energético local - <i>Busca binária, Busca sequencial e Bubble Sort</i> . . . . .	27
Figura 5 – Consumo energético local - <i>Produto de matrizes</i> . . . . .	27
Figura 6 – Diagrama de classes dos modelos criados. . . . .	32
Figura 7 – Associação da modelagem proposta entre as respectivas classes e (a) Dispositivos móveis, (b) Tomada de decisão de <i>offloading</i> , (c) Rede sem fio, (d) Infraestrutura de nuvem. . . . .	33
Figura 8 – Diagrama de sequência para ilustração de chamada de modelo para tomada de decisão de <i>offloading</i> . . . . .	35
Figura 9 – Diagrama ilustrativo dos passos para execução do Experimento #4. . . . .	42
Figura 10 – Experimento #1: Tempos totais de execução (ambiente real vs. simulado), rede 4G, com diferentes tamanhos de imagem e <i>smartphone</i> (Instância Medium). . . . .	43
Figura 11 – Experimento #1: Tempos totais de execução (ambiente real vs. simulado), banda 4G, com diferentes tamanhos de imagem e <i>smartphone</i> (Instância Micro). . . . .	44
Figura 12 – Experimento #2: Decisão de <i>offloading</i> levando-se em conta os algoritmos <i>Busca binária e Bubble Sort</i> em rede 3G. . . . .	46
Figura 13 – Experimento #2: Decisão de <i>offloading</i> levando-se em conta os algoritmos <i>Busca binária e Bubble Sort</i> em rede 4G. . . . .	47
Figura 14 – Experimento #2: Decisão de <i>offloading</i> levando-se em conta o algoritmo <i>Produto de matrizes</i> em rede 3G. . . . .	48
Figura 15 – Experimento #2: Decisão de <i>offloading</i> levando-se em conta o algoritmo <i>Produto de matrizes</i> em rede 4G. . . . .	49
Figura 16 – Energia consumida em 30 minutos de execução - Ambiente Local . . . . .	50
Figura 17 – Energia consumida em 30 minutos de execução - Rede WiFi. . . . .	50
Figura 18 – Energia consumida em 30 minutos de execução - Rede 3G. . . . .	51
Figura 19 – Frequência - Tomada de decisão de <i>offloading</i> em ambiente WiFi. . . . .	52
Figura 20 – Frequência - Tomada de decisão de <i>offloading</i> em ambiente 3G. . . . .	52

## LISTA DE TABELAS

Tabela 1 – Algumas propostas de <i>offloading</i> . . . . .	15
Tabela 2 – Síntese dos trabalhos correlatos. . . . .	23
Tabela 3 – Critérios para o projeto dos experimentos . . . . .	38
Tabela 4 – Parâmetros de carga . . . . .	39
Tabela 5 – Critérios para o projeto dos experimentos . . . . .	41
Tabela 6 – Parâmetros de carga . . . . .	41
Tabela 7 – Tomada de decisão de offloading: Produto de matrizes . . . . .	46
Tabela 8 – Tomada de decisão de offloading: Bubblesort . . . . .	46

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Contextualização e Justificativa</b>	<b>14</b>
<b>1.2</b>	<b>Objetivo</b>	<b>16</b>
<b>1.3</b>	<b>Contribuição e metodologia</b>	<b>16</b>
<b>1.4</b>	<b>Estrutura da dissertação</b>	<b>17</b>
<b>2</b>	<b>CONCEITOS E REVISÃO DA LITERATURA</b>	<b>18</b>
<b>2.1</b>	<b>Computação Móvel em Nuvem e <i>Offloading</i></b>	<b>18</b>
<b>2.2</b>	<b>Trabalhos correlatos</b>	<b>18</b>
<b>2.2.1</b>	<i>Sobre arquiteturas de MCC</i>	<b>19</b>
<b>2.2.2</b>	<i>Sobre frameworks MCC</i>	<b>20</b>
<b>2.2.3</b>	<i>Sobre simuladores MCC</i>	<b>21</b>
<b>2.2.4</b>	<i>Sobre modelagem analítica e de offloading</i>	<b>21</b>
<b>2.3</b>	<b>Resumo do capítulo</b>	<b>23</b>
<b>3</b>	<b>PROPOSTA</b>	<b>24</b>
<b>3.1</b>	<b>Mecanismo proposto</b>	<b>24</b>
<b>3.2</b>	<b>Simulação de ambiente MCC</b>	<b>31</b>
<b>3.3</b>	<b>Resumo do capítulo</b>	<b>36</b>
<b>4</b>	<b>PLANEJAMENTO DOS EXPERIMENTOS</b>	<b>37</b>
<b>4.1</b>	<b>Experimentos para validação de modelos em MCC</b>	<b>37</b>
<b>4.1.1</b>	<i>Experimento #1: Tempo total de execução em offloading estático</i>	<b>37</b>
<b>4.1.2</b>	<i>Experimento #2: Tomada de decisão em offloading dinâmico</i>	<b>38</b>
<b>4.2</b>	<b>Experimentos com o mecanismo proposto</b>	<b>38</b>
<b>4.2.1</b>	<i>Experimento #3: Validação do modelo proposto</i>	<b>40</b>
<b>4.2.2</b>	<i>Experimento #4: Comparação entre modelos de offloading</i>	<b>40</b>
<b>4.3</b>	<b>Resumo do capítulo</b>	<b>42</b>
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>43</b>
<b>5.1</b>	<b>Resultados dos experimentos #1 e #2</b>	<b>43</b>
<b>5.2</b>	<b>Resultados dos experimentos #3 e #4</b>	<b>45</b>
<b>5.3</b>	<b>Resumo do capítulo</b>	<b>53</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>54</b>

<b>6.1</b>	<b>Estudos em Perspectiva</b> . . . . .	<b>55</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>56</b>

## 1 INTRODUÇÃO

Este capítulo está organizado em quatro seções. A seção 1.1 contextualiza o tema abordado nesta dissertação, destacando a problemática em foco e alguns artigos da literatura correlata; a partir daí, elaboramos três questões de partida e estabelecemos nosso objetivo central na seção 1.2; na sequência, a contribuição e o delineamento estratégico estão brevemente descritos na seção 1.3; por fim, a seção 1.4 informa a organização dos capítulos restantes.

### 1.1 Contextualização e Justificativa

Os avanços da microeletrônica, dos sistemas ciberfísicos, da computação embarcada e das redes sem fio propiciaram um desenvolvimento significativo da computação ubíqua e pervasiva. Os *smartphones*, por exemplo, têm se tornado bastante populares por suportarem um grande número de aplicativos, tais como jogos, *e-commerce* e redes sociais *on-line*. Embora estes aplicativos estejam cada vez mais exigentes em termos de processamento, armazenamento e banda, as melhorias no *hardware* dos dispositivos móveis e na vida útil das suas baterias não têm acompanhado o ritmo crescente da demanda por recursos computacionais (KHAN *et al.*, 2014) (JUSTINO; BUYYA, 2014).

Uma alternativa para atender a demanda crescente por maiores capacidades computacionais dos dispositivos móveis é simplesmente a melhoria do seu *hardware*, mas os custos financeiros desta alternativa normalmente são elevados (ABOLFAZLI *et al.*, 2012). Por outro lado, soluções que façam uso de uma infraestrutura computacional existente podem ser factíveis e economicamente vantajosas. Podemos vislumbrar, por exemplo, soluções de processamento e de armazenamento em dispositivos remotos tendo as nuvens computacionais em perspectiva.

Neste contexto, disponibilizar serviços de nuvens computacionais para expansão das capacidades físico-computacionais dos dispositivos móveis é um dos benefícios providos pela chamada Computação Móvel em Nuvem (*Mobile Cloud Computing - MCC*) (SCHÜRING, 2011), (DINH *et al.*, 2013) (GOMES *et al.*, 2015). A transferência de processamento de um dispositivo móvel para uma nuvem remota ou para uma infraestrutura computacional local é conhecida por *offloading* ou descarregamento de dados (MOTA *et al.*, 2016) (KUMAR *et al.*, 2013) (CUERVO *et al.*, 2010) (ABOLFAZLI *et al.*, 2013).

Na literatura recente, podemos encontrar soluções de *offloading* relacionando vazão (*throughput*), quantidade de processamento realizado e o consumo energético nos dispositivos

móveis (WU *et al.*, 2013), (KUMAR *et al.*, 2013), (SILVA *et al.*, 2016). Outros trabalhos têm avaliado a viabilidade do *offloading* levando em conta o consumo energético de procedimentos (CORRAL *et al.*, 2014), (SHIRAZ *et al.*, 2015a), (SEGATA *et al.*, 2014). Ainda, tratando-se de MCC, alguns autores propuseram a inclusão de outros elementos, como as *cloudlets*, para compôr cenários MCC e auxiliar os dispositivo móveis na execução de tarefas custosas (SATYANARAYANAN *et al.*, 2009).

A Tabela 1 resume alguns destes trabalhos. Os trabalhos em geral consideram apenas a nuvem como fonte de execução remota, sendo que alguns desconsideram o consumo energético como uma grandeza-chave para a tomada de decisão. Ainda, vemos que a classificação da tomada de decisão não ocorre em nenhum trabalho de acordo com a natureza ou grau de complexidade da tarefa a ser analisada. Entendemos que estratégias de *offloading* devem interferir minimamente nos recursos disponíveis no dispositivo móvel e que quanto maior o número de variáveis relacionadas, mais complexa pode tornar-se o procedimento de decisão.

Tabela 1 – Algumas propostas de *offloading*.

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
(KUMAR <i>et al.</i> , 2013)	✓			✓	✓	
(MIETTINEN; NURMINEN, 2010)	✓		✓	✓	✓	
(FLORES; SRIRAMA, 2013)	✓			✓	✓	
(WU <i>et al.</i> , 2013)	✓		✓	✓	✓	
(ZHANG <i>et al.</i> , 2013)	✓		✓	✓	✓	
(SEGATA <i>et al.</i> , 2014)	✓		✓	✓	✓	
(SHIRAZ <i>et al.</i> , 2015b)	✓		✓	✓	✓	✓
(SONG <i>et al.</i> , 2014)	✓		✓	✓	✓	

Legenda: A) Nuvem; B) *Cloudlet*; C) Energia; D) Dinâmico; E) Considera tamanho de entrada; F) Diferenciação de tarefas.

Apesar da vasta literatura sobre computação móvel, observamos uma carência de modelos e simuladores para estudo e análise dos problemas relacionados à escassez de hardware tendo as nuvens computacionais como solução em perspectiva (AHMED A. E SABYASACHI, 2014), (DINH *et al.*, 2013), (QI; GANI, 2012), (SATYANARAYANAN *et al.*, 2009), (LI *et al.*, 2013), (CUERVO *et al.*, 2010), (CHUN *et al.*, 2011), (KUMAR *et al.*, 2013), (AMORETTI *et al.*, 2015), (SILVA *et al.*, 2016).



## 1.2 Objetivo

Nesta dissertação de mestrado buscamos responder as seguintes questões: (i) Sendo necessário o *offloading*, qual o local mais indicado para sua concretização? (ii) A energia empregada durante execução ou transmissão da tarefa pode ser relacionada como seu custo computacional? (iii) O tipo de conectividade aliado ao grau de complexidade da tarefa é relevante para uma tomada de decisão acurada no sentido de consumo energético?

Desta forma, o objetivo central deste trabalho é propor um mecanismo de *offloading* considerando a relação entre o consumo energético de um dado procedimento local, tamanho das variáveis de entrada e de saída, sua ordem de complexidade algorítmica e a interface de rede sem fio do dispositivo móvel. A ideia do mecanismo é decidir, sob o ponto de vista de economia energética, qual sistema computacional dentro de um ambiente MCC é mais apropriado para executar o procedimento passível de *offloading*.

## 1.3 Contribuição e metodologia

A contribuição central deste trabalho é um modelo para tomada de decisão de *offloading* que leva em conta a ordem de complexidade algorítmica dos procedimentos em execução em dispositivos móveis e seu consumo energético. É possível recomendar o *offloading* através de uma análise estática de um código-fonte potencialmente custoso. Para tal, estabelecemos os seguintes passos metodológicos:

- Proposição de um modelo capaz de representar as principais entidades presentes em um cenário típico de MCC (vide Seção 3.2);
- Validação da modelagem proposta (vide Seção 4.1);
- Verificação de associação entre as variáveis de energia e de complexidade algorítmica (ainda, vide Seção 5.1);
- Novo mecanismo de decisão de *offloading* que leve em consideração o consumo energético e a complexidade algorítmica do procedimento (vide Seção 3.1);
- Verificação e validação do mecanismo proposto (vide Seções 4.2);
- Comparação dos resultados alcançados pelo mecanismo proposto frente a uma modelagem de decisão de *offloading* da literatura (vide Seção 5.2).

## 1.4 Estrutura da dissertação

Esse documento foi organizado da seguinte forma: no Capítulo 2 realizamos um levantamento de conceitos e trabalhos correlatos ao tema de computação móvel em nuvens (*Mobile Cloud Computing - MCC*) com enfoque nas abordagens de tomada de decisão de *offloading* . No Capítulo 3 apresentamos a proposta que norteia este trabalho. No Capítulo 4 é apresentada a etapa de projeto dos experimentos que visam validar e verificar a efetividade da proposta. No Capítulo 5 são apresentados os resultados alcançados e, por fim, no Capítulo 6 mostramos as conclusões e perspectivas futuras oriundas deste trabalho.

## 2 CONCEITOS E REVISÃO DA LITERATURA

Neste capítulo discutimos alguns conceitos e trabalhos da literatura recente que estão relacionados a esta dissertação. Damos início à discussão deste capítulo com um breve descritivo sobre os conceitos de computação móvel em nuvem (*Mobile Cloud Computing* - MCC) e *offloading*.

### 2.1 Computação Móvel em Nuvem e *Offloading*

Computação móvel em nuvem (*Mobile Cloud Computing* - MCC) integra o paradigma de computação em nuvem com tecnologias associadas à computação móvel e às redes sem fio. De acordo com (SHIRAZ *et al.*, 2015b), MCC é um modelo de computação distribuída que amplia a visão de computação utilitária das nuvens computacionais para os dispositivos móveis, superando assim suas limitações de recursos.

Dentre problemas relacionados a limitação de recursos computacionais, podemos citar a autonomia energética, capacidade de armazenamento e de processamento (DINH *et al.*, 2013). Através da MCC, é possível estender esses recursos físicos através do provisionamento de recursos virtualizados de uma infraestrutura de nuvem, seja ela local ou remota. A transferência (total ou parcial) de processamento e dados considerados de alto custo computacional para um servidor remoto é conhecida como *offloading* (FERNANDO *et al.*, 2013).

### 2.2 Trabalhos correlatos

Vários trabalhos sobre Computação Móvel em Nuvem têm sido publicados nos últimos anos. A título de exemplo, (ABOLFAZLI *et al.*, 2012) propuseram uma taxonomia MCC sob a perspectiva do dispositivo móvel, tratando o *offloading* como uma solução puramente de *software* (Figura 1).

Alguns dos trabalhos sobre o estado-da-arte em MCC são brevemente discutidos nas subseções seguintes. Os trabalhos foram organizados de acordo com a aderência à nossa proposta e subdivididos em 3 subseções: arquiteturas MCC, *frameworks* MCC e modelos em MCC para tomadas de decisão de *offloading*.

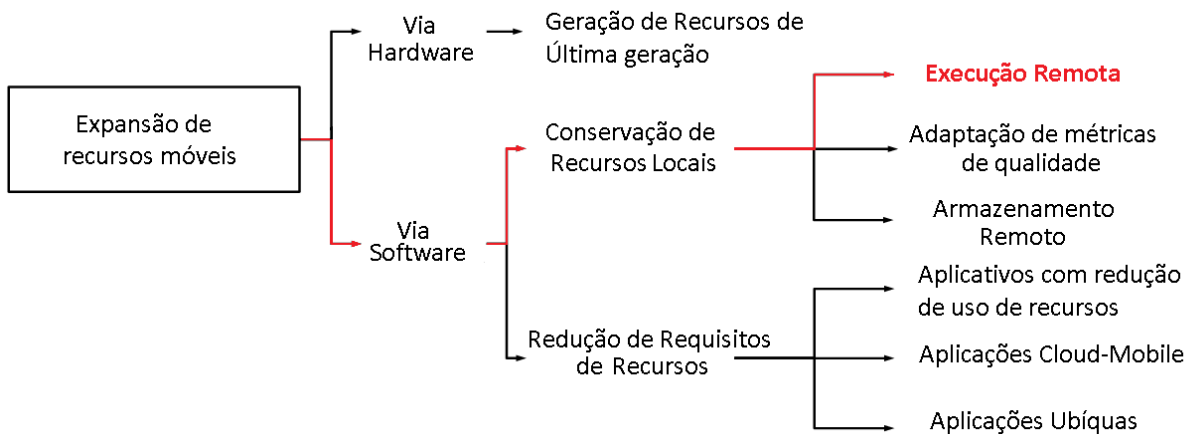


Figura 1 – Taxonomia de melhorias em MCC sob a perspectiva dos dispositivos móveis. *Offloading* destacado no fluxo em vermelho. Adaptada de : (ABOLFAZLI *et al.*, 2012)

### 2.2.1 Sobre arquiteturas de MCC

Em uma visão geral de MCC, os dispositivos móveis utilizam-se de nuvens computacionais como extensões de sua capacidade de processamento e armazenamento (QI; GANI, 2012), (DINH *et al.*, 2013) (vide Figura 2). Nesta visão geral a nuvem computacional é vista como um recurso remoto e que pode ser utilizado por dispositivos móveis.

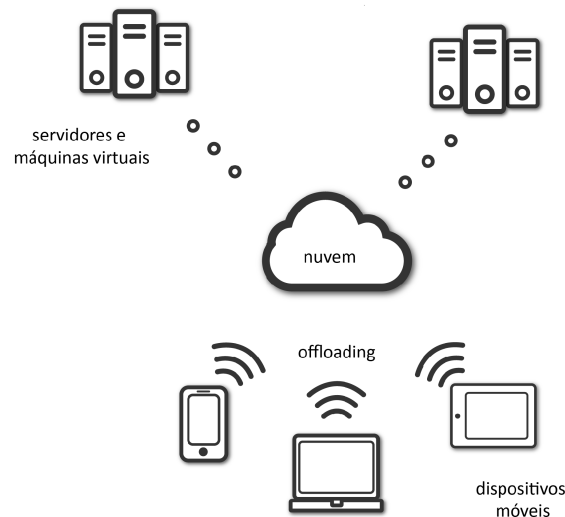


Figura 2 – Visão geral MCC. Adaptada de (QI; GANI, 2012).

(SATYANARAYANAN *et al.*, 2009) foram os que primeiro apresentaram o conceito de *cloudlet*, o qual pode ser entendido como uma nuvem computacional disponível através de uma rede local sem fio e, portanto, mais próxima ao dispositivo móvel. O uso de *cloudlets* pode ser interessante em ambientes de conectividade limitada ou inexistente (e.g. ambientes

fechados ou distantes de redes móveis) e onde a latência é um fator importante na qualidade do desempenho da rede.

(FERNANDO *et al.*, 2013) propuseram uma arquitetura que explora a utilização dos próprios dispositivos móveis disponíveis em rede para a alocação de tarefas. Em sua proposta, existem módulos para a descoberta e utilização oportunista de dispositivos móveis nesta arquitetura. Essa abordagem é baseada na premissa de alta probabilidade de problemas relacionados à conectividade entre dispositivos e pode ser considerada em casos específicos (e.g. locais remotos) em que não exista disponibilidade de redes móveis e Internet.

(MAGURAWALAGE CHATHURA M., 2014) exploraram o agrupamento de diversas *cloudlets* em uma camada intermediária entre os dispositivos e as instâncias em nuvens. Os autores adaptaram conceitos relacionados às *cloudlets* e sua abstração em relação aos dispositivos móveis. Nesta arquitetura, decisões de *offloading* entre *cloudlets* ou nuvem são feitas baseadas em critérios de energia e disponibilidade de serviços.

(RAVI; PEDDOJU, 2015) apresentaram uma arquitetura híbrida em que considera a necessidade de realização de *offloading* entre diferentes perspectivas, tais como: (i) *offloading* entre os próprios dispositivos móveis (ii) entre dispositivos móveis e *cloudlet* (iii) entre dispositivos móveis e nuvens computacionais de maneira direta. Nesta abordagem os autores também consideraram os diferentes protocolos de comunicação juntamente com o consumo energético dos mesmos para, dependendo do canal de comunicação escolhido, aprimorar as considerações energéticas no processo de *offloading*.

### 2.2.2 Sobre frameworks MCC

Alguns exemplos de *frameworks* foram propostos por (CUERVO *et al.*, 2010), (KOSTA *et al.*, 2012), (CHUN *et al.*, 2011), (SHIRAZ *et al.*, 2015a) e (COSTA *et al.*, 2015). Estes trabalhos utilizam ferramentas (e.g. MAUI, Clonecloud, Thinkair, DCOF e MpOS) que implementam características encontradas em MCC, tais como tomada de decisão de *offloading*.

Essas ferramentas gerenciam recursos os quais vão desde a virtualização de uma sistema operacional em uma máquina virtual remota até a reescrita ou edição de um aplicativo móvel. Assim, para reusar esses *frameworks* ou replicar os resultados apresentados nos respectivos artigos, é necessário conhecer desde ferramentas de manuseio de máquinas virtuais (*Virtual Machines*, VM) até o profundo manuseio de *softwares* de criação/edição de aplicativos para dispositivos móveis.

### 2.2.3 Sobre simuladores MCC

Este ainda é um tópico pouco explorado na literatura. (AMORETTI *et al.*, 2015) propuseram um ambiente de simulação MCC baseado em prerrogativas de probabilidade de execução de *offloading*. De forma complementar, recentemente propusemos a modelagem e simulação de *offloading* mediante dois tipos distintos: estático e dinâmico (SILVA Jr. *et al.*, 2015). Ambos os trabalhos utilizam-se de extensões de ambientes de simulação (DEUS (AMORETTI *et al.*, 2009) e CloudSim(CALHEIROS *et al.*, 2011)), respectivamente.

Nas propostas de simulação, observamos duas vertentes: a primeira, de (AMORETTI *et al.*, 2015), com um teor estocástico. Ou seja, a modelagem do *offloading* ocorre mediante uma dada probabilidade de sua ocorrência em um ambiente MCC. A segunda, de (SILVA Jr. *et al.*, 2015), é orientada a eventos e possui a prerrogativa da simulação de realização de *offloading* mediante indicação em código (através de instâncias de classes indicadoras para tal) ou de decisão favorável para tal através de algum algoritmo de tomada de decisão.

### 2.2.4 Sobre modelagem analítica e de *offloading*

O *survey* de (KUMAR *et al.*, 2013) trata de diferentes abordagens de *offloading* e sua utilização para poupar recursos energéticos. Os autores ainda propuseram uma modelagem de *offloading* dinâmico que leva em conta banda e quantidade de computação das procedimentos passíveis de execução remota.

(MIETTINEN; NURMINEN, 2010) propuseram um modelo para realização de *offloading* baseado na seguinte inequação:

$$E_{envioERcebimentoCloud} < E_{local}, \quad (2.1)$$

em que  $E_{envioERcebimentoCloud}$  representa o custo energético de transmissão e recepção de dados para execução do procedimento em nuvem e  $E_{local}$  representa a energia consumida unicamente no dispositivo móvel em uma execução local. Entretanto, os autores limitaram os testes de validação do modelo com dispositivos de um único fabricante sendo, portanto, de difícil generalização. Em nossa proposta, utilizamos medições realizadas considerando diferentes tipos de procedimentos de acordo com sua complexidade algorítmica. Acreditamos que a variação do grau de complexidade, bem como a variação da parametrização de tarefas a serem realizadas, permitam uma maior generalização.

(SEGATA *et al.*, 2014) propuseram um modelo de *offloading* linear que mostra a razão entre consumo médio de energia por dados em meios de transmissão distintos (2G, 3G e WiFi). Através de medições realizadas em diversos dispositivos, os autores apresentaram a Equação 2.2

$$y[J] = \beta x[MB], \quad (2.2)$$

sendo  $y[J]$  o consumo total dado pelo produto do consumo médio por Megabyte ( $\beta$ ) (simbolizando a relação de Joule por Megabyte) e a quantidade de dados transmitida ( $x[MB]$ ). É importante mencionarmos que o fator  $\beta$  varia conforme a rede escolhida e o tipo de transmissão (*download* ou *upload*).

(WU *et al.*, 2013) propuseram uma modelagem analítica de *offloading* levando em consideração o consumo de energia de dispositivos móveis de acordo com a Equação 2.3

$$P_{phone} = P_{CPU}(c) + P_{RF}(c), \quad (2.3)$$

em que, dado uma configuração  $c$  de hardware,  $P_{phone}$  representa o total de energia gasto pelo dispositivo móvel na execução de uma tarefa,  $P_{CPU}$  é o total de energia gasta por seu processador e  $P_{RF}$  pela sua interface de comunicação. Essas entidades, dependendo da realização ou não do *offloading*, podem apresentar estados diferentes de utilização. Os autores denominam esses estados de inativo (*idle*) e ativo (*active*). Visto que questões energéticas são imprescindíveis para ambientes MCC, em nossa proposta nós também consideramos a grandeza Consumo Energético.

(ZHANG *et al.*, 2013) apresentaram uma modelagem considerando aspectos de natureza estocástica para execução de procedimentos em dispositivos móveis ou em nuvens computacionais. Os autores consideraram a modelagem de execução de procedimentos no âmbito do dispositivo móvel e a modelagem de energia gasta durante o envio de dados para a nuvem. Neste trabalho, notamos a distinção destes dois momentos (processamento e transmissão) na proposta dos autores.

Os trabalhos relacionados resumem os estudos e pesquisas em MCC. A Tabela 2 sumariza esses trabalhos de acordo com sua contribuição primária em questão no contexto MCC.

Tabela 2 – Síntese dos trabalhos correlatos.

	A	B	C	D	E
(DINH <i>et al.</i> , 2013)				✓	
(QI; GANI, 2012)	✓				
(FERNANDO <i>et al.</i> , 2013)	✓	✓		✓	
(SATYANARAYANAN <i>et al.</i> , 2009)	✓				
(RAVI; PEDDOJU, 2015)	✓				
(KUMAR <i>et al.</i> , 2013)			✓	✓	
(SEGATA <i>et al.</i> , 2014)			✓	✓	
(COSTA <i>et al.</i> , 2015)		✓			
(KOSTA <i>et al.</i> , 2012)		✓			
(CHUN <i>et al.</i> , 2011)		✓	✓		
(CUERVO <i>et al.</i> , 2010)		✓			
(MIETTINEN; NURMINEN, 2010)			✓		
(LIU <i>et al.</i> , 2015)			✓		
(ABOLFAZLI <i>et al.</i> , 2012)			✓		
(AMORETTI <i>et al.</i> , 2015)					✓
(SILVA Jr. <i>et al.</i> , 2015)					✓
(ZHANG <i>et al.</i> , 2013)			✓		
(MAGURAWALAGE CHATHURA M., 2014)	✓				
(SILVA <i>et al.</i> , 2016)				✓	
(WU <i>et al.</i> , 2013)			✓		

Legenda: A) Arquiteturas e cenários MCC; B) *Frameworks*; C) Modelagem de *offloading*; D) *Surveys*, taxonomias; E) Simuladores.

### 2.3 Resumo do capítulo

Este capítulo discutiu alguns trabalhos da literatura sobre MCC. Existem diversas propostas que contemplaram a discussão e ampliação das arquitetura de MCC existentes. Outros trabalhos propuseram melhorias na tomada de decisão de *offloading* com o intuito de obter uma melhor relação, em termos de economia de recursos, para a execução de procedimentos. Existem trabalhos que trouxeram propostas de diversos *frameworks* MCC. Esses trabalhos dissertaram acerca de propostas para implementação de técnicas de MCC e melhoria de aplicações existentes. Ainda, alguns trabalhos propuseram a modelagem analítica de diversos elementos de MCC. Esses trabalhos relacionaram entidades para, sobretudo, o benefício energético na realização de *offloading* em MCC.



### 3 PROPOSTA

A proposta central deste trabalho consiste na elaboração de um mecanismo de tomada de decisão de *offloading* que relaciona o consumo energético de procedimentos, as conexões existentes (3G, WiFi) para execução remota, a complexidade algorítmica e dados de entrada e saída. Ao final deste capítulo será possível a compreensão das entidades que compõem a proposta, bem como os algoritmos que serão considerados e discutidos para realização da tomada de decisão de *offloading*.

#### 3.1 Mecanismo proposto

A fim de alcançar os objetivos elencados no Capítulo 1, as seguintes hipóteses foram adotadas: (i) o grau de complexidade algorítmica é relevante para medição de consumo energético em casos de *offloading*; (ii) a variável “consumo energético” é utilizada na tomada de decisão de *offloading* em ambiente de nuvem, *cloudlet* e em outros dispositivos móveis; (iii) a *cloudlet* e um dispositivo móvel passível de *offloading* devem estar na mesma rede local (RAVI; PEDDOJU, 2015) (SATYANARAYANAN *et al.*, 2009). Em nossa proposta, a execução de procedimentos externos ao dispositivo móvel se distingue basicamente através do tipo de conectividade utilizada durante a realização de *offloading*.

O mecanismo segue a mesma relação descrita na Inequação 2.1. Entretanto, visto que o *offloading* ocorre entre diferentes infraestruturas: nuvem, *cloudlet* e outros dispositivos móveis (RAVI; PEDDOJU, 2015), em nossa proposta um dado procedimento possui as seguintes possibilidades de execução: (i) Execução puramente local (ii) Execução remota em nuvem e (iii) Execução remota em *cloudlet*.

A Figura 3 ilustra os passos realizados para decisão do local onde uma tarefa submetida pode ser executada. A decisão de qual ambiente executar uma determinada tarefa é realizada através de estimativas de dados de consumo energético obtidos de acordo com cada local de execução (ambiente local, *cloudlet* ou nuvem). Para realizarmos a escolha do local mais apropriado para execução da tarefa, sob o ponto de vista energético, os Algoritmos 1, 2 e 3 foram desenvolvidos.

O Algoritmo 1 é utilizado no mecanismo para tomada de decisão de *offloading* proposto e possui os seguintes parâmetros de entrada: (i)  $\theta$ : uma variável que indica a ordem de complexidade do procedimento a ser analisado, e que pode assumir valores enumeráveis como:

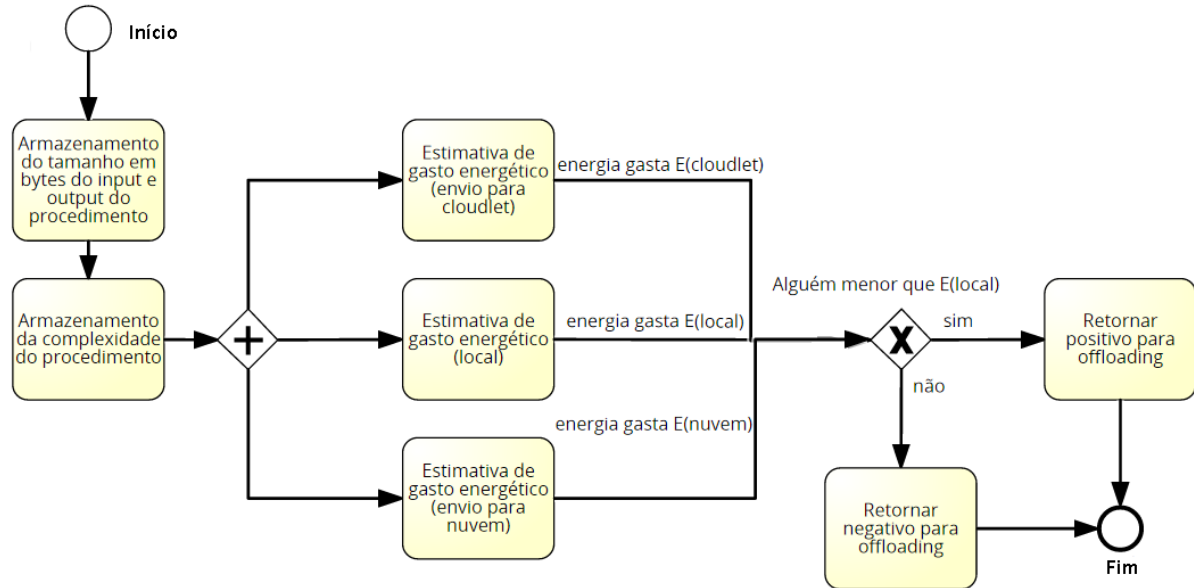


Figura 3 – Diagrama ilustrativo de tomada de decisão de *offloading* proposta.

$k$  (constante),  $\log n$ ,  $n$ ,  $n^2$  e  $n^3$ ; (ii) In e Out: indicam, respectivamente, o tamanho em Kbyte da entrada e saída do procedimento que será analisado.

No Algoritmo 1, a estimativa de consumo energético é fornecida às variáveis *localEnergy*, *wifiEnergy* e *mobileNetworkEnergy* através de chamadas de sub-rotinas descritas nos Algoritmos 2 e 3. Após as atribuições das variáveis com suas respectivas estimativas de consumo energético, a comparação entre seus valores é realizada no Algoritmo 1 em suas instruções seguintes, onde a tomada de decisão realizada pelo Algoritmo é atribuída à variável *result*, sendo essa uma *Enumeration* representativa do possível local para execução da tarefa: Ambiente local, *cloudlet* ou nuvem. Essa comparação de valores baseia-se nas Inequações 2.1 e 3.1.

Após sua execução, o Algoritmo 1 retorna um tipo *Enumeration* com três valores possíveis: *Local*, *Wifi* e *MobileNetwork*. Esses valores indicam o local onde o procedimento pode ser executado no contexto do *offloading*. O resultado do Algoritmo 1 informa ao *framework* ou simulador MCC em que local o procedimento será realizado.

Os Algoritmos 2 e 3 são sub-rotinas chamadas pelo Algoritmo 1 para captação de estimativas de consumo energético. O Algoritmo 2 foi implementado e baseado em medições realizadas neste trabalho que propusemos para relacionar consumo energético e ordem de complexidade algorítmica. Já o Algoritmo 3 retorna valores baseados em estimativas de consumo descritas em (SEGATA *et al.*, 2014). Os valores calculados possuem como base valores médios de  $\beta$  (simbolizando a relação de Joule por Megabyte) para download de 0.46, em WiFi, e 7.01,

em 3G e upload de 0.99, em WiFi, e 9.88, em 3G para os casos de transmissão via WiFi e 3G.

É válido mencionarmos que os Algoritmos 1, 2 e 3 levam em consideração tarefas com complexidade algorítmica distintas entre si através de estruturas de decisão existentes internamente. Para efeito de comparação, utilizamos algoritmos de busca (Busca binária e sequencial), ordenação (*Bubble sort*) e operações matriciais (Produto de matrizes) para verificação e validação da proposta.

As Figuras 4 e 5 mostram a tendência de crescimento do consumo energético em relação ao tamanho da entrada nos algoritmos Busca binária, sequencial, *Bubblesort* e *Produto de matrizes*. Esses valores foram captados através da execução dos algoritmos citados com entradas entre 100 a 100000 de tamanho (Busca binária, sequencial, *Bubblesort*) e entre 100 a 1000 (produto de matrizes) juntamente com a captação de estimativas de consumo pelo uso da ferramenta Trepn Profiler<sup>1</sup>. O intervalo de confiança dos valores obtidos é de 95%. Entretanto, para os algoritmos de Busca binária e sequencial, a verificação de consumo energético não apresentou variações significativas e, para efeitos de comparação, foi simplificada para um consumo próximo de zero conforme ilustrado na Figura 4 e no Algoritmo 2.

Em nosso trabalho, observamos que a grandeza *Consumo Energético* cresce de maneira proporcional ao tamanho da entrada dos algoritmos utilizados de ordem igual a  $O(n^2)$  e  $O(n^3)$ . Entendemos que, conforme o tamanho da entrada aumenta, o gasto energético para a execução do algoritmo e seu tempo também sofre aumento. Isso pode ser visto através das Figuras 4 e 5 as quais ilustram a relação entre essas dimensões em um contexto de execução local.

No mecanismo proposto, para realização da tomada de decisão de *offloading*, optamos por utilizar algoritmos que interferissem minimamente no consumo de recursos computacionais e energéticos. Desse modo, a regressão linear é a técnica utilizada para modelar a relação entre o valor de entrada de um algoritmo de  $O(n^2)$  e  $O(n^3)$  e seu custo energético. Tal técnica foi escolhida por ser utilizada para modelagem em ambientes de menor poder computacional e sistemas embarcados (CARVALHO *et al.*, 2011).

Para que o Algoritmo 1 retorne uma tomada de decisão em favor do *offloading*, o Algoritmo 3 deve retornar valores de consumo energético menores que o retornado pelo

<sup>1</sup> Disponível em: <https://developer.qualcomm.com/software/trepn-power-profiler>

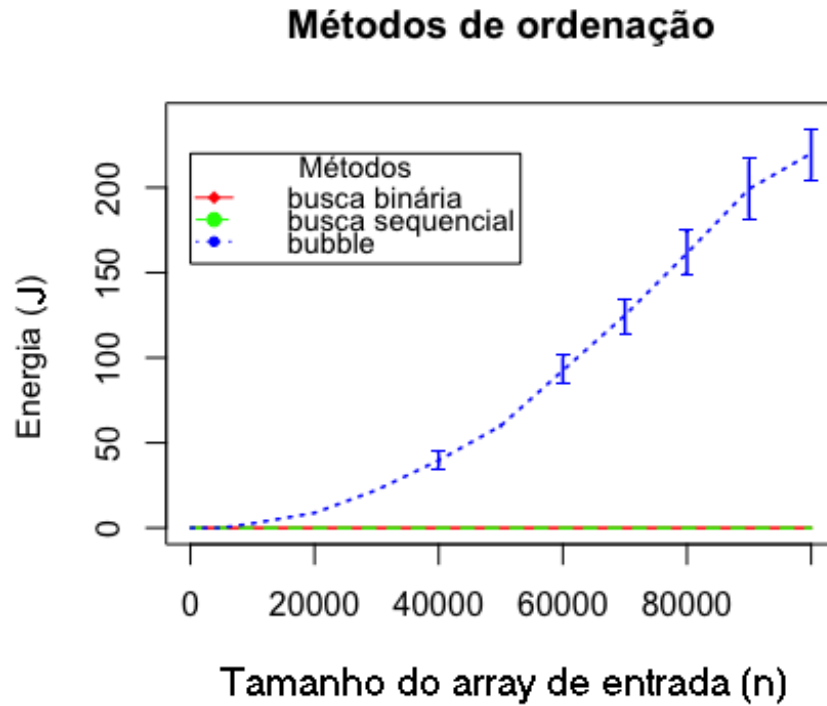


Figura 4 – Consumo energético local - *Busca binária*, *Busca sequencial* e *Bubble Sort*.

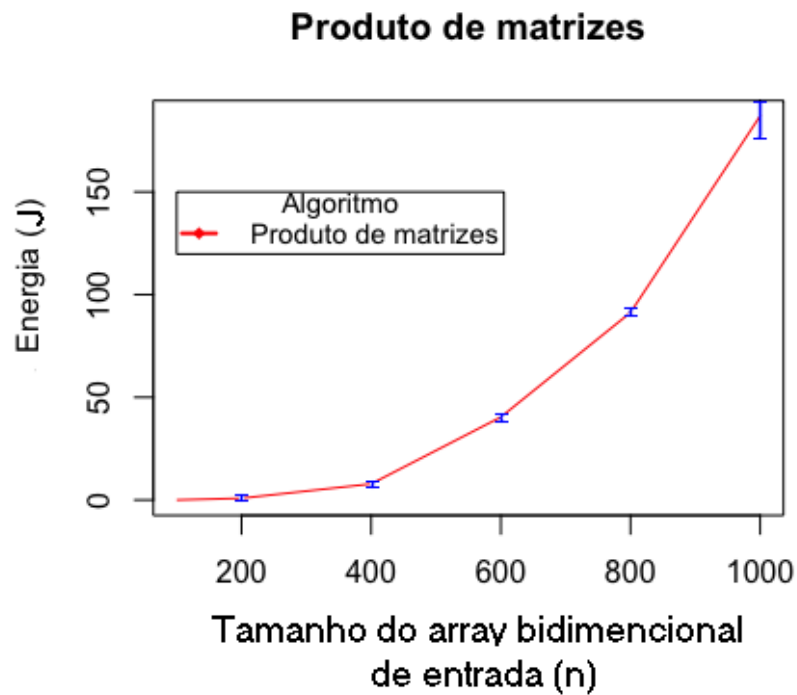


Figura 5 – Consumo energético local - *Produto de matrizes*

---

**Algoritmo 1:** Procedimento para tomada de decisão de *offloading*. Nomenclatura: (getOffloadingDecision)

---

**Entrada:**  $\theta$ , In, Out

**Saída:** result

**início**

$localEnergy \leftarrow getEnergyEstimationLocal(\theta, In, Out);$

$wifiEnergy \leftarrow getEnergyEstimation(Wifi, In, Out);$

$mobileNetworkEnergy \leftarrow getEnergyEstimation(3G, In, Out);$

$result \leftarrow null;$

**se**  $localEnergy < wifiEnergy$  e  $localEnergy < mobileNetworkEnergy$  **então**

    # condicional que resulta em execução local

$result \leftarrow local;$

**fim**

**senão**

**se**  $wifiEnergy < localEnergy$  e  $wifiEnergy < mobileNetworkEnergy$  **então**

        # condicional que resulta em execução via WiFi (e.g. *cloudlet* ou outro

        dispositivo

$result \leftarrow Wifi;$

**fim**

**senão**

$result \leftarrow mobileNetwork;$

        # caso contrário, executa remotamente em nuvem

**fim**

**fim**

return  $result;$

**fim**

---

Algoritmo 2, independente do tipo de rede, conforme a comparação abaixo.

$$E_{envioERecebimentoParaCloudlet} < E_{local} \quad (3.1)$$

O consumo de energia utilizada na execução local de um procedimento qualquer precisa ser maior do que o consumo de energia na transmissão e recepção de dados para a nuvem computacional, *cloudlet* ou outro dispositivo móvel para realização de *offloading*. A Figura 3 ilustra e reforça a comparação de diferentes gastos energéticos que são descritos no Algoritmo 1.

Ainda, na Figura 3 estão contemplados passos intermediários para a tomada de decisão. Os primeiros dois blocos representam o armazenamento de informações sobre os

---

**Algoritmo 2:** Procedimento para obtenção de estimativa de consumo energético local.

Nomenclatura: (getEnergyEstimationLocal)

---

**Entrada:**  $\theta$ , In, Out

**Saída:** energy

**início**

energy  $\leftarrow$  0;

linearCoefficientAlgorithmON2  $\leftarrow$  -9,9063;

angularCoefficientAlgorithmON2  $\leftarrow$  0,00211;

linearCoefficientAlgorithmON3  $\leftarrow$  -71,254;

angularCoefficientAlgorithmON3  $\leftarrow$  0,22787;

**se**  $\theta == 1$  **ou**  $\theta == \log n$  **ou**  $\theta == n$  **então**

    energy  $\leftarrow$  0;

    return energy;

**fim**

**senão**

**se**  $\theta == n2$  **então**

        energy  $\leftarrow$  (angularCoefficientAlgorithmON2 \* (In+Out) +

        linearCoefficientAlgorithmON2); # Regressão linear da Figura 4;

        return energy;

**fim**

**senão**

**se**  $\theta == n3$  **então**

            energy  $\leftarrow$  (angularCoefficientAlgorithmON3 \* (In+Out) +

            linearCoefficientAlgorithmON3); # Regressão linear da Figura 5;

            return energy;

**fim**

**senão**

            return energy;

**fim**

**fim**

**fim**

**fim**

---

---

**Algoritmo 3:** Procedimento para obtenção de estimativa de consumo energético via rede móvel ou WiFi. Nomenclatura: (getEnergyEstimation)

---

**Entrada:** networkType, In, Out

**Saída:** energy

**início**

energy  $\leftarrow$  null;

**se** NetworkType == 3G **então**

    In3G $_{\beta}$   $\leftarrow$  9.88;

    Out3G $_{\beta}$   $\leftarrow$  7.01;

    energy  $\leftarrow$  In $_{\beta}$  \* In + Out $_{\beta}$  \* Out;

**fim**

**senão**

    InWifi $_{\beta}$   $\leftarrow$  0.99;

    OutWifi $_{\beta}$   $\leftarrow$  0.46;

    energy  $\leftarrow$  In $_{\beta}$  \* In + Out $_{\beta}$  \* Out;

**fim**

return energy;

**fim**

---

procedimentos que podem sofrer *offloading*. Os dados obtidos, conforme descrito no Algoritmo 2, possibilitam a estimativa do consumo energético de procedimentos localmente realizados. As estimativa de consumo em cada âmbito são comparadas conforme as Inequações 2.1 e 3.1 e respectivas implementações ns Algoritmo 3 conforme (SEGATA *et al.*, 2014). Finalmente, a decisão de *offloading* é tomada através da execução do Algoritmo 1.

Caso o retorno do Algoritmo 1 indique que deve ser realizado *offloading*, é necessária a existência de conectividade pelo dispositivo móvel no meio selecionado. Caso contrário, o procedimento será executado localmente. As oscilações do estado de conexão no momento da realização do *offloading* e a localização geográfica dos dispositivos móveis não estão presentes no escopo da proposta, neste primeiro momento. Apesar disso, entendemos que a consideração de localização seria importante para reforçar a atuação do mecanismo proposto em cenários onde podem existir: (i) mudanças de contexto com base na localização; (ii) oscilações em largura de banda disponível; (iii) interrupção e retomada de conexão em redes móveis.

Destacamos ainda que a economia de recursos energéticos em nossa proposta está ligada somente ao contexto local, ou seja, o mecanismo proposto não considera gastos energéticos

na execução de procedimentos em outras fontes que não sejam o próprio dispositivo. A única consideração energética neste aspecto ocorre durante a etapa de transmissão e recepção de dados para outros ambientes de execução (e.g. nuvens e *cloudlets*).

### 3.2 Simulação de ambiente MCC

Para simularmos o mecanismo proposto e compararmos com sua implementação em um *framework* real, realizamos a modelagem dos componentes MCC através da especialização de componentes internos do CloudSim (CALHEIROS *et al.*, 2011). CloudSim é um simulador de nuvens computacionais através de eventos discretos. Devido a característica de representar e simular o comportamento de infraestrutura de nuvem e sua alocação de tarefas, estendemos seus componentes. Não obstante, podemos afirmar que o nosso modelo está desacoplado de ferramentas e soluções externas, uma vez que nenhum componente interno do CloudSim teve seu comportamento original alterado.

A simulação contempla, atualmente, um típico cenário de MCC (*vide* Figura 2). A modelagem possui como referência o *offloading* do tipo estático ou dinâmico (KUMAR *et al.*, 2013) (DINH *et al.*, 2013). Neste sentido, propusemos a criação de cinco modelos para representarmos entidades existentes na arquitetura em questão: (i) dispositivo móvel em contexto MCC (ii) tarefa passível de *offloading*, (iii) mecanismo de *offloading* estático e dinâmico, (iv) banda entre dispositivos e nuvem e, (v) infraestrutura de nuvem como um *pool* de servidores e VMs para execução remota de procedimentos.

A Figura 6 ilustra um diagrama de classes das entidades criadas. Essas classes representam os componentes ilustrados na Figura 2. Sua associação com a modelagem proposta é ilustrada na Figura 7, na qual é possível distinguir quais classes representam quais modelos. Já a simulação e interação entre as classes da Figura 7 é representada na Figura 8.

O modelo de dispositivos móveis é representado pelas classes *Device* e *DeviceTaskExecutor* (Figura 7(a)). *Device* contém informações sobre um dado dispositivo como: modelo (atributo *DeviceHardware*) e identificador único (e.g. IMEI). *DeviceTaskExecutor* possui uma operação chamada de *executeTasksOnDevice* a qual é responsável por simular a execução de uma dada tarefa em um dispositivo móvel. Arelada a um *Device*, a estrutura *taskList* contém uma lista de tarefas.

As classes *Offloader*, *StaticOffloader* e *DynamicOffloader* (Figura 7(b)) possuem informações sobre a natureza de *offloading* a que uma tarefa é submetida. Essas classes imple-



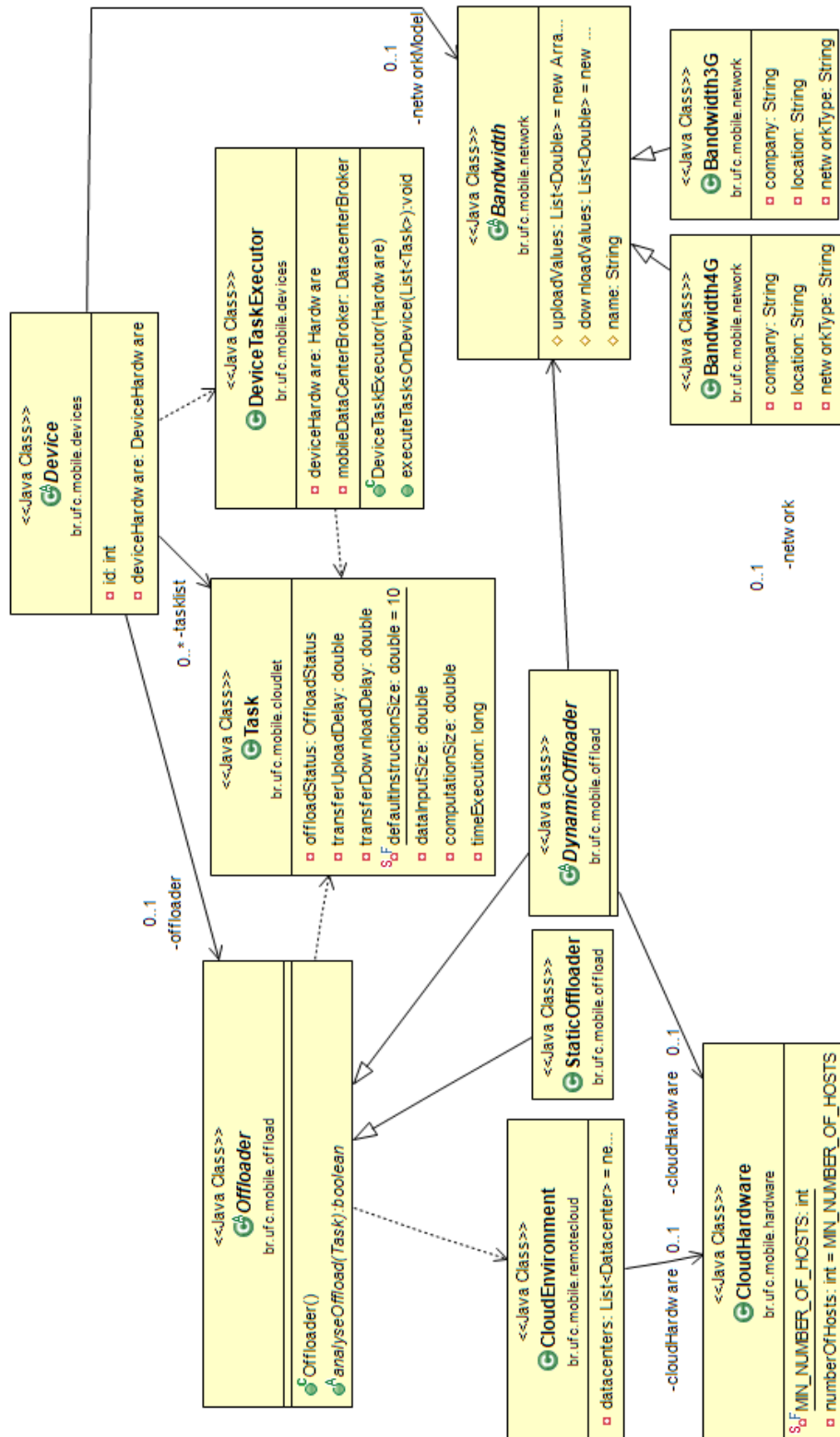


Figura 6 – Diagrama de classes dos modelos criados.

mentam e representam os modelos de mecanismos de tomada de decisão de *offloading* estático e dinâmico. *DynamicOffloader* é uma especialização com implementação de *offloading* dinâmico em termos da proposta de (KUMAR *et al.*, 2013) e simplificada por (COSTA *et al.*, 2014). Seu comportamento segue a Equação 3.2.

$$\frac{W}{P_m} > \frac{D_u}{V_u} + \frac{D_d}{V_d}, \quad (3.2)$$

em que  $W$  representa (em MIPS) a quantidade de processamento realizado por uma dada tarefa,  $P_m$  representa o poder computacional de um dispositivo móvel,  $D_u$  é a quantidade de dados enviados à nuvem,  $D_d$  é quantidade de dados recebidos,  $V_u$  significa a taxa de *upload* e, de maneira análoga,  $V_d$  é a taxa de *download*. Outra especialização de *Offloader*, a classe *StaticOffloader* implementa uma marcação na tarefa para informar que a mesma sofrerá *offloading*.

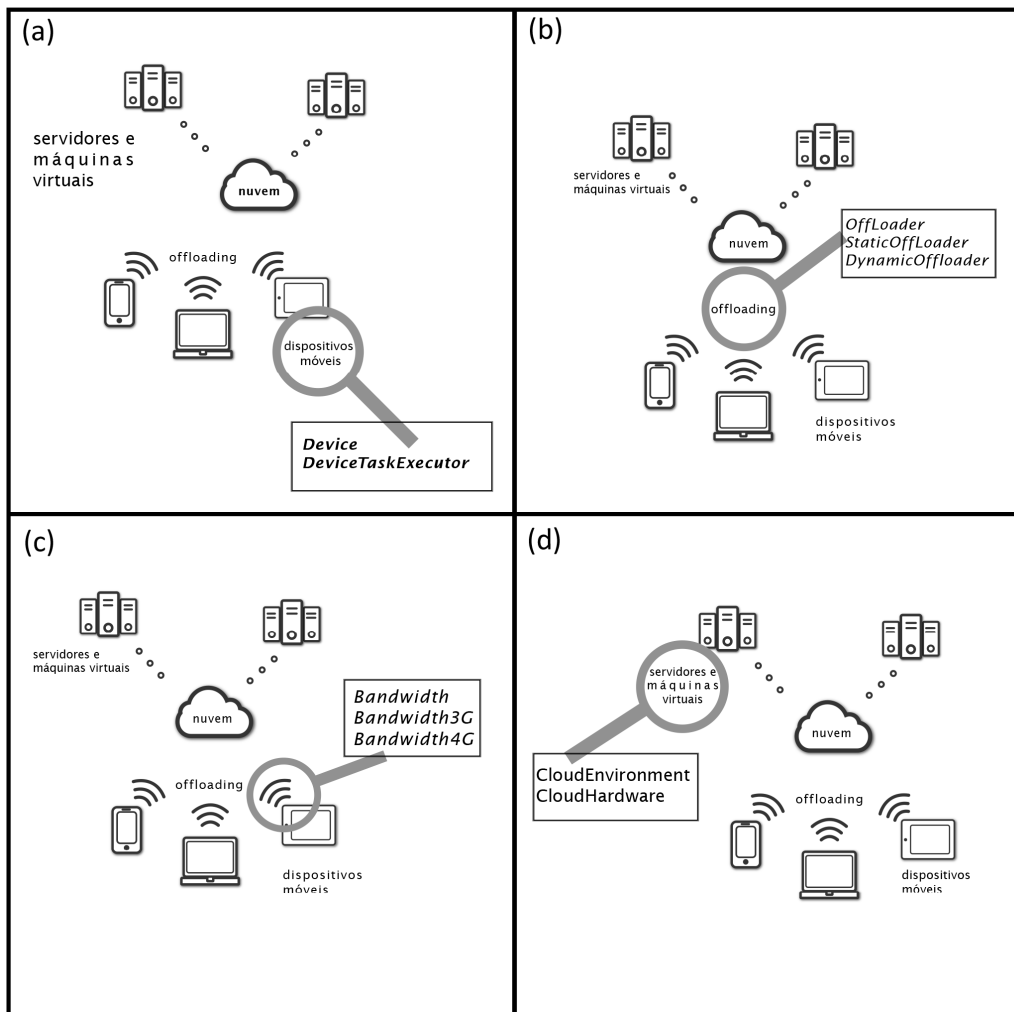


Figura 7 – Associação da modelagem proposta entre as respectivas classes e (a) Dispositivos móveis, (b) Tomada de decisão de *offloading*, (c) Rede sem fio, (d) Infraestrutura de nuvem.

Ambas as especializações da classe *Offloader* precisam ser referenciadas por um objeto do tipo *Task* e isso é necessário para o cálculo do tempo total de execução do processo de *offloading*, independente de sua especialização, o qual é expresso pela Equação 3.3:

$$T_{Total} = T_{Upload} + T_{Execucao} + T_{Download}, \quad (3.3)$$

na qual ( $T_{Upload}$  e  $T_{Download}$ ) expressam o tempo de envio de informações necessárias ao *offloading* e recebimento de resultados oriundos da nuvem, respectivamente.  $T_{Execucao}$  representa o tempo de uma tarefa executada local ou remotamente. Notar que na ausência de *offloading*,  $T_{Upload}$  e  $T_{Download}$  são nulos e somente  $T_{Execucao}$  representa o tempo para execução da tarefa no dispositivo móvel.

Os modelos de banda são representados pelas entidades *Bandwidth*, *Bandwidth3G* e *Bandwidth4G* (Figura 7(c)) as quais são baseadas nos dados oriundos de taxas de transmissão de dados 3G e 4G (COSTA *et al.*, 2014). Ainda, a classe *Device* possui um atributo do tipo *Bandwidth* (*networkData*) e, com sua inicialização, o dispositivo consegue calcular o tempo que uma tarefa leva ao ser realizado seu *download* e *upload*. Essa informação é utilizada durante a simulação para cálculo do tempo final de execução.

A *modelagem de uma infraestrutura de nuvem* é necessária para a simulação de execução de uma *tarefa* em uma nuvem computacional. *CloudHardware* e *CloudEnvironment* implementam esse modelo e possuem informações sobre o tempo de execução remota do procedimento e sobre a quantidade de VMs (Figura 7(d)). Neste trabalho, a informação mais relevante no lado servidor é o tempo de execução remota da tarefa. Essa informação varia de acordo com o tipo de nuvem a ser considerada em simulação conforme (COSTA *et al.*, 2014). Desta forma, nossos modelos contemplam os casos de execução em nuvens (instância Amazon tipo Micro e Medium) e em dispositivos móveis.

As entidades exibidas na Figura 7 se relacionam para seguir um fluxo de interação conforme a Figura 8, que mostra a sequência de instanciação para simulação de *offloading* em MCC. Primeiramente, uma instância responsável pela simulação de execução da tarefa no próprio device (*DeviceExecutor*) e em nuvem (*CloudExecutor*) é inicializada. Logo em seguida, a instância da tarefa (*Task*) é criada, com suas propriedades ajustadas (e.g. tamanho em bytes, número de instruções executadas por segundo) e atreladas com uma instância representante de um dado dispositivo (*Device*).

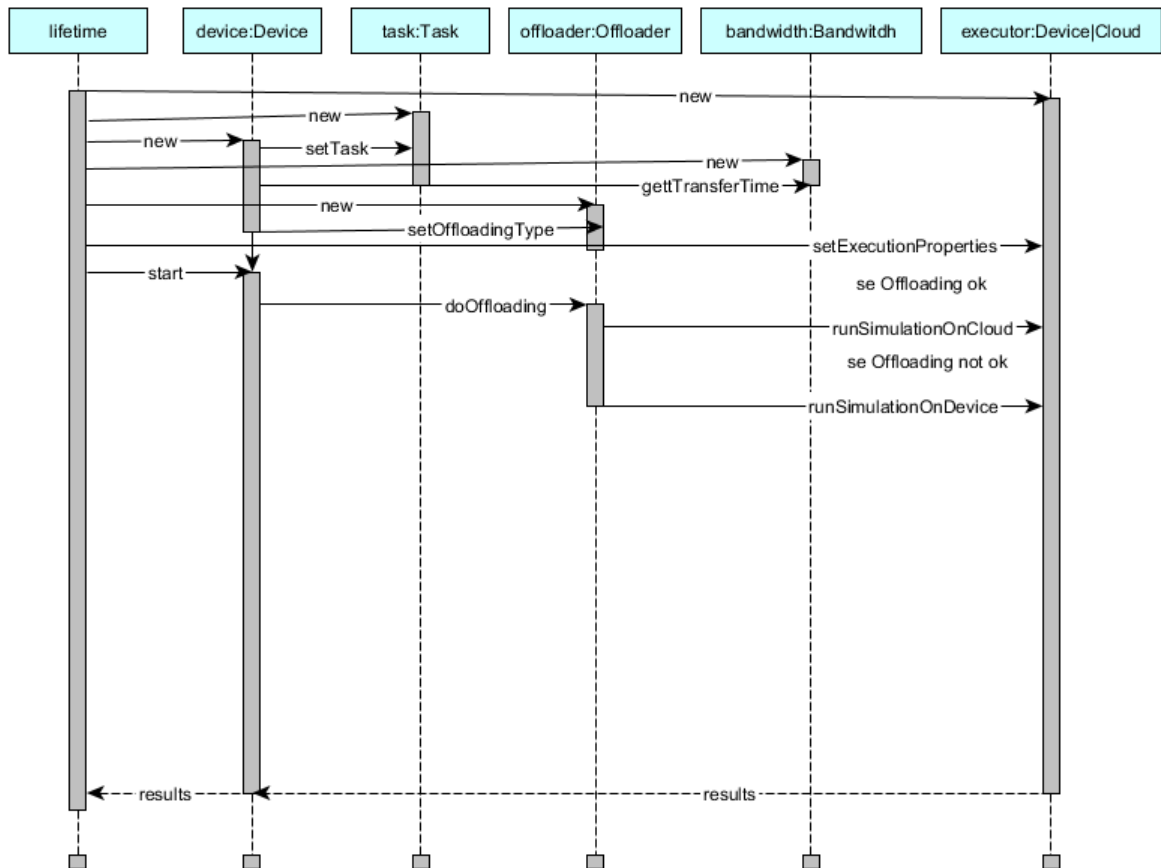


Figura 8 – Diagrama de sequência para ilustração de chamada de modelo para tomada de decisão de *offloading*.

A seguir a instância da largura de banda desejada é criada (e.g *Bandwidth3G*). Seu método *getTransferTime()* é invocado quando necessário saber uma estimativa de tempo de transmissão em caso de *offloading*. A instância do modelo para tomada de decisão de *offloading* é criada e atrelada à instância do dispositivo, caso seja necessária sua utilização. Em seguida, de acordo com as propriedades da tarefa, o executor é parametrizado e, finalmente, o método *doOffloading(...)* é chamado. Em caso positivo, ou seja, numa decisão favorável (tratando-se de *offloading* dinâmico) o *offloading* deve ser simulado numa instância de execução em nuvem e, caso contrário, em uma de instância de execução em dispositivo móvel.

Outro aspecto a ser mencionado é a possibilidade de extensão e customização de componentes de nossa proposta como a inclusão de novos mecanismos de *offloading* e novas modelagens de banda e de outros dispositivos móveis.

### 3.3 Resumo do capítulo

Este capítulo apresentou a proposta que norteia este trabalho, bem como as entidades que compuseram sua arquitetura e a relação entre elas. O mecanismo de tomada de decisão de *offloading* foi apresentado juntamente com os algoritmos que o compõe. Os algoritmos propostos, estimam o consumo energético de procedimentos em âmbito de execução local e execução remota via rede 3G e WiFi. Com isso, os algoritmos realizam o chaveamento do contexto mais apropriado para execução de procedimentos, baseando-se no tamanho das entradas das tarefas a serem executadas e sua complexidade algorítmica. Ainda, os modelos criados para simulação de um cenário MCC foram apresentados e os relacionados entre esses foi discutido para compreensão em um contexto de simulação.

## 4 PLANEJAMENTO DOS EXPERIMENTOS

Neste capítulo, descrevemos o planejamento dos experimentos realizados, quatro ao todo. Os dois primeiros foram concebidos para validação dos modelos que representam um cenário típico em MCC. No primeiro experimento, o tempo total de execução foi utilizado como métrica para verificação de acurácia dos modelos. No segundo experimento, o modelo para tomada de decisão aplica a equação de (KUMAR *et al.*, 2013) com o intuito de validar a tomada de decisão de *offloading*.

Nos dois últimos experimentos, validamos o mecanismo proposto de tomada de decisão de *offloading*. No terceiro experimento, a implementação do mecanismo no *framework* MpOS foi comparada com a própria simulação utilizando os modelos de *offloading* elaborados no Capítulo 3. No quarto experimento, uma comparação entre os algoritmos de *offloading* proposto e o de (KUMAR *et al.*, 2013) foi realizada para comparação de desempenho em redução de consumo energético.

### 4.1 Experimentos para validação de modelos em MCC

Para validação da modelagem das entidades em MCC, dois experimentos foram projetados e executados em distintos contextos de realização de *offloading*. O Experimento #1 visou a validação da precisão no cálculo do tempo total de execução tratando exclusivamente de *offloading* estático. O Experimento #2 foi projetado com o intuito de validar a tomada de decisão do *offloading* dinâmico. Ambos estão descritos nas duas subseções seguintes.

#### 4.1.1 Experimento #1: Tempo total de execução em *offloading* estático

O Experimento #1 utilizou uma simulação de *offloading* estático para delegar a execução da aplicação BenchImage<sup>1</sup> à nuvem. Essa aplicação foi desenvolvida por (COSTA *et al.*, 2014) para aplicação de filtros em imagens através da utilização de uma nuvem computacional, *cloudlet* ou execução local para tal, mediante decisão tomada pelo usuário.

Ainda, o experimento em questão contou com dois modelos de smartphones e dois tipos de instância da Amazon EC2 (Micro e Medium) configuradas com o Ubuntu Server 12.04 64-bit. O modelo de largura de banda de rede 4G foi utilizado no experimento para simular a comunicação entre o dispositivo móvel e a nuvem. Neste modelo, as taxas de *upload* e *download*

<sup>1</sup> Disponível em: <https://play.google.com/store/apps/details?id=br.ufc.mdcc.benchimage>

foram baseadas em (COSTA *et al.*, 2014).

Neste experimento, simulamos uma aplicação de filtro de imagens *Cartoonizer* aplicado à imagens com tamanhos distintos. Os fatores dos experimentos e seus valores (níveis) estão nas Tabelas 3 e 4. Durante a execução dos cenários, o tempo total de execução da aplicação (Equação 3.3) foi calculado para fins de comparação com os resultados alcançados em (COSTA *et al.*, 2014).

#### 4.1.2 Experimento #2: Tomada de decisão em offloading dinâmico

O Experimento #2 consistiu na tomada de decisão do *offloading* dinâmico com base nos seguintes fatores: quantidade de computação realizada e quantidade de comunicação dispositivo-nuvem (KUMAR *et al.*, 2013) para cada algoritmo (Busca binária, *bubble sort* e produto de matrizes). A comunicação entre o *smartphone* e a nuvem foi modelada considerando redes 3G e 4G. Nestes modelos de largura de banda, as taxas de *upload* e *download* são geradas segundo dados apresentados em (COSTA *et al.*, 2014) Ao adicionarmos modelos com taxa de transmissão compatível com redes 3G, consideramos que a variação desta taxa mostra maior veracidade em cenários de decisão corriqueiros para os usuário de aplicativos, sobretudo no Brasil (Tabelas 3 e 4).

## 4.2 Experimentos com o mecanismo proposto

Para criação do modelo de tomada de decisão de *offloading* foram coletados dados referentes ao consumo energético durante a execução dos procedimentos *cpu-bound* (*Busca*

Tabela 3 – Critérios para o projeto dos experimentos

Itens	Descrição
Métricas	<b>Experimento 1:</b> tempo total de execução; <b>Experimento 2:</b> ocorreu/não ocorreu migração
Parâmetros	<b>Experimento 1:</b> filtro da imagem, tipo de <i>offloading</i> , tipo da rede; <b>Experimento 2:</b> tipo de <i>offloading</i> , modelo do dispositivo móvel, tipo da instância, taxa de download e upload
Fatores	<b>Experimento 1:</b> tamanho da imagem, modelo do dispositivo móvel, tipo da instância; <b>Experimento 2:</b> quantidade de comunicação, quantidade de computação, tipo da rede
Técnica de análise	Simulação via CloudSim
Carga de Trabalho	<b>Experimento 1:</b> BenchImage; <b>Experimento 2:</b> Busca binária, <i>Bubble Sort</i> e Produto de matrizes

Tabela 4 – Parâmetros de carga

Parâmetros	Configuração
Tamanho da imagem (JPEG em MegaPixels)	0.3MP, 1MP, 2MP, 4MP e 8MP
Modelo do smartphone	LG Optimus G E977 e Samsung GT-i8190 Galaxy SIII Mini
Tipo da instância	Amazon EC2 Medium e EC2 Micro
Tipo de Rede	<b>Experimento 1:</b> 4G; <b>Experimento 2:</b> 3G e 4G
Tipo de <i>offloading</i>	<b>Experimento 1:</b> estático; <b>Experimento 2:</b> dinâmico
<b>Experimento 2:</b> Dados enviados (kB)	[200, 400, 600, 800,1000] Produto de matrizes [2000, 4000, 6000, 8000] Busca binária e <i>Bubble Sort</i>

*binária, Busca sequencial, Bubblesort e Produto de matrizes*). Tal nicho foi selecionado devido a grande necessidade de *offloading* em procedimentos que exigem muita computação (CORRAL *et al.*, 2014), (SHIRAZ *et al.*, 2015a) e por conta desse tipo de procedimentos também serem utilizados em aplicações para benchmark em MCC (SILVA *et al.*, 2016).

A coleta dos dados foi realizada através da ferramenta Treprn Profiler. Escolhemos essa ferramenta por se tratar de uma solução não-invasiva para coleta de consumo energético em dispositivos móveis e por se tratar de uma solução que possui um vasto alcance no número de aparelhos suportados. Outro aspecto é a possibilidade de realização de estimativas baseadas em consumo energético unicamente de uma determinada aplicação. Diferentemente de soluções em *hardware*, que realizam a captação de dados ligados ao consumo geral do sistema. Portanto, neste trabalho, assumimos que essa ferramenta possui uma estimativa razoável para o estudo aqui realizado.

Tal ferramenta, ainda, disponibiliza um grande número de modelos de dispositivos para realização de medições. Além disso, possui capacidade de extrapolação para realização de medições ligadas ao consumo de I/O e GPU, em uma perspectiva futura.

Para validação do mecanismo proposto, mais dois experimentos foram projetados e executados em contextos de realização de *offloading* distintos, os quais estão descritos nas duas subseções seguintes. Para tal, foram desenvolvidas duas aplicações em Android: (i) executamos a primeira em conjunto com a ferramenta *Treprn Profiler* para obtermos dados referentes ao consumo energético dos algoritmos utilizados neste trabalho. Esses dados foram representados através de um modelo de regressão linear e utilizados nas tomadas de decisão dos Algoritmos 1 e 2; (ii) a segunda aplicação foi desenvolvida em conjunto com as APIs do *framework* MpOS (COSTA *et al.*, 2015) para execução do modelo de tomada de decisão de *offloading* proposto e realização dos experimentos #3 e #4.

Nas subseções a seguir, serão descritos os experimentos relativos à validação do



mecanismo proposto e a comparação do mesmo com execuções de procedimentos puramente locais e com outro procedimento de tomada de decisão de *offloading*.

#### **4.2.1 Experimento #3: Validação do modelo proposto**

O Experimento #3 consistiu da implementação do mecanismo de *offloading* proposto em diferentes ambientes. Essa implementação foi feita no MpOS, um *framework* real de MCC (COSTA *et al.*, 2015) e também realizada em uma ferramenta de simulação proposta por (SILVA Jr. *et al.*, 2015).

No MpOS realizamos a especialização de classes que detinham métodos para a tomada de decisão de *offloading*. No ambiente simulado, a implementação ocorreu de forma análoga, sendo apenas os métodos de tomada de decisão sobrescritos para reaproveitamento da arquitetura já disponível. A validação cruzada obedeceu a parametrização descrita nas Tabelas 5 e 6. A tomada de decisão de *offloading* nos diferentes ambientes foi a métrica analisada neste experimento.

#### **4.2.2 Experimento #4: Comparação entre modelos de offloading**

O Experimentos #4 comparou tomadas de decisão de *offloading* considerando três vertentes para análise: (i) execução de procedimentos somente em ambiente local; (ii) execução de procedimentos utilizando o mecanismo proposto, e; (iii) execução de procedimentos considerando a proposta de (KUMAR *et al.*, 2013).

A Figura 9 resume os passos necessários para a realização do Experimento #4. Primeiramente, foram considerados apenas os algoritmos que tiveram um tempo de execução acima de 100 milissegundos para o experimento em questão. Essa escolha deu-se pelo fato do *software* de medição de energia conseguir realizar leituras, sem ocorrência de gargalos ou lentidão, em intervalos mínimos de 100 milissegundos. Caso o intervalo da medida fosse menor, a medição *per se* poderia interferir no desempenho do aplicativo desenvolvido para realização do experimento.

Para a execução do procedimento em si, uma outra aplicação em Android foi desenvolvida para executar e permitir a escolha dos algoritmos *Bubble Sort* e Produto de matrizes de acordo com os seguintes ambientes de execução: (i) local; (ii) *Offloading* com mecanismo proposto (iii) *Offloading* com proposta de (KUMAR *et al.*, 2013). Esse algoritmos foram escolhidos devido ao seu tempo de execução ser muito maior do que os algoritmos Busca sequencial e

Tabela 5 – Critérios para o projeto dos experimentos

Itens	Descrição
Métricas	<b>Experimento 3:</b> decisão de <i>offloading</i> (dispositivo, WiFi ou 3G); <b>Experimento 2:</b> decisão binária de <i>offloading</i>
Parâmetros	<b>Experimento 3 e 4:</b> procedimento suscetível ao <i>offloading</i>
Fatores	<b>Experimento 3 e 4:</b> complexidade algorítmica, tamanho (em bytes) da entrada do procedimento, tamanho (em bytes) da saída do procedimento
Técnica de análise	<b>Experimento 3:</b> Simulação e Execução de procedimentos em <i>framework</i> real <b>Experimento 4:</b> Execução de procedimentos em <i>framework</i> real
Carga de Trabalho	<b>Experimento 3 e 4:</b> <i>Busca binária, Busca sequencial, Bubblesort e Produto de matrizes</i>

Tabela 6 – Parâmetros de carga

Parâmetros	Configuração
<b>Experimento 1 e 2:</b> Ambientes de execução dos algoritmos	Remota: EC2 Micro (1 VCPUs, 1 2 ECUs, 1 GB de memória) Local: 1 CPU física Core i5 + 6GB de memória Nexus 5: Quad Core + 2GB de memória
Tipo de Rede	<b>Experimento 3 e 4:</b> 3G e WiFi
<b>Experimento 3 e 4:</b> Quantidade de Comunicação	[200 até 1.000.000] (em Bytes) <i>Produto de matrizes</i> [200 até 10.000.000](em Bytes) <i>Busca binária Busca sequencial e Bubblesort</i>

Busca binária, considerando os valores dos mesmos valores de entrada do algoritmo *Bubble Sort*. Outro fator que foi levado em consideração foi a demanda maior de recursos computacionais e energético que os algoritmos *Bubble Sort* e Produto de matrizes obtiveram nas medições realizadas de acordo com as Figuras 4 e 5.

O aplicativo foi implementado com uma dependência do *framework* MpOS devido à necessidade de sua utilização para a análise de *offloading* e, em caso positivo, execução do procedimento remotamente.

A utilização da aplicação deu-se com três cenários. O primeiro, de execução puramente local. O segundo decidindo entre execução local ou remota. A terceira abordagem, com decisão entre execução local ou em uma *cloudlet*. Após a execução dos cenários citados, a coleta de resultados de ambos os experimentos e sua posterior análise foram realizadas.

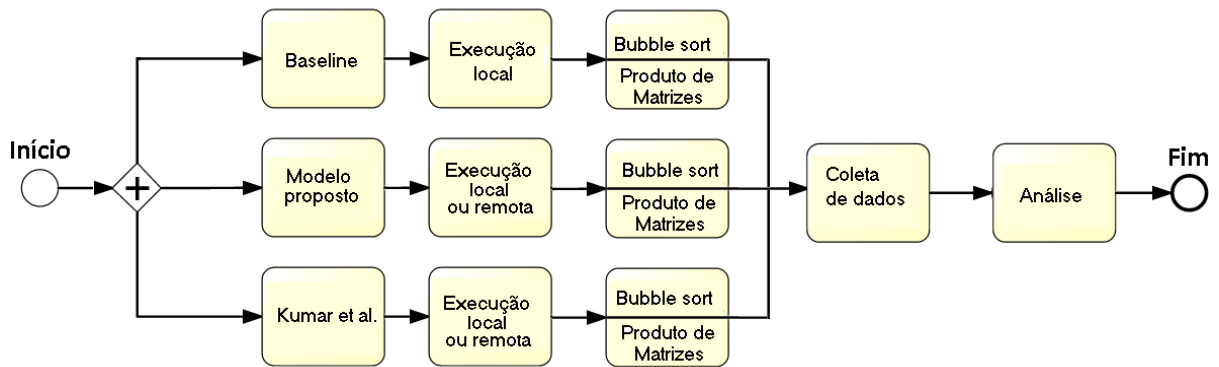


Figura 9 – Diagrama ilustrativo dos passos para execução do Experimento #4.

### 4.3 Resumo do capítulo

Este capítulo apresentou o delineamento dos experimentos realizados durante a elaboração deste trabalho. Quatro experimentos foram projetados com o intuito de validar a proposta descrita no Capítulo 3. Dentro dos experimentos em questão, existem dois grupos distintos. O primeiro grupo detalhou os experimentos para validação da modelagem em MCC. Após isso, no segundo grupo, os experimentos compatíveis com a proposta de mecanismo de *offloading* foram realizados e retroalimentados pelos dados dos experimentos anteriores.

## 5 RESULTADOS E DISCUSSÕES

### 5.1 Resultados dos experimentos #1 e #2

Os resultados do Experimento #1 são apresentados nas Figuras 10 e 11. A Figura 10 ilustra o tempo total para execução de uma tarefa de processamento de imagem em uma instância do tipo *Medium* da Amazon no ambiente real e simulado. O valor de tempo exibido na Figura 10 simboliza, conforme a Equação 3.3, a soma dos tempos de envio, processamento e recebimento de resultados da tarefa de aplicação de filtro de imagem realizada. A Figura 11 expõe a mesma métrica de tempo de execução, baseando-se também na Equação 3.3, porém com a instância do tipo *Micro* da Amazon.

Em ambas as situações ilustradas pelas Figuras 10 e 11, os dados captados e relativos ao cálculo do tempo total de execução foram obtidos em um intervalo de confiança de 95%.

Os resultados do experimento possuem uma acurácia de até 98.9% em comparação com valores reais obtidos em (COSTA *et al.*, 2014). No entanto, observamos que para a imagem de 8 Megapixels, existe uma diferença de até 84% no tempo total observado e simulado. O

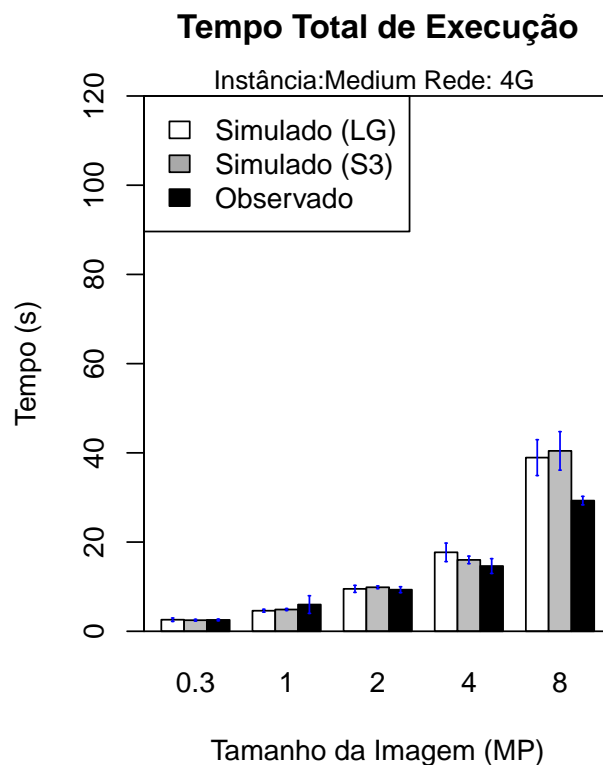


Figura 10 – Experimento #1: Tempos totais de execução (ambiente real vs. simulado), rede 4G, com diferentes tamanhos de imagem e *smartphone* (Instância Medium).

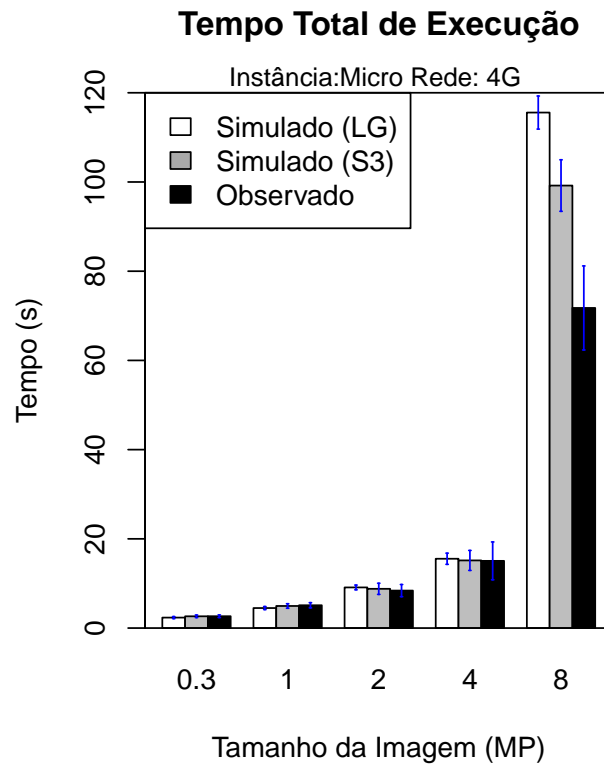


Figura 11 – Experimento #1: Tempos totais de execução (ambiente real vs. simulado), banda 4G, com diferentes tamanhos de imagem e *smartphone* (Instância Micro).

mesmo comportamento é encontrado para a instância Micro (Figura 11). Tal discrepância é explicada através da Equação 3.2, onde o tempo total de execução é calculado com base no tempo de *download* e *upload* das imagens. Considerando que, imagens maiores podem sofrer uma distorção maior no tempo de *download* e *upload* em relação a imagens menores, a possível perda de precisão no cálculo será maior. Outro aspecto a ser considerado é a compressão de diferentes formatos de imagem. Taxas de transmissão de dados podem sofrer influência considerando esse aspecto e, portanto, produzindo resultados ligeiramente diferentes do observado em *frameworks* de MCC reais.

Outro aspecto observado foi que o tempo total de execução para imagens de 8 Megapixels é bastante diferente entre os dois casos. Isso ocorre devido ao fato da capacidade de processamento da instância *Medium* ser maior que a da instância *Micro*. Essa diferença traduz-se em uma simulação que considera maior tempo de processamento para realização da tarefa em imagens maiores.

O Experimento #2 apresenta a tomada de decisão do *offloading* para três algoritmos clássicos (Figuras 12, 13, 14 e 15), de implementações triviais e com ordens de complexidade

distintas. O algoritmo Busca binária  $O(\log n)$  pode ser utilizado em programas de buscas de dados (e.g. aplicativo de armazenamento de contatos). O algoritmo *Bubble Sort*  $O(n^2)$  pode ser utilizado em programas que realizam ordenação de dados, como gerenciadores de arquivos em dispositivos móveis. Já o algoritmo produto de matrizes  $O(n^3)$  pode ser utilizado em *softwares* de processamento de imagens e sinais de uma maneira geral.

Nas Figuras 12 e 13 observamos que, dada a mesma quantidade de comunicação, o algoritmo *Bubble Sort* utiliza-se de um pouco mais de instruções na sua execução do que o algoritmo de Busca binária. Também notamos que é necessária um menor *input* para promover a execução do algoritmo *Bubble Sort* na nuvem do que no algoritmo de Busca binária. Portanto, observamos que a complexidade dos algoritmos impacta na tomada de decisão de *offloading* entre esses dois procedimentos. Tal observação é confirmada pelas Figuras 14 e 15, nas quais são apresentadas a decisão de *offloading* do algoritmo de Produto de matrizes para as bandas 3G e 4G. Tais figuras mostram que a tomada de decisão em favor do *offloading* não ocorreu em apenas um caso (*vide* Figura 14 , Matrix - local) , mesmo com uma variação considerável da carga.

As Figuras 12, 13 diferem em escala das Figuras 14 e 15. O modelo de tomada de decisão mostrou que o algoritmo de Produto de matrizes é bem mais suscetível a sofrer migração para execução em nuvem. Realizando uma comparação entre esses resultados e aqueles apresentados pelos algoritmos Busca binária e *Bubble sort*, observamos que para uma quantidade de processamento maior, o algoritmo de *offloading* opta pela execução remota com pouca influência da banda da rede 3G ou 4G.

Baseados nesses resultados, inferimos que algoritmos de maior ordem complexidade (e.g. *Produto de matrizes*) são mais suscetíveis ao *offloading* quando comparados a algoritmos de menor grau complexidade (e.g. Busca binária). Podemos ainda, afirmar que o modelo produziu resultados esperados com a descrição e variação de fatores encontrados na Equação 3.2.

## 5.2 Resultados dos experimentos #3 e #4

Os resultados do Experimento #3 mostraram que a implementação do modelo proposto tanto no simulador (SILVA Jr. *et al.*, 2015) quanto no *framework* Mpos (COSTA *et al.*, 2015) obtiveram resultados idênticos no que diz respeito a tomada de decisão de *offloading* para diferentes tamanhos da entrada, conforme apresentado nas Tabelas 7 e 8. As duas implementações foram realizadas considerando o consumo energético de procedimentos em ambiente local e transmissão de informações via 3G ou WiFi, quando necessário.

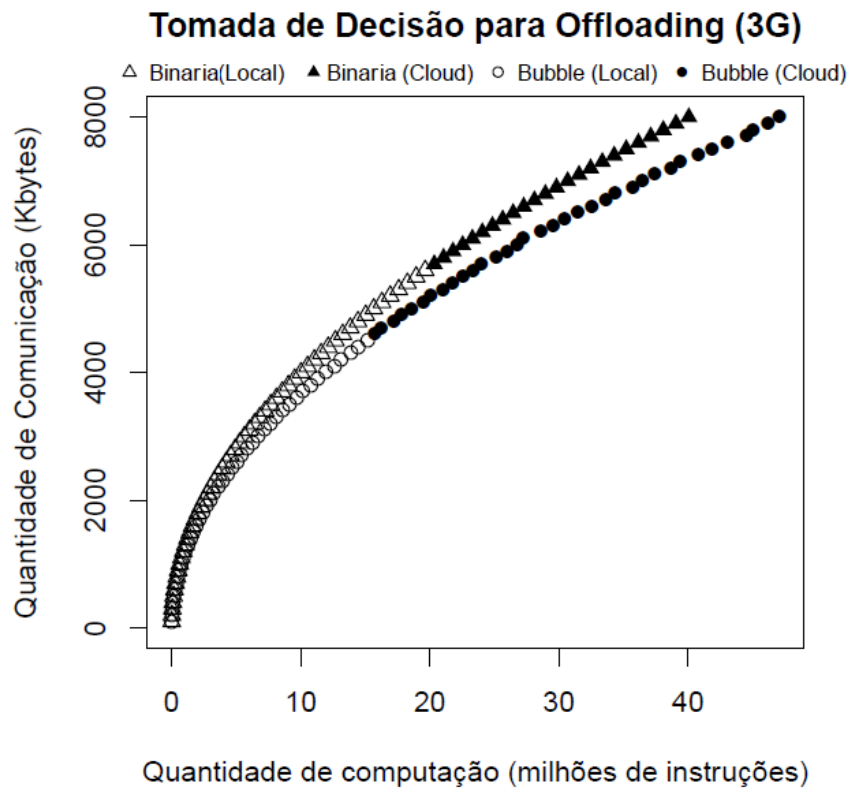


Figura 12 – Experimento #2: Decisão de *offloading* levando-se em conta os algoritmos Busca binária e *Bubble Sort* em rede 3G.

Tabela 7 – Tomada de decisão de offloading: Produto de matrizes

Tamanho da Entrada - 2 dimensões	100	200	400	600	800	1000
Simulador	não	sim	sim	sim	sim	sim
Framework	não	sim	sim	sim	sim	sim

Tabela 8 – Tomada de decisão de offloading: Bubblesort

Tamanho da entrada - 1 dimensão	500	1000	5000	10000	50000	100000
Simulador	não	não	sim	sim	sim	sim
Framework	não	não	sim	sim	sim	sim

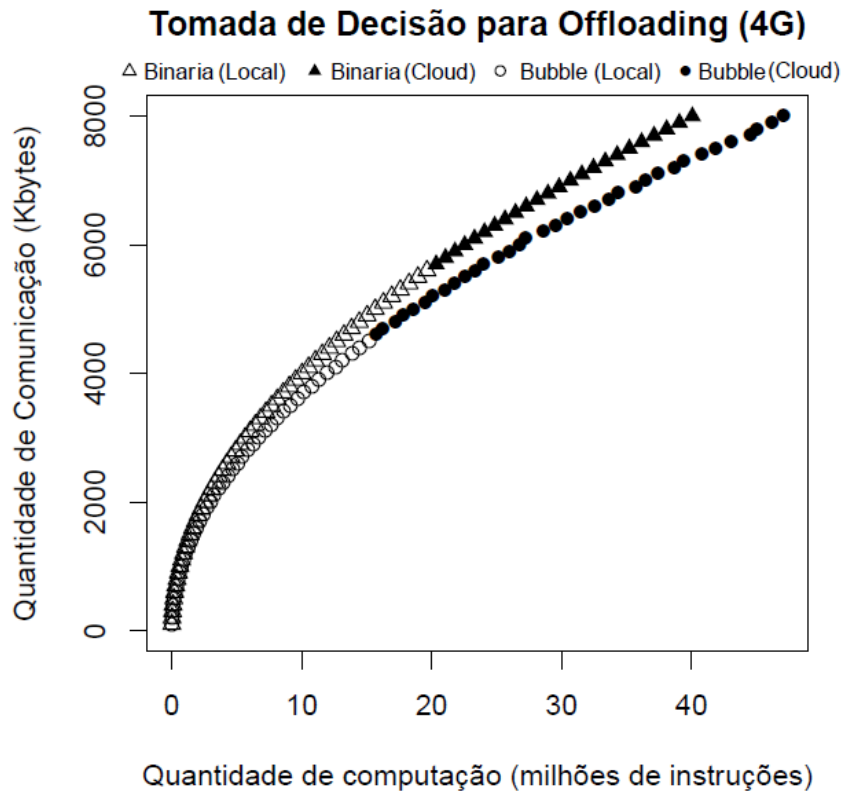


Figura 13 – Experimento #2: Decisão de *offloading* levando-se em conta os algoritmos Busca binária e *Bubble Sort* em rede 4G.

Os resultados do Experimento 3# mostraram uma hegemonia de *offloading* através da WiFi. Isso ocorre devido a estimativa do consumo de energia calculada pelo modelo da transmissão de dados via WiFi ser sempre menor do que o consumo estimado pelo modelo via 3G. Portanto, em um ambiente com disponibilidade de ambas as conexões, o *offloading* via WiFi seria requisitado primariamente. Isso é esperado, pois a transmissão via WiFi consome menos energia do que uma transmissão via rede 3G (SEGATA *et al.*, 2014).

Conforme ilustrado na Figura 4, os algoritmos busca sequencial e busca binária não consumiram uma quantidade de energia considerável nas análises realizadas. Portanto, a decisão de execução local foi unânime em ambos os ambientes onde o mecanismo foi implementado. Desse modo, os resultados obtivos no Experimento #1 mostraram que a simulação do mecanismo proposto está conseguindo representar o comportamento encontrado no ambiente real de maneira satisfatória.

No Experimento #4, medições foram realizadas com o intuito de verificar a eficiência quanto ao consumo energético do mecanismo proposto. Utilizamos um intervalo de confiança das medições de 95% na execução do experimento. Nos resultados em questão podemos destacar



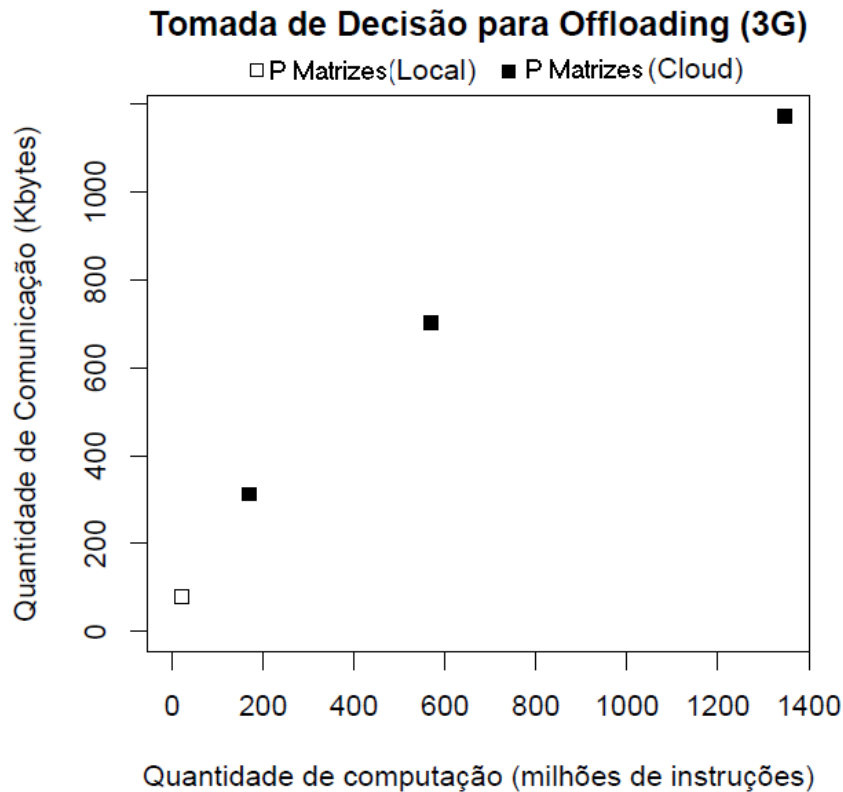


Figura 14 – Experimento #2: Decisão de *offloading* levando-se em conta o algoritmo Produto de matrizes em rede 3G.

que dos três cenários avaliados de *offloading* (execução local, WiFi e 3G), o mecanismo proposto e a proposta de (KUMAR *et al.*, 2013) obtiveram resultados próximos, com diferenças que oscilam entre 2% até 10% a depender da banda disponível e do tipo de algoritmo para realização do *offloading*.

O resultado da execução local, como era esperado, foi o que deteve o maior percentual de consumo energético. Diferentemente dos cenários com possibilidade de *offloading*, onde notamos melhorias de até 12%, conforme ilustrado nas Figuras 17 e 18 comparadas à Figura 16.

No cenário com rede WiFi disponível, mostrado na Figura 17, o modelo proposto resultou em uma parte (entre 20% a 35%, vide Figura 19) dos procedimentos sendo executada localmente. Isso ocorre devido ao modelo não considerar a largura de banda da rede na tomada de decisão de *offloading*. Consequentemente, isso ocasionou uma piora, em termos de consumo energético, variando entre 2% e 3% em relação à proposta de (KUMAR *et al.*, 2013), que resultou apenas em execuções remotas. Tal diferença se deve à decisão de *offloading* de (KUMAR *et al.*, 2013) levar em consideração a vazão de dados entre o dispositivo móvel e o ambiente remoto. Em um ambiente WiFi, com uma alta disponibilidade de largura de banda, o *offloading* torna-se

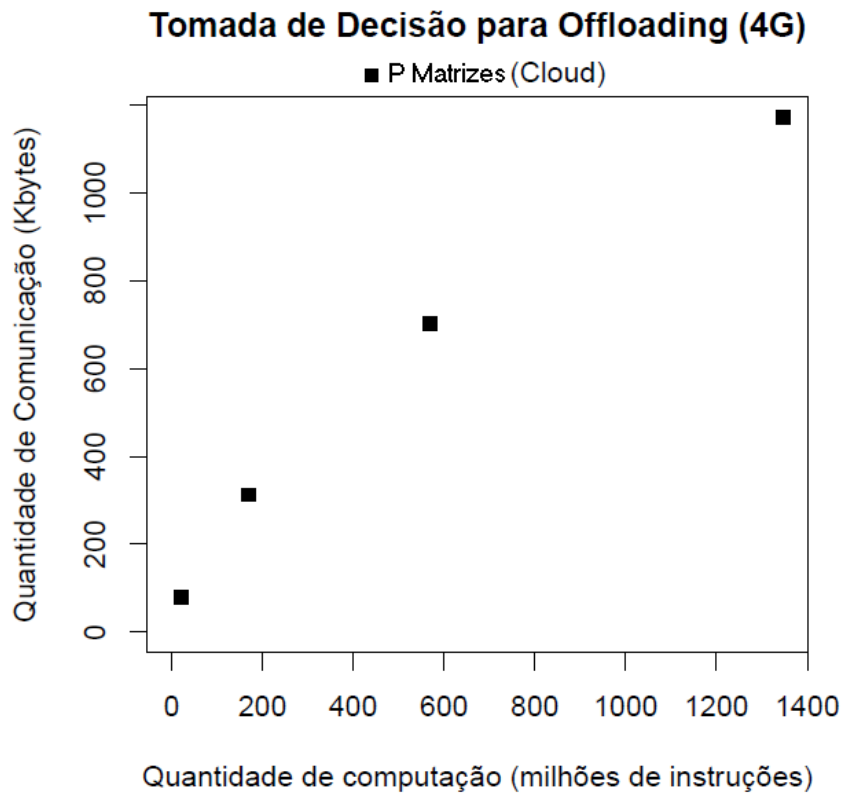


Figura 15 – Experimento #2: Decisão de *offloading* levando-se em conta o algoritmo Produto de matrizes em rede 4G.

mais vantajoso do ponto de vista energético.

Em contra partida, no cenário de *offloading* via 3G, com taxa de transmissão e recepção apresentando maior variação, o modelo proposto resultou em menos execuções locais quando comparado ao mesmo cenário com *offloading* via WiFi (*vide* Figura 20). Diferentemente, o modelo de (KUMAR *et al.*, 2013) possuiu um maior número de execuções locais. Desse modo, percebemos que as execuções locais executadas pelo modelo de (KUMAR *et al.*, 2013) ou pela ausência de mecanismo de *offloading* demandaram um maior consumo energético, conforme ilustrado na Figura 16.

Os resultados mostraram diferenças notáveis no que diz respeito à tecnologia utilizada (3G e WiFi) para tomada de decisão do *offloading*. Nos cenários em que a rede WiFi estava disponível, notamos que nosso mecanismo realizou um número maior de execuções locais entre 20% a 60% a mais do que a tomada de decisão de (KUMAR *et al.*, 2013), como visto na Figura 19. Essa diferença resultou em um consumo energético levemente maior (entre 3% a 6%, *vide* Figura 17) por parte da nossa proposta. Já em ambiente 3G, nosso mecanismo optou por menos execuções locais, diferença de até 60% e, conseqüentemente, obteve uma redução de consumo energético de 6% a 8%, *vide* Figura 18.

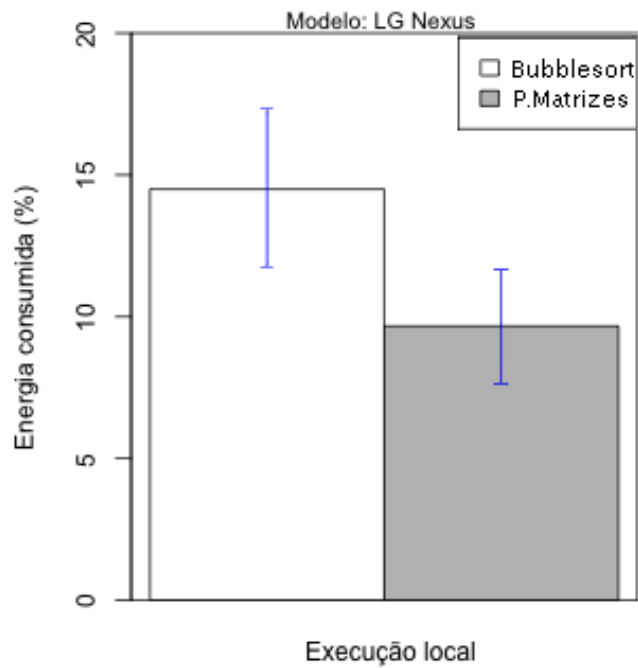


Figura 16 – Energia consumida em 30 minutos de execução - Ambiente Local

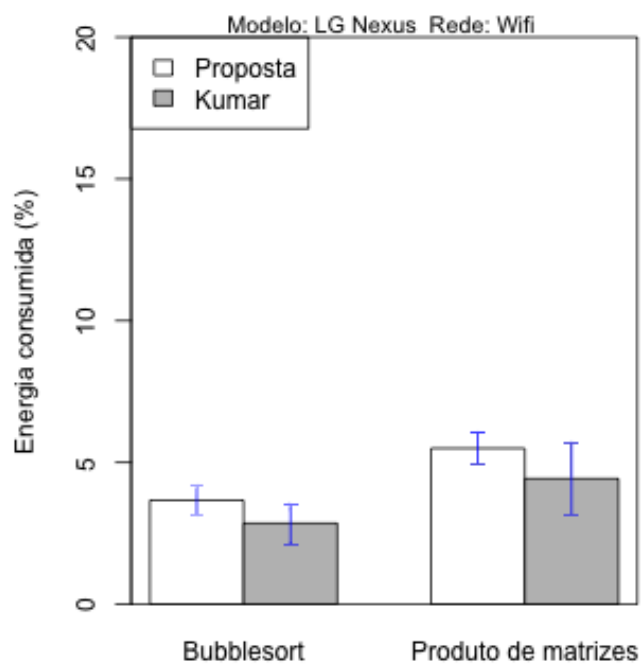


Figura 17 – Energia consumida em 30 minutos de execução - Rede WiFi.

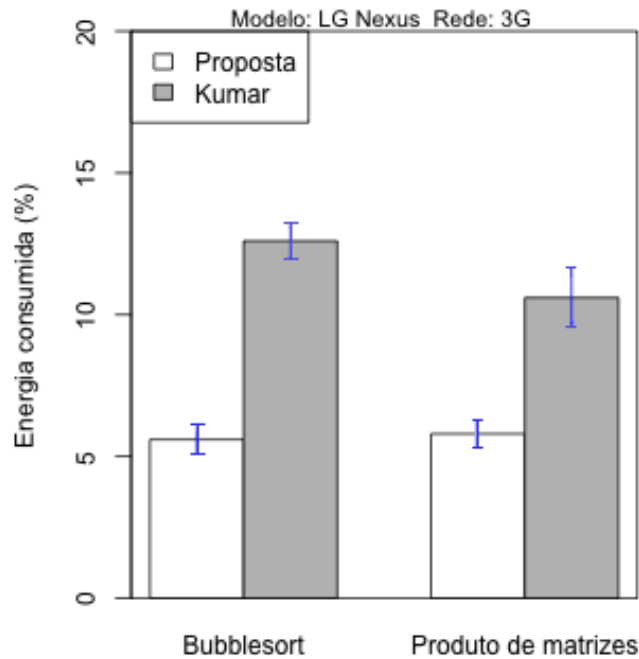


Figura 18 – Energia consumida em 30 minutos de execução - Rede 3G.

Portanto, a proposta deste trabalho é mais adequada, em termos de consumo energético, a ambientes com uma conectividade mais restrita (3G). Enquanto que o modelo proposto por (KUMAR *et al.*, 2013) é mais aderente a ambientes que dispõem de *cloudlets* e uma infraestrutura em rede local robusta para execuções de procedimentos em MCC.

A partir dos resultados obtidos, o mecanismo proposto mostrou-se acurado e possível de ser utilizado em ambientes de simulação e de uso real de MCC. O modelo de *offloading* proposto apresentou melhoras significativas no consumo energético de procedimentos, em ambientes de conectividade 3G, quando comparado a uma execução puramente local ou mesmo com a tomada de decisão de *offloading* de (KUMAR *et al.*, 2013).

Também podemos destacar que nos Experimentos #1 e #2, no contexto de simulação, foram simulados dois tipos de *offloading*, estático e dinâmico. A partir dos resultados obtidos, observamos que o ambiente de simulação foi capaz de representar cenários reais de *offloading* estático com uma acurácia de até 98.9%. Com relação ao *offloading* dinâmico, foram realizadas tomadas de decisão baseadas no custo computacional e na quantidade de dados a serem transmitidos, a partir das quais concluímos que a ordem de complexidade algorítmica da carga é diretamente proporcional à probabilidade de *offloading*.

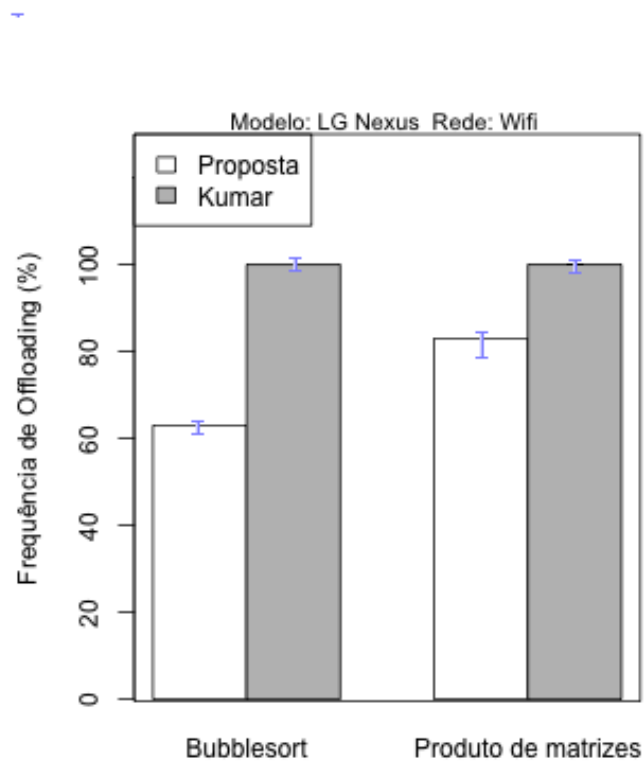


Figura 19 – Frequência - Tomada de decisão de *offloading* em ambiente WiFi.

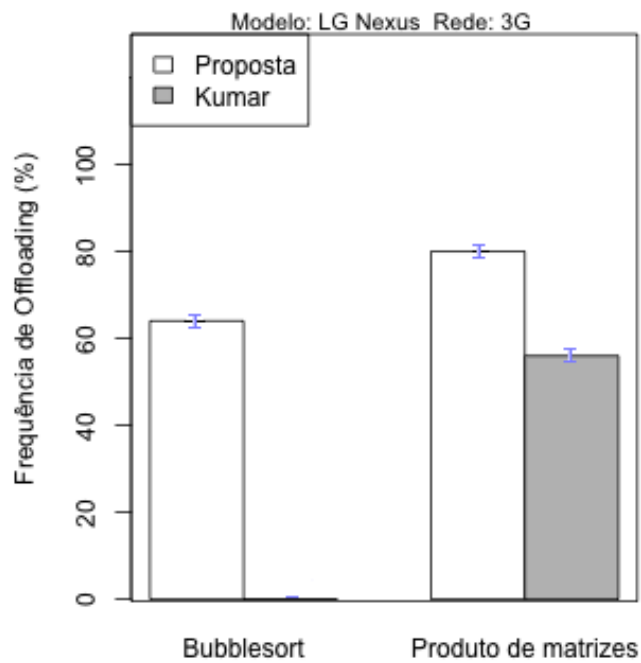


Figura 20 – Frequência - Tomada de decisão de *offloading* em ambiente 3G.

### 5.3 Resumo do capítulo

Neste capítulo, mostramos os resultados obtidos com a execução dos experimentos propostos. No experimento #1, observamos que a modelagem proposta para representação de entidades ligadas ao contexto MCC mostrou-se satisfatória ao representar um cenário real e descrito em (COSTA *et al.*, 2014). Ainda, no Experimento #1, destacamos que devido a natureza das imagens maiores serem suscetíveis a uma maior distorção, com relação ao tempo de transmissão em rede, os resultados com imagens de até 8MP distoaram dos encontrados em ambiente real. No Experimento #2, implementamos a estratégia de (KUMAR *et al.*, 2013) de tomada de decisão de *offloading* para simulação de cenários de *offloading* estático e dinâmico (com e sem tomada de decisão). Neste experimento estabelecemos uma relação entre complexidade algorítmica e *offloading*. Nos resultados do Experimento #3, observamos que a implementação do mecanismo proposto e sua implementação em um *framework* real produziram resultados bastante próximos em tomadas de decisão de *offloading*. Por fim, no Experimento #4, verificamos o mecanismo proposto neste trabalho, para realização de execução remota de procedimentos, é mais indicado de ser utilizado em cenários com conectividade via 3G.

## 6 CONCLUSÃO

Nesta dissertação, propusemos um mecanismo para tomada de decisão de *offloading* que leva em conta a complexidade algorítmica dos procedimentos executados em dispositivos móveis e os respectivos consumos energéticos. O mecanismo proposto conta com quatro entidades principais: dispositivo móvel, instância de nuvem computacional remota, largura de banda de rede sem fio entre o dispositivo móvel e a nuvem e um algoritmo para a tomada de decisão de *offloading*.

Através da implementação do mecanismo proposto no *framework* MpOS (COSTA *et al.*, 2014) (vide Seção 4.1), concluímos (i) que seria possível representar de maneira satisfatória um cenário real de MCC com conexão direta entre o dispositivo móvel e uma nuvem computacional (vide Seção 5.1) e (ii) que há uma associação direta entre o consumo energético do dispositivo móvel e a complexidade algorítmica do procedimento em execução.

Para procedimentos com complexidade inferior a  $O(n^2)$  e com  $n \leq 400$  kBytes, o tempo total de execução no dispositivo móvel foi de 10 a 100 ms (vide Seção 3.1). Neste caso, notamos que o gasto de energia foi ínfimo e, portanto, constatamos que não vale a pena realizar o *offloading*. Em contrapartida, para procedimentos cuja complexidade é igual ou maior a  $O(n^2)$  e com  $n > 20$  kB, o *offloading* do mecanismo proposto reduziu o consumo energético em até 8% com relação ao de Kumar (KUMAR *et al.*, 2013). Em comparação com a execução puramente local de procedimentos, conseguimos reduzir o consumo energético em até 14% (vide Seção 5.2).

A principal contribuição deste trabalho de mestrado é um mecanismo extensível para tomada de decisão de *offloading* em dispositivos móveis. O algoritmo de decisão proposto é capaz de definir o local mais adequado para a execução do referido procedimento, se no próprio dispositivo, em uma *cloudlet* ou em uma nuvem computacional remota. Nossa proposta reduz o consumo energético de dispositivos móveis cuja interface de rede sem fio tenha banda inferior ao padrão IEEE 802.11 b/g/n (2.4 GHz).

Na qualidade de contribuição secundária, desenvolvemos uma ferramenta para modelagem e simulação de arquiteturas de MCC (SILVA Jr. *et al.*, 2015), cujo código-fonte encontra-se disponível em repositório público<sup>1</sup>.

<sup>1</sup> Disponível em: <https://github.com/UFC-GREAt-PPGETI/MCCSimulator>

## 6.1 Estudos em Perspectiva

Como sequência deste trabalho, podemos sugerir (i) detecção automática da complexidade algorítmica dos procedimentos passíveis de *offloading*, (ii) a modelagem de aspectos de mobilidade na tomada de decisão, (iii) a inclusão de mecanismos de tolerância a falhas, (iv) inclusão de comparativo com outras interfaces de rede (e.g. LTE 4G, *bluetooth*), (v) inclusão do custo monetário relativo aos planos de dados de redes celulares, (vi) integração de dispositivos móveis com nós sensores sem fio e objetos inteligentes em uma perspectiva de integração Cloud-IoT (BOTTA *et al.*, 2016), (vii) teste do modelo proposto com outros níveis de granularidade de código (e.g. *threads*, objetos, classes, módulos) (ZHANG *et al.*, 2016), (viii) realização de análise e medições de consumo energético dos algoritmos mediante *hardware* específico para tal (e.g. Power Monitor<sup>2</sup>).

Sugerimos ainda a incorporação de novas entidades ao modelo, tais como as *cloudlets*, juntamente com novos cenários MCC (e.g. computação em névoa, nuvem das coisas). Uma outra perspectiva em vista, a qual particularmente julgamos bastante interessante, seria trabalhar a recomendação de *offloading* dinâmico a partir do consumo energético no próprio sistema operacional do dispositivo móvel (COUTO *et al.*, 2014; MAGALHAES *et al.*, 2015; ALI *et al.*, 2016).

---

<sup>2</sup> Especificações disponíveis em: <https://www.msoon.com/LabEquipment/PowerMonitor/>



## REFERÊNCIAS

ABOLFAZLI, S.; SANAEI, Z.; AHMED, E.; GANI, A.; BUYYA, R. Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges. **CoRR**, abs/1306.4956, 2013. Disponível em: <<http://dblp.uni-trier.de/db/journals/corr/corr1306.html#AbolfazliSAGB13>>.

ABOLFAZLI, S.; SANAEI, Z.; GANI, A. Mobile cloud computing: A review on smartphone augmentation approaches. **CoRR**, abs/1205.0451, 2012. Disponível em: <<http://arxiv.org/abs/1205.0451>>.

AHMED A. E SABYASACHI, A. Cloud computing simulators: A detailed survey and future direction. In: **Advance Computing Conference (IACC), 2014 IEEE International**. [S.l.: s.n.], 2014. p. 866–872.

ALI, F. A.; SIMOENS, P.; VERBELEN, T.; DEMEESTER, P.; DHOEDT, B. Mobile device power models for energy efficient dynamic offloading at runtime. **Journal of Systems and Software**, v. 113, p. 173 – 187, 2016. ISSN 0164-1212. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0164121215002666>>.

AMORETTI, M.; AGOSTI, M.; ZANICHELLI, F. Deus: a discrete event universal simulator. In: DALLE, O.; WAINER, G. A.; PERRONE, L. F.; STEA, G. (Ed.). **SimuTools**. ICST/ACM, 2009. p. 58. ISBN 978-963-9799-45-5. Disponível em: <<http://dblp.uni-trier.de/db/conf/simutools/simutools2009.html#AmorettiAZ09>>.

AMORETTI, M.; GRAZIOLI, A.; ZANICHELLI, F. A modeling and simulation framework for mobile cloud computing. **Simulation Modelling Practice and Theory**, v. 58, Part 2, p. 140 – 156, 2015. ISSN 1569-190X. Special issue on Cloud Simulation. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1569190X15000799>>.

BOTTA, A.; DONATO, W. de; PERSICO, V.; PESCAPÉ, A. Integration of cloud computing and internet of things. **Future Gener. Comput. Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 56, n. C, p. 684–700, mar. 2016. ISSN 0167-739X. Disponível em: <<http://dx.doi.org/10.1016/j.future.2015.09.021>>.

CALHEIROS, R. N.; RANJAN, R.; BELOGLAZOV, A.; ROSE, C. A. F. D.; BUYYA, R. Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. **Softw. Pract. Exper.**, John Wiley & Sons, Inc., New York, NY, USA, v. 41, n. 1, p. 23–50, jan. 2011. ISSN 0038-0644. Disponível em: <<http://dx.doi.org/10.1002/spe.995>>.

CARVALHO, C.; GOMES, D. G.; AGOULMINE, N.; SOUZA, J. N. de. Improving prediction accuracy for wsn data reduction by applying multivariate spatio-temporal correlation. **Sensors**, v. 11, n. 11, p. 10010, 2011. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/11/11/10010>>.

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**. New York, NY, USA: ACM, 2011. (EuroSys '11), p. 301–314. ISBN 978-1-4503-0634-8. Disponível em: <<http://doi.acm.org/10.1145/1966445.1966473>>.

CORRAL, L.; GEORGIEV, A. B.; SILLITTI, A.; SUCCI, G. A study of energy-aware implementation techniques: Redistribution of computational jobs in mobile apps. **Sustainable**

**Computing: Informatics and Systems**, p. –, 2014. ISSN 2210-5379. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2210537914000912>>.

COSTA, P. B.; REGO, P. A. L.; COUTINHO, E. F.; TRINTA, F. A. M.; SOUZA, J. N. de. Uma análise do impacto da qualidade da internet móvel na utilização de cloudlets. In: **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2014)**. Florianópolis: SBC, 2014. p. 223–236.

COSTA, P. B.; REGO, P. A. L.; ROCHA, L. S.; TRINTA, F. A. M.; SOUZA, J. N. de. Mpos: A multiplatform offloading system. In: **Proceedings of the 30th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2015. (SAC '15), p. 577–584. ISBN 978-1-4503-3196-8. Disponível em: <<http://doi.acm.org/10.1145/2695664.2695945>>.

COUTO, M.; CARÇÃO, T.; CUNHA, J.; FERNANDES, J. P.; SARAIVA, J. Detecting anomalous energy consumption in android applications. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 8771 LNCS, p. 77–91, 2014.

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**. New York, NY, USA: ACM, 2010. (MobiSys '10), p. 49–62. ISBN 978-1-60558-985-5. Disponível em: <<http://doi.acm.org/10.1145/1814433.1814441>>.

DINH, H. T.; LEE, C.; NIYATO, D.; WANG, P. A survey of mobile cloud computing: architecture, applications, and approaches. **Wireless Communications and Mobile Computing**, 2013. Disponível em: <<http://dx.doi.org/10.1002/wcm.1203>>.

FERNANDO, N.; LOKE, S. W.; RAHAYU, W. Mobile cloud computing: A survey. **Future Gener. Comput. Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 29, n. 1, p. 84–106, jan. 2013. ISSN 0167-739X. Disponível em: <<http://dx.doi.org/10.1016/j.future.2012.05.023>>.

FLORES, H.; SRIRAMA, S. Adaptive code offloading for mobile cloud applications: Exploiting fuzzy sets and evidence-based learning. In: **Proceeding of the Fourth ACM Workshop on Mobile Cloud Computing and Services**. New York, NY, USA: ACM, 2013. (MCS '13), p. 9–16. ISBN 978-1-4503-2072-6. Disponível em: <<http://doi.acm.org/10.1145/2482981.2482984>>.

GOMES, D. G.; TOLOSANA-CALASANZ, R.; AGOULMINE, N. Introduction to special issue on green mobile cloud computing (green mcc). **Sustainable Computing: Informatics and Systems**, v. 8, p. 37 –, 2015. ISSN 2210-5379. Special Issue on Computing for a Greener Water/Energy/Emissions Nexus; edited by Carol J. Miller and Special Issue on Green Mobile Cloud Computing (Green MCC); edited by Danielo G. Gomes, Rafael Tolosana-Calasanz, and Nazim Agoulmine. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2210537915000529>>.

JUSTINO, T.; BUYYA, R. Outsourcing resource-intensive tasks from mobile apps to clouds: Android and aneka integration. In: **Cloud Computing in Emerging Markets (CCEM), 2014 IEEE International Conference on**. [S.l.: s.n.], 2014. p. 1–8.

KHAN, A. R.; OTHMAN, M.; MADANI, S. A.; KHAN, S. U. A survey of mobile cloud computing application models. **Communications Surveys & Tutorials, IEEE**, iee, v. 16, n. 1, p. 393–413, 2014.

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: GREENBERG, A. G.; SOHRABY, K. (Ed.). **INFOCOM**. IEEE, 2012. p. 945–953. ISBN 978-1-4673-0773-4. Disponível em: <<http://dblp.uni-trier.de/db/conf/infocom/infocom2012.html#KostaAHMZ12>>.

KUMAR, K.; LIU, J.; LU, Y.-H.; BHARGAVA, B. A survey of computation offloading for mobile systems. **Mob. Netw. Appl.**, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 18, n. 1, p. 129–140, fev. 2013. ISSN 1383-469X. Disponível em: <<http://dx.doi.org/10.1007/s11036-012-0368-0>>.

LI, J.; BU, K.; LIU, X.; XIAO, B. Enda: Embracing network inconsistency for dynamic application offloading in mobile cloud computing. In: **Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing**. New York, NY, USA: ACM, 2013. (MCC '13), p. 39–44. ISBN 978-1-4503-2180-8. Disponível em: <<http://doi.acm.org/10.1145/2491266.2491274>>.

LIU, J.; AHMED, E.; SHIRAZ, M.; GANI, A.; BUYYA, R.; QURESHI, A. Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions. **Journal of Network and Computer Applications**, Elsevier, v. 48, p. 99–117, 2015.

MAGALHAES, F. H. S. de; ROCHA, L. S.; GOMES, D. G. Ecodroid: Uma ferramenta para análise e visualização de consumo de energia em aplicativos android. In: **III Workshop de Visualização, Evolução e Manutenção de Software (VEM)**. [S.l.: s.n.], 2015. p. 81–88.

MAGURAWALAGE CHATHURA M., Y. K. H. L. Z. J. S. Energy-efficient and network-aware offloading algorithm for mobile cloud computing. **Computer Networks**, v. 74, n. Part B, p. 22–33, 2014.

MIETTINEN, A. P.; NURMINEN, J. K. Energy efficiency of mobile clients in cloud computing. In: **Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing**. Berkeley, CA, USA: USENIX Association, 2010. (HotCloud'10), p. 4–4. Disponível em: <<http://dl.acm.org/citation.cfm?id=1863103.1863107>>.

MOTA, V. F.; MOURA, H.; SILVA, V. F.; MACEDO, D. F.; GHAMRI-DOUDANE, Y.; NOGUEIRA, u. . h. J. M. S. Analisando a capacidade de descarregamento de redes moveis por meio de redes oportunisticas. In: **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2016)**. Salvador: SBC, 2016.

QI; GANI, A. Research on mobile cloud computing: Review, trend and perspectives. In: **Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on**. [S.l.: s.n.], 2012. p. 195–202.

RAVI, A.; PEDDOJU, S. K. Handoff strategy for improving energy efficiency and cloud service availability for mobile devices. **Wirel. Pers. Commun.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 81, n. 1, p. 101–132, mar. 2015. ISSN 0929-6212. Disponível em: <<http://dx.doi.org/10.1007/s11277-014-2119-y>>.

SATYANARAYANAN, M.; BAHL, P.; CACERES, R.; DAVIES, N. The case for vm-based cloudlets in mobile computing. **IEEE Pervasive Computing**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 8, n. 4, p. 14–23, out. 2009. ISSN 1536-1268. Disponível em: <<http://dx.doi.org/10.1109/MPRV.2009.82>>.

SCHÜRING, M. Mobile cloud computing—open issues and solutions. In: **15th Twente Student Conference on IT, Enschede, The Netherlands**. [S.l.: s.n.], 2011.

SEGATA, M.; BLOESSL, B.; SOMMER, C.; DRESSLER, F. Towards energy efficient smart phone applications: Energy models for offloading tasks into the cloud. In: **Communications (ICC), 2014 IEEE International Conference on**. [S.l.: s.n.], 2014. p. 2394–2399.

SHIRAZ, M.; GANI, A.; SHAMIM, A.; KHAN, S.; AHMAD, R. Energy efficient computational offloading framework for mobile cloud computing. **Journal of Grid Computing**, Springer Netherlands, v. 13, n. 1, p. 1–18, 2015. ISSN 1570-7873. Disponível em: <<http://dx.doi.org/10.1007/s10723-014-9323-6>>.

SHIRAZ, M.; SOOKHAK, M.; GANI, A.; SHAH, S. A. A. A Study on the Critical Analysis of Computational Offloading Frameworks for Mobile Cloud Computing. **Journal of Network and Computer Applications**, v. 47, p. 47–60, 2015.

SILVA, F. A.; ZAICANER, G.; QUESADO, E.; DORNELAS, M.; SILVA, B.; MACIEL, P. Benchmark applications used in mobile cloud computing research: a systematic mapping study. **The Journal of Supercomputing**, Springer US, v. 72, n. 4, p. 1431–1452, 2016.

SILVA Jr., L. S. d.; MAGALHÃES, D. M. V.; GOMES, D. G. Modelagem e simulação de offloading para computação móvel em nuvem. In: **Anais / XXXV Congresso da Sociedade Brasileira de Computação**. Porto Alegre: SBC, 2015. p. 91–100.

SONG, J.; CUI, Y.; LI, M.; QIU, J.; BUYYA, R. Energy-traffic tradeoff cooperative offloading for mobile cloud computing. In: **2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)**. [S.l.: s.n.], 2014. p. 284–289. ISSN 1548-615X.

WU, H.; HUANG, D.; BOUZEFRANE, S. Making offloading decisions resistant to network unavailability for mobile cloud collaboration. In: **Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on**. [S.l.: s.n.], 2013. p. 168–177.

ZHANG, W.; TAN, S.; XIA, F.; CHEN, X.; LI, Z.; LU, Q.; YANG, S. A survey on decision making for task migration in mobile cloud environments. **Personal Ubiquitous Comput.**, Springer-Verlag, London, UK, UK, v. 20, n. 3, p. 295–309, jun. 2016. ISSN 1617-4909. Disponível em: <<http://dx.doi.org/10.1007/s00779-016-0915-y>>.

ZHANG, W.; WEN, Y.; GUAN, K.; KILPER, D. C.; LUO, H.; WU, D. O. Energy-optimal mobile cloud computing under stochastic wireless channel. **IEEE Transactions on Wireless Communications**, v. 12, n. 9, p. 4569–4581, 2013. Disponível em: <<http://dblp.uni-trier.de/db/journals/twc/twc12.html#ZhangWGKLW13>>.