



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada
Mestrado Acadêmico

Gustavo Rostirolla

ElCity: Um Modelo Elástico e Multinível de Economia
de Energia para Cidades Inteligentes

São Leopoldo, 2016

Gustavo Rostirolla

ELCITY: UM MODELO ELÁSTICO E MULTINÍVEL DE ECONOMIA DE
ENERGIA PARA CIDADES INTELIGENTES

Dissertação apresentada como requisito
parcial para a obtenção do título de Mestre
pelo Programa de Pós-Graduação em
Computação Aplicada da Universidade do
Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Rodrigo da Rosa Righi

São Leopoldo
2016

R839e Rostirolla, Gustavo.
Elcity: um modelo elástico e multinível de economia de energia para cidades inteligentes / Gustavo Rostirolla. – 2016.
105 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2016.

"Orientador: Prof. Dr. Rodrigo da Rosa Righi."

1. Consumo de energia. 2. Planejamento urbano. 3. Computação em nuvem. I. Righi, Rodrigo da Rosa. II Título.

CDU 004.7

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecária: Raquel Herbez França – CRB 10/1795)

Gustavo Rostirolla

EiCity:
Um Modelo Elástico e Multinível de Economia
de Energia para Cidades Inteligentes

Dissertação apresentada à Universidade do Vale do
Rio dos Sinos – Unisinos, como requisito parcial para
obtenção do título de Mestre em Computação
Aplicada.

Aprovado em 26 de Agosto de 2016

BANCA EXAMINADORA

Prof. Dr. Rodrigo da Rosa Righi - UNISINOS

Prof. Dr. Cristiano André da Costa - UNISINOS

Prof. Dr. Antônio Marcos Alberti - INATEL

Prof. Dr. Rodrigo da Rosa Righi

Visto e permitida a impressão
São Leopoldo,

Prof. Dr. Sandro José Rigo
Coordenador PPG em Computação Aplicada

*Around here, however, we don't look backwards for very long.
We keep moving forward, opening up new doors and doing new things,
because we're curious... and curiosity keeps leading us down new paths.*
— WALT DISNEY

AGRADECIMENTOS

Gostaria de agradecer minha família, por acreditar e investir em mim. Mãe, sua presença e alegria significaram segurança e certeza de que não estou sozinho nesta caminhada. Pai, seu exemplo de persistência e honestidade foi o que deram, em alguns momentos, força para seguir. Adrielle, obrigado pelos anos de amizade, dedicação, e paciência ao revisar a “cidade inteligente e o mundo das nuvens”, tenho orgulho de ser teu irmão.

Aos colegas Vinícius, Ivam e Eduardo do laboratório de redes e sistemas distribuídos, sempre presentes, propondo novas discussões e apoiando esta pesquisa. Ao meu orientador Prof. Rodrigo da Rosa Righi, por acreditar em mim, por todas as tardes dedicadas a discussão de artigos, e conversas que contribuíram não só para meu crescimento acadêmico, mas também pessoal. Aos demais professores do programa, em especial aos professores Cristiano André da Costa e Jorge Luis Victória Barbosa pelo apoio e contribuições em diversos momentos.

RESUMO

Como resultado da migração rural e suburbana para as cidades, a vida urbana tornou-se um desafio significativo para os cidadãos e para a gestão da cidade, impondo uma enorme preocupação na utilização sustentável de recursos como energia, água, transportes e habitação. Cidades inteligentes são a maior aposta para enfrentar estes desafios de forma eficiente através de um acompanhamento em tempo real que visa auxiliar no planejamento inteligente e desenvolvimento urbano sustentável. Entretanto, para que este acompanhamento ocorra de forma eficiente e possibilite a utilização sustentável dos recursos é necessária uma integração entre cidadãos, dispositivos da cidade, administração pública e a plataforma de computação em nuvem, onde os dados serão armazenados e processados. Para tal, propõe-se o modelo ElCity, um modelo que combina dados de cidadãos e dispositivos da cidade para permitir uma gestão elástica multinível do consumo de energia de uma cidade. Como decisão de projeto, essa gestão deve ocorrer de forma automática, sem afetar a qualidade dos serviços já oferecidos. A principal contribuição do modelo ElCity diz respeito à extensão do conceito de elasticidade em nuvem para os demais níveis (smartphones dos cidadãos, dispositivos da cidade e *data center*), alterando o estado dos componentes em cada nível entre ligado e desligado de acordo com a demanda. Desta forma, este trabalho apresenta o modelo ElCity, detalhando seus módulos distribuídos ao longo das três fontes de dados, além dos experimentos que utilizam dispositivos da cidade e dados de cidadãos de Roma para explorar a economia de energia. Os resultados são promissores, onde o módulo monitor de energia permite estimar o consumo de energia das aplicações elásticas baseado em traços de CPU e memória com uma precisão média e mediana de 97,15% e 97,72%. Além disso, os resultados apontam a possibilidade de redução de mais de 90% no consumo de energia relativo a iluminação pública na cidade de Roma, obtido através da análise de dados de localização de seus cidadãos.

Palavras-chave: Cidades Inteligentes. Consumo de Energia. Planejamento Urbano. Computação em Nuvem. Elasticidade.

ABSTRACT

As a result of rural and suburban migration to the cities, urban life has become a significant challenge for citizens and, particularly, for city administrators who must manage the sustainable use of resources such as energy, water and transportation. Smart cities are the biggest vision to efficiently address these challenges through a real-time monitoring, providing an intelligent planning and a sustainable urban development. However, to accomplish them we need a tightly integration among citizens, city devices, city administrators and the data center platform where all data is stored, combined and processed. In this context, we propose ElCity, a model that combines citizens and city devices data to enable an elastic multi-level management of energy consumption for a particular city. As design decision, this management must occur automatically without affecting the quality of already offered services. The main contribution of ElCity model concerns the exploration of the cloud elasticity concept in multiple target levels (smartphones from citizens, city devices involved in the public lightning and data center nodes), turning on or off the resources on each level in accordance with their demands. In this way, this work presents the ElCity architecture, detailing its modules distributed along the three data sources, in addition to an experiment that uses city devices and citizens data from Rome to explore energy saving. The results are promising, with an Energy Monitor module that allows the estimation of the energy consumption of elastic applications based on CPU and memory traces with an average and median precision of 97.15% and 97.72%. Moreover, the results point to a reduction of more than 90% in the energy spent in public lightning in the city of Rome which was obtained thanks to an analysis of geolocation data from their citizens.

Keywords: Smart Cities. Energy Consumption. Urban Planning. Cloud Computing. Elasticity.

LISTA DE FIGURAS

Figura 1:	Abordagens dos sistemas atuais.	23
Figura 2:	Níveis atingidos pelo modelo ElCity.	25
Figura 3:	Fluxograma das etapas de desenvolvimento da pesquisa.	26
Figura 4:	Problemas de pesquisa em Computação Ubíqua.	28
Figura 5:	Ciclo de um sistema sensível a contexto.	30
Figura 6:	Taxonomia de técnicas para melhoria da eficiência energética em sistemas distribuídos de larga escala.	34
Figura 7:	Consumo de energia dos diversos sensores de localização disponíveis (GPS, WiFi e GSM).	42
Figura 8:	Árvore lógica de mobilidade LMT (<i>logical mobility tree</i>).	43
Figura 9:	Arquitetura do ACE.	44
Figura 10:	Arquitetura do SmartDC.	45
Figura 11:	Arquitetura do Scallop4SC.	47
Figura 12:	Arquitetura propostas por Khan e Kiani (2012).	48
Figura 13:	Arquitetura de alto nível do DC4Cities.	49
Figura 14:	Níveis de integração do sistema de iluminação inteligente.	50
Figura 15:	Arquitetura do Sensing as a Service.	51
Figura 16:	Arquitetura de alto nível do UCC.	52
Figura 17:	Arquitetura ElCity, destacando os três pontos que realizam as ações de economia de energia: (1) gerenciador responsável por ligar e desligar os dispositivos da cidade de acordo com a demanda dos cidadãos; (2) gerenciador responsável por controlar a aplicação de recomendação de economia de energia na nuvem, propondo a adição ou remoção de máquinas virtuais para apoiar a execução; (3) gerenciador responsável pela redução do consumo de energia nos smartphones ligando e desligando os sensores de acordo com a demanda e consumo de energia de cada sensor.	59
Figura 18:	Utilização de rede, CPU e memória principal durante a execução de uma carga composta.	64
Figura 19:	Comparativo do consumo instantâneo entre o consumo previsto e o consumo medido.	64
Figura 20:	Medição de consumo em ambiente de nuvem utilizando o monitor de energia.	65
Figura 21:	Distribuição de nós, VMs e processos usando a infraestrutura de nuvem do AutoElastic, na qual cada VM engloba um único processo da aplicação e cada nó executa c VMs, em que c denota o número de CPUs do nó.	68
Figura 22:	Representação gráfica dos padrões de carga utilizados na avaliação do monitor de energia.	76
Figura 23:	Consumo de energia (em Watts) dos diferentes padrões de carga, sem elasticidade.	77
Figura 24:	Consumo energético para diferentes quantidades de máquinas virtuais e cargas de trabalho variando os thresholds inferior e superior.	78
Figura 25:	Comportamento do consumo energético das diferentes cargas de trabalho variando os thresholds inferior e superior.	79

Figura 26: Mapa da cidade de Roma, com rotas em forma de mapa de calor e com pontos indicando dispositivos da cidade. Em verde os com menor movimentação, azul com movimentação média e vermelho com maior movimentação.	82
Figura 27: Cenários avaliados e situação desejável: cidade inteligente, considerando o consumo do <i>data center</i> e dispositivos da cidade (lâmpadas com uma placa de sensores/atuadores), apresentando um consumo de energia mais baixo quando comparado com o obtido em uma cidade padrão (somente lâmpadas ligando em horários fixos).	83
Figura 28: Perfis de tempo de execução (a) e consumo de energia (b) quando considerada a alocação de VMs e variação dos limites inferiores e superiores.	84
Figura 29: Comportamento da aplicação para recomendação de economia de energia quando executado com elasticidade no ambiente de nuvem, variando os limites superiores e inferiores.	85
Figura 30: Consumo de energia das lâmpadas de vapor de sódio com e sem elasticidade nos dispositivos da cidade.	85
Figura 31: Consumo de energia das lâmpadas de LED com e sem elasticidade nos dispositivos da cidade.	86
Figura 32: Consumo de energia dos dispositivos da cidade com e sem elasticidade.	87

LISTA DE TABELAS

Tabela 1:	Componentes de uma cidade inteligente e aspectos relacionados a vida urbana	36
Tabela 2:	Avaliação dos trabalhos relacionados com base nos quesitos definidos. . .	53
Tabela 3:	Tempo e energia medida através do monitor de energia com e sem a elasticidade, variando os thresholds superior e inferior (quando a elasticidade está ativa).	80
Tabela 4:	Tempo e consumo de energia da execução sequencial da aplicação de recomendação de economia de energia variando o número de cidadãos e dispositivos.	83
Tabela 5:	Tempo de execução e consumo de energia variando os limites inferior e superior da aplicação de recomendação de economia de energia na nuvem com 316 usuários e 100 dispositivos.	83
Tabela 6:	Tempo de execução da aplicação de recomendação de economia de energia na nuvem com os limiares 30/70 e consumo de energia de diferentes dispositivos da cidade (lâmpadas), variando o número de cidadãos e dispositivos. O consumo de energia na cidade padrão é o consumo dos dispositivos cidade sem elasticidade. O consumo de energia na cidade inteligente é obtido através da soma dos dispositivos com elasticidade e da aplicação.	87
Tabela 7:	Contribuições do modelo ElCity comparado com os trabalhos relacionados com base nos quesitos previamente definidos.	91

LISTA DE SIGLAS

ACM	Association for Computing Machinery
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CPU	Central Processing Unit
FAPERGS	Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IEEE	Institute of Electrical and Electronics Engineers
IaaS	Infrastructure as a Service
JMS	Java Message Service
LED	Light Emitting Diode
MPI	Message Passing Interface
MPMD	Multiple program-multiple data
NFS	Network File System
PaaS	Platform as a Service
PCR	Principal Component Regression
RAM	Random Access Memory
RFID	Radio-Frequency IDentification
SSH	Secure Shell
SaaS	Software as a Service
SLA	Service Level Agreement
SQL	Structured Query Language
TI	Tecnologia da Informação
VM	Virtual Machine

SUMÁRIO

1 INTRODUÇÃO	21
1.1 Motivação	22
1.2 Questão de Pesquisa	23
1.3 Objetivos	24
1.4 Etapas de Desenvolvimento da Pesquisa	25
1.5 Organização do Texto	26
2 FUNDAMENTAÇÃO TEÓRICA	27
2.1 Dispositivos Móveis	27
2.1.1 Computação Ubíqua	27
2.1.2 Contexto	29
2.1.3 Consumo de Energia em Dispositivos Móveis	30
2.2 Computação em Nuvem	31
2.2.1 Classificações de Computação em Nuvem	31
2.2.2 Consumo de Energia em Computação em Nuvem	33
2.3 Cidades Inteligentes	35
2.3.1 Classificação de Cidades Inteligentes	35
2.3.2 Planejamento Urbano	36
2.3.3 Consumo de Energia em Cidades Inteligentes	37
2.4 Considerações Parciais	38
3 TRABALHOS RELACIONADOS	41
3.1 Metodologia de Pesquisa e Escolha dos Trabalhos Relacionados	41
3.2 Dispositivos Móveis	41
3.2.1 EnLoc	42
3.2.2 ACE	43
3.2.3 SmartDC	44
3.3 Computação em Nuvem	45
3.3.1 Scallop4SC	45
3.3.2 A Cloud-based Architecture for Citizen Services in Smart Cities	46
3.3.3 Renewable Energy-aware Data Center Operations for Smart Cities The DC4Cities Approach	48
3.4 Cidades Inteligentes	49
3.4.1 Smart Lighting solutions for Smart Cities	49
3.4.2 Sensing as a Service Model for Smart Cities Supported by Internet of Things	51
3.4.3 An urban control center for the energy governance of a smart city	52
3.5 Análise e Oportunidades de Pesquisa	53
3.6 Considerações Parciais	55
4 MODELO ELCITY	57
4.1 Decisões de Projeto	57
4.2 Arquitetura	58
4.3 Detalhe das Decisões de Projeto nos Diferentes Níveis	60
4.3.1 Cidadãos	61
4.3.2 Data Center	61
4.3.3 Dispositivos da Cidade	62
4.4 Monitor de Energia	62

4.5 Gerenciadores de Elasticidade	66
4.5.1 Gerenciador de Elasticidade dos Dispositivos Móveis	66
4.5.2 Gerenciador de Elasticidade dos Dispositivos da Cidade	67
4.5.3 Gerenciador de Elasticidade do Data Center de Nuvem	67
4.6 Aplicação de Recomendação de Economia de Energia	70
4.7 Considerações Parciais	73
5 AVALIAÇÃO DOS RESULTADOS	75
5.1 Descrevendo o Ambiente de Testes	75
5.2 Avaliação do Monitor de Energia	75
5.2.1 Metodologia de Avaliação do Monitor de Energia	76
5.2.2 Consumo de Energia Sem Elasticidade	77
5.2.3 Consumo de Energia Com Elasticidade Ativa	77
5.3 Analisando Consumo de Energia na Cidade Inteligente	80
5.3.1 Metodologia de Avaliação da Aplicação de Recomendação de Economia de Energia	81
5.3.2 Consumo de Energia do Data Center	82
5.3.3 Consumo de Energia da Iluminação Pública	84
5.3.4 Consumo de Energia da Cidade Inteligente Com e Sem Elasticidade	86
5.4 Considerações Parciais	87
6 CONCLUSÃO	89
6.1 Contribuições	90
6.2 Limitações	91
6.3 Trabalhos Futuros	92
6.4 Publicações	92
REFERÊNCIAS	99

1 INTRODUÇÃO

Como resultado da migração rural e suburbana para as cidades, a vida urbana tornou-se um desafio significativo para os cidadãos e para a gestão da cidade (PERERA et al., 2014). Lixo, trânsito, energia, água, educação, desemprego, saúde e gestão da criminalidade são alguns dos problemas críticos (SU; LI; FU, 2011). Espera-se que até 2020 cerca de 80% da população esteja vivendo em cidades (European Environment Agency, 2006), o que implica em uma enorme carga nos recursos limitados como energia, água, transportes e habitação. Tendo em vista este crescimento contínuo, a necessidade de planejar e implementar soluções sustentáveis para melhorar o planejamento urbano é cada vez mais evidente.

Cidades inteligentes são a maior aposta para enfrentar estes desafios de forma eficiente (VöGLER et al., 2016). Utilizando tecnologias da informação e comunicação, espera-se que as cidades inteligentes forneçam um acompanhamento em tempo real, além de auxiliar no planejamento inteligente e no desenvolvimento urbano sustentável (PERERA et al., 2014). A gestão deste tipo de acompanhamento e apoio ao planejamento faz com que as cidades inteligentes tornem-se um desafio, devido a demanda por alto poder de processamento e armazenamento para disponibilizar as informações necessárias em um tempo aceitável para os usuários (KHAN; KIANI, 2012; ISMAIL, 2016). Particularmente, considerando o panorama de economia de energia, é possível observar que existem diferentes níveis de fontes de consumo, incluindo (i) *data centers*, (ii) dispositivos dos cidadãos (neste trabalho também chamados de usuários) e (iii) dispositivos da cidade (SCHLEICHER et al., 2016).

Todos estes níveis impactam de forma diferente no consumo de energia da cidade e, se observados de forma isolada, podem ter um pequeno potencial de economia de energia. Como exemplo podemos citar a movimentação de um cidadão com seu *smartphones*, utilizando este dispositivo para obter dados de contexto, onde o maior potencial de economia seria na utilização dos sensores do próprio dispositivo. No entanto, se observarmos este cenário de forma mais abrangente, seu padrão de movimentação poderia ser utilizado para otimizar diversos serviços ao seu redor, como transporte e iluminação pública. Este potencial se torna ainda maior quando esta visão se expande para todos os cidadãos de uma determinada cidade, podendo potencializar a otimização de diversos recursos.

Em um primeiro nível é possível observar os *data centers*, que são considerados como um dos maiores representantes quando se fala de consumo de energia (KHAN; KIANI, 2012). Em 2006, em particular, os *data centers* eram responsáveis pelo consumo de 61,4 bilhões de KWh (Quilowatt hora) nos Estados Unidos (LE et al., 2010). Além disso, em outro estudo realizado por Koomey (2011), os *data centers* foram apontados como responsáveis por consumir cerca de 1,3% da eletricidade mundial. Song, Zhang e Eriksson (2015) enfatizam que a eficiência energética está relacionada ao quanto *CPU-Bound* ou

*IO-Scavating*¹ um serviço executado no *data center* é. Devido a este fato, otimizações de desempenho nos serviços executados em um *data center* são importantes e possibilitam a economia de energia. Hoje, é possível observar o uso de computação em nuvem, seja de forma pública ou privada, em *data centers*. Mesmo considerando as demandas de migração dentro da nuvem, a utilização de energia permanece sendo um fator relevante (ORGERIE; ASSUNÇÃO; LEFEVRE, 2014).

No segundo nível, é possível observar o consumo de energia da recarga dos dispositivos dos usuários, neste caso, *smartphones*. Como já foi destacado por Satyanarayanan (1996), a bateria de dispositivos móveis, apesar dos grandes avanços nesta área, permanece limitada, especialmente quando se considera o uso de vários sensores para obter dados de contexto (NATH, 2012). Além disso, os próprios usuários também podem ajudar na obtenção de uma cidade inteligente que visa economizar energia. É possível usar os recursos de computação e de detecção de dispositivos móveis dos cidadãos, para oferecer serviços inteligentes e sensíveis ao contexto, se utilizado em conjunto com a computação em nuvem, bem como proporcionar o gerenciamento de energia com base em padrões de uso do dispositivo (PEJOVIC; MUSOLESI, 2015; VÖGLER et al., 2016).

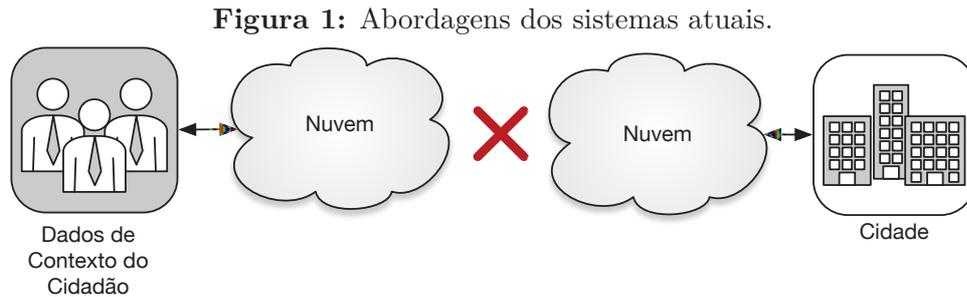
No terceiro nível pode-se considerar os dispositivos da cidade, onde destacam-se as despesas relacionadas com a iluminação pública. Em média, este tipo de dispositivo é responsável por 19% da utilização global de eletricidade em uma cidade e 6% da emissão total de gases do efeito estufa (CASTRO; JARA; SKARMETA, 2013). Com o objetivo de permitir uma melhor economia de energia, uma possível solução seria o desligamento de lâmpadas quando cidadãos não são encontrados em um raio específico de cobertura e/ou durante os períodos pré-definidos de acordo com o comportamento dos usuários. Um ponto importante é que este controle precisa ocorrer de forma transparente para os usuários, sem afetar a mobilidade e a segurança.

1.1 Motivação

A análise do estado da arte de cidades inteligentes possibilitou a elaboração da seguinte declaração: os modelos não consideram economia de energia integrando os três níveis acima mencionados. Além disso, observou-se iniciativas que empregam aquisição do contexto do usuário com um consumo de energia reduzido, mas não utilizam estes dados na gerência de energia nem no planejamento da cidade (CONSTANDACHE et al., 2009; NATH, 2012; CHON et al., 2014). Outros trabalhos concentram-se em utilizar a computação em nuvem para processar os dados de cidades inteligentes (YAMAMOTO; MATSUMOTO; NAKAMURA, 2012; KLINGERT et al., 2015; KHAN; KIANI, 2012), mas eles não consideram os dados dos usuários para avaliar possíveis planos de economia

¹Interações de rede estruturada sobre o protocolo TCP também usam ciclos de CPU para lidar com tarefas de processamento em nível de software para garantia de confiabilidade da transmissão de dados.

de energia. Por fim, alguns estudos consideram apenas a gestão das cidades inteligentes, negligenciando questões como o consumo de energia, sustentabilidade e escalabilidade de recursos (CASTRO; JARA; SKARMETA, 2013; PERERA et al., 2014; CARLI et al., 2014). De forma resumida, observou-se uma lacuna na combinação de diferentes fontes de dados como o *data center*, dispositivos e comportamento dos cidadãos e dispositivos localizados na cidade, para a redução ou gerência de energia em cidades inteligentes.



Fonte: elaborado pelo autor.

As propostas analisadas são pontuais, sem a integração necessária para cidades inteligentes, e as que apresentam algum nível de integração não levam em conta o principal ponto, os usuários. Na Figura 1 é possível observar que a integração ocorre apenas entre o usuário e a nuvem, ou a administração pública e a nuvem, e muitas vezes é unilateral sendo que não há integração entre os dados de contexto obtidos pelo usuário e a administração pública. O único componente em comum em sistemas de gestão e de usuários é a nuvem (KLINGERT et al., 2015). É necessário um modelo que integre o usuário à gestão da cidade, a fim de explorar a gerência de energia nos diversos níveis de uma cidade inteligente, além de possibilitar um retorno com investimentos mais próximos as reais necessidades dos cidadãos. Ainda, é fundamental que esta proposta leve em consideração a redução do alto consumo de energia dos *data centers* (LE et al., 2010; KOOMEY, 2011), onde os dados da cidade e usuários serão armazenados, processados e disponibilizados, além de considerar o consumo da bateria decorrente da obtenção dos dados de contexto, e a utilização destes dados para melhoria e redução do consumo de energia dos dispositivos localizados na cidade.

1.2 Questão de Pesquisa

A questão de pesquisa que o modelo proposto busca responder é a seguinte: *Como seria um modelo de gerência de energia multinível para cidades inteligentes que não compromete a qualidade dos serviços oferecidos e que atenda tanto cidadãos como a administração pública?*

A ideia desta proposta é um modelo elástico de economia multinível de energia para cidades inteligentes, considerando três níveis: (i) cidadão; (ii) *data center* de computação em nuvem; e (iii) dispositivos da cidade. A gerência de energia nos três níveis deve ocorrer

de forma transparente, o usuário deve fornecer dados de contexto para a administração pública através de uma infraestrutura sustentável de armazenamento e processamento de dados, assim como os dados coletados e processados devem servir para uma melhor gerência de energia da cidade. A economia de energia deve estar presente em todas as etapas, na obtenção de contexto, na infraestrutura de armazenamento e processamento de dados (*data center* de computação em nuvem) e na cidade.

1.3 Objetivos

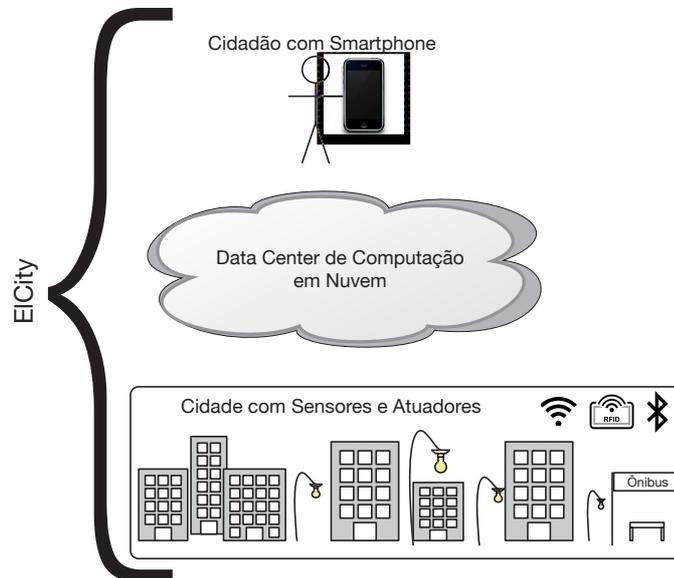
Considerando os dados acima apresentados, propõe-se um modelo chamado **ElCity**², um modelo elástico de economia multinível de energia para cidades inteligentes. A palavra-chave multinível refere-se aos três diferentes níveis apresentados: (i) os cidadãos; (ii) o *data center* de computação em nuvem; e (iii) a cidade, conforme apresentado na Figura 2. Cidadãos (usuários) são responsáveis por fornecer dados de contexto para outros níveis, visando sempre a economia de energia em seus *smartphones*. Em relação à cidade, este trabalho concentra-se especificamente na iluminação pública, que normalmente representa uma das maiores despesas com energia das prefeituras. O *data center* é utilizado para processar e armazenar os dados fornecidos pelos cidadãos e dispositivos localizados na cidade, gerando planos de economia de energia para essas fontes de dados. A administração pública pode utilizar as informações geradas no *data center* para projetar uma cidade mais sustentável, de modo a otimizar não só a tomada de decisões e os investimentos em áreas de maior movimento, mas também reduzir os gastos com iluminação pública. Na proposta também é possível observar uma extensão do conceito de elasticidade, presente na computação em nuvem, que se refere à capacidade de ligar e desligar automaticamente os recursos dos três níveis considerados. Tendo em vista estas questões, o objetivo principal deste trabalho é desenvolver um modelo elástico de economia multinível de energia para cidades inteligentes.

Para atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- (i) Realizar o levantamento dos conceitos básicos necessários para o entendimento de obtenção de dados de contexto de usuários, computação em nuvem e cidades inteligentes;
- (ii) Realizar um estudo sobre o estado da arte relacionado a modelos e técnicas para economia de energia em cidades inteligentes, computação em nuvem e obtenção de contexto de usuários através de dispositivos móveis;
- (iii) Comparar os trabalhos encontrados neste estudo com base em critérios definidos e encontrar lacunas;

²<https://github.com/grostirolla1/elcity>

Figura 2: Níveis atingidos pelo modelo ElCity.



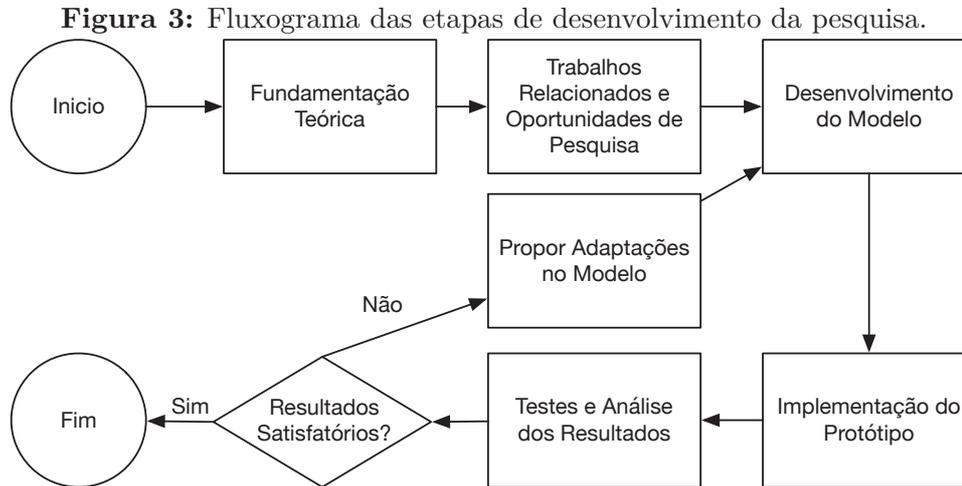
Fonte: elaborado pelo autor.

- (iv) Desenvolver um modelo capaz de prover economia de energia de forma transparente, sem afetar os serviços disponíveis nos três níveis analisados;
- (v) Desenvolver um protótipo do modelo para a realização de testes;
- (vi) Avaliar o protótipo com a utilização de dados reais;
- (vii) Analisar os resultados obtidos a fim de determinar quais são os níveis de consumo de energia mais decisivos entre os descritos anteriormente.

1.4 Etapas de Desenvolvimento da Pesquisa

O desenvolvimento desta pesquisa ocorreu de acordo com o fluxograma apresentado na Figura 3. No fluxograma estão presentes 6 etapas: (i) Fundamentação Teórica; (ii) Trabalhos Relacionados e Oportunidades de Pesquisa; (iii) Desenvolvimento do Modelo; (iv) Implementação do Protótipo; (v) Testes e Análise dos Resultados e (vi) Propor Alterações no Modelo. Na primeira etapa foi realizado um estudo das teorias que envolvem o tema de pesquisa a fim de prover a base necessária para o entendimento deste trabalho. Após isto, iniciou-se um levantamento dos trabalhos relacionados ao tema de pesquisa com o objetivo de identificar os problemas envolvendo o tema, e possíveis lacunas/opportunidades de pesquisa. Na etapa (iii) é proposto um modelo, visando preencher a lacuna identificada respondendo a questão de pesquisa proposta. Por fim, as etapas (iv), (v)

e (vi) consistem na implementação de um protótipo, realização de testes e análise dos resultados, o que por sua vez pode resultar em alterações no modelo proposto de acordo com o necessário. Ao final do desenvolvimento da pesquisa foi realizada a elaboração de artigos científicos relativos ao tema e escrita desta dissertação.



Fonte: elaborado pelo autor.

1.5 Organização do Texto

Esta proposta está organizada da seguinte forma, o Capítulo 2 introduz conceitos fundamentais para a compreensão do restante do trabalho e da pesquisa proposta. O Capítulo 3 apresenta uma lista com trabalhos relacionados aos temas desta pesquisa, bem como um comparativo e identificação das lacunas encontradas. O Capítulo 4 apresenta o modelo proposto para preencher a lacuna identificada no capítulo anterior e responder a questão de pesquisa. O Capítulo 5 traz a metodologia de avaliação do modelo proposto e os resultados que foram obtidos. Por fim, o Capítulo 6 apresenta as conclusões que foram obtidas ao final desta pesquisa. Ainda, para facilitar a compreensão os Capítulos 3, 4 e 5 contam com uma seção de considerações parciais ao fim, o que visa apresentar de forma resumida o seu conteúdo e facilitar a compreensão.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar conceitos considerados fundamentais para o desenvolvimento deste trabalho, sendo dividido em três seções principais, onde cada uma das seções representa um dos níveis que é considerado neste trabalho. Primeiramente são apresentados fundamentos relacionados a dispositivos móveis, com uma visão geral de computação ubíqua, um conceito pertinente pois o sistema deve utilizar informações disponíveis no ambiente, detectar serviços e dispositivos e adaptar-se a eles. A seguir é apresentado contexto, e por fim, é apresentado o problema do consumo de energia em dispositivos móveis.

A segunda seção apresenta conceitos relacionados a computação em nuvem, um item fundamental para a integração multinível proposta, e também um dos responsáveis por boa parcela do consumo de energia mundial, como apresentado no capítulo anterior. Por fim, são apresentadas as cidades inteligentes, a definição que será seguida por este trabalho, suas classificações, a importância de cidades inteligentes no planejamento urbano e no consumo de energia de cidades.

2.1 Dispositivos Móveis

Esta seção tem como objetivo introduzir os principais conceitos que foram utilizados envolvendo os dispositivos móveis, de onde serão coletados os dados de contexto do cidadão. Para tal, apresenta-se abaixo conceitos relacionados a computação ubíqua, contexto, problemas e alternativas relacionadas ao consumo de energia em dispositivos móveis.

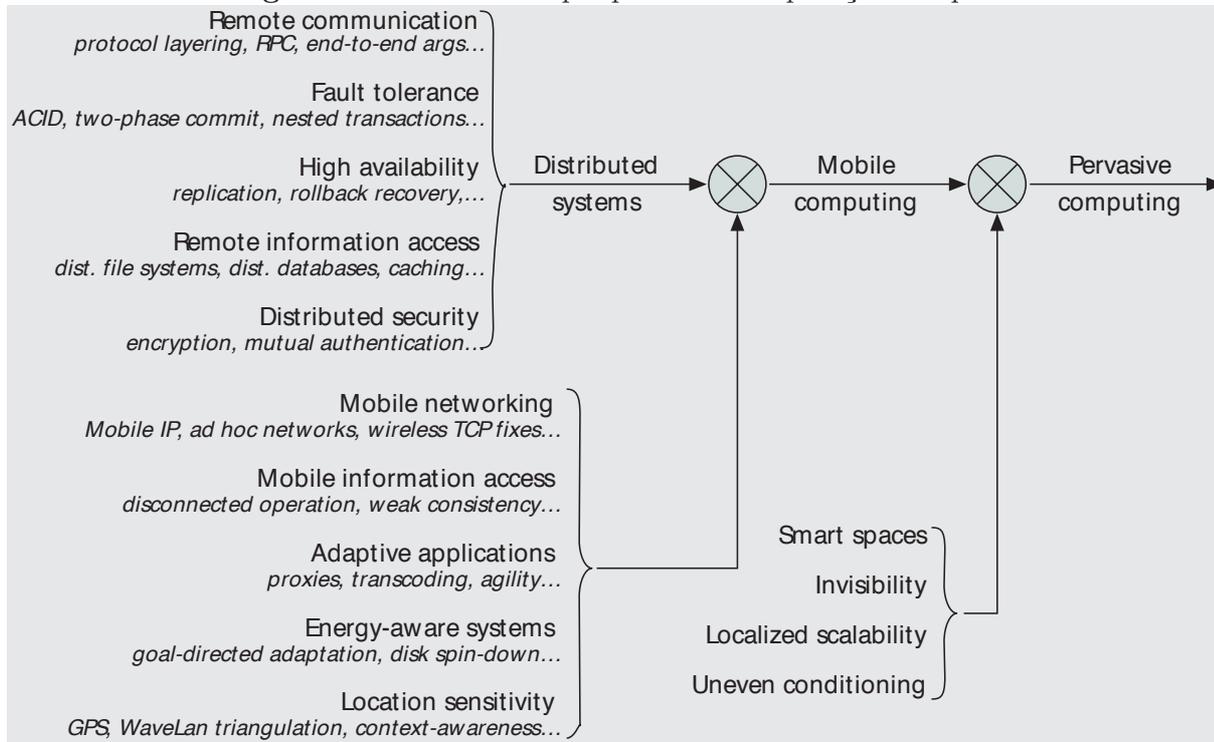
2.1.1 Computação Ubíqua

Computação Ubíqua é um conceito apresentado por Weiser (1991) no qual o autor caracteriza como será a computação do século XXI. Segundo Weiser (1991) o “desaparecimento” de uma determinada tecnologia no dia a dia de um usuário é uma consequência de sua importância e boa aceitação, como é o caso da escrita. O autor ainda atribui esta característica de desaparecimento não a tecnologia em si, mas a própria psicologia humana, que ao aprender suficientemente algo, deixa de ter ciência de sua utilização.

Quando Weiser (1991) apresentou sua visão de computação ubíqua não se dispunha da tecnologia necessária para que ela se tornasse realidade, e a implementação não saiu dos laboratórios da Xerox PARC. Satyanarayanan (2001) dez anos depois afirmou que os dispositivos necessários para tornar a visão de Weiser realidade já estavam disponíveis e, devido a forte relação entre a computação ubíqua e as áreas de computação móvel e sistemas distribuídos, elencou problemas de pesquisa presentes na computação ubíqua, ilustrados na Figura 4. Alguns destes problemas ainda continuam presentes, como é o

caso dos “sistemas cientes de energia” (*Energy-aware systems*).

Figura 4: Problemas de pesquisa em Computação Ubíqua.



Fonte: obtida em (SATYANARAYANAN, 2001).

Segundo Satyanarayanan (2001) esta figura mostra como os problemas de pesquisa em computação ubíqua (apresentado como *Pervasive computing*) se relacionam com os de computação móvel e sistemas distribuídos, além do aumento da complexidade dos problemas encontrados nestas áreas. Como problemas encontrados o autor lista variação imprevisível na qualidade da rede, baixa confiança e robustez dos dispositivos móveis, limitações de recursos e preocupação com o consumo de energia da bateria, sendo que esta última necessita que hajam melhorias tanto em nível de software quanto hardware.

Costa, Yamin e Geyer (2008) ressalta a adaptabilidade como um conceito central da computação ubíqua, ou seja, para ser ubíquo o sistema deve utilizar informações relevantes disponíveis no ambiente, detectar serviços e dispositivos e adaptar-se a eles. Como os autores expõem, um sistema que não precise ser configurado de forma estática ou manual para encontrar os recursos de um determinado ambiente, a qualquer momento. Satyanarayanan (2001) aponta a proatividade como uma faca de dois gumes. Ou seja, sistemas proativos devem ser cuidadosamente projetados para não perturbar o usuário, ferindo o conceito de invisibilidade.

Pesquisas recentes em computação ubíqua estão cada vez mais focadas no fornecimento de capacidades cognitivas para dispositivos móveis, permitindo que eles possam ser treinados para prever eventos futuros a partir de dados de sensores atuais e de análise de eventos passados (PEJOVIC; MUSOLESI, 2015), principalmente com foco em cidades

inteligentes. Existe ainda uma grande preocupação na limitação imposta pela bateria, já apontada por Weiser (1991), Satyanarayanan (1996) e Satyanarayanan (2001), e agora sendo uma das preocupações das mais diversas áreas envolvendo computação móvel, como é o caso dos sistemas de localização (CHON et al., 2014; LIN et al., 2010).

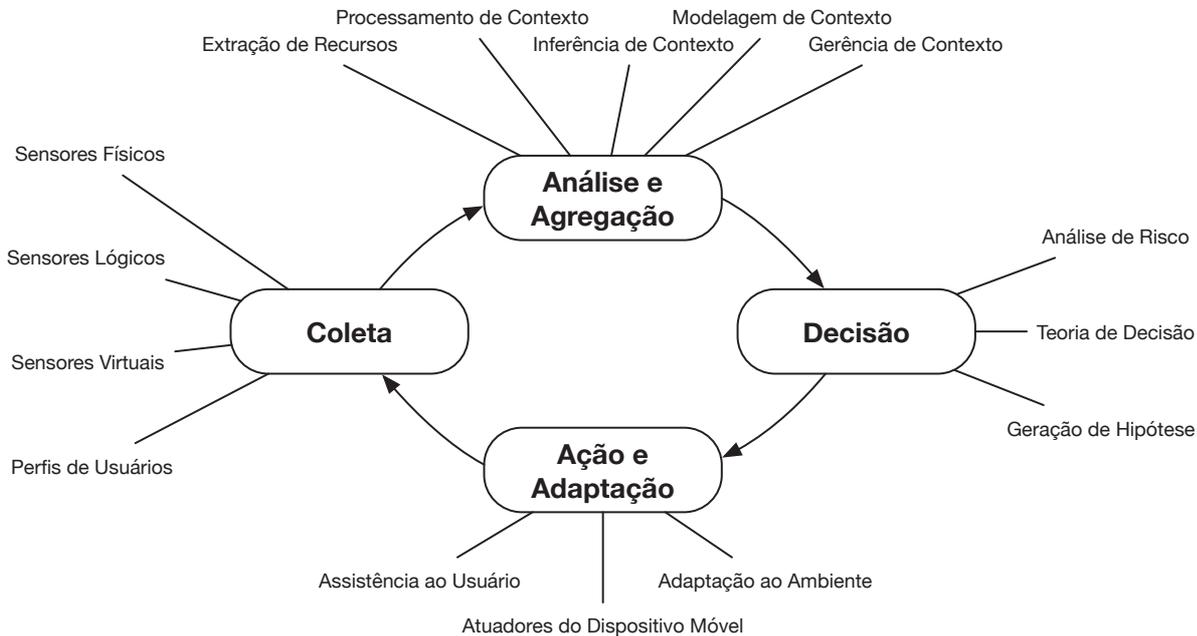
2.1.2 Contexto

O termo computação sensível a contexto tem como principal referência o artigo apresentado por Dey (2001). O objetivo da computação sensível a contexto é coletar e utilizar informações que cercam o usuário para então selecionar, configurar e prover uma variedade de serviços pertinentes naquele momento (HOAREAU; SATOH, 2009). O contexto de um usuário pode conter atributos como localização física, estado fisiológico (temperatura corporal e frequência cardíaca), estado emocional, dentre outros (SATYANARAYANAN, 2001).

Alguns autores como Costa, Yamin e Geyer (2008) e Hoareau e Satoh (2009) apresentam contexto dividido em duas partes, sensibilidade a contexto e gerência de contexto. A parte de sensibilidade a contexto trata da aquisição de dados de contexto, abstração destes dados e por fim do entendimento do contexto (HOAREAU; SATOH, 2009). A gerência de contexto é responsável pela tomada de ações em resposta aos dados coletados, e realizar as adaptações necessárias, podendo estas alterações expandirem as capacidades do dispositivo utilizando os recursos deste novo ambiente (COSTA; YAMIN; GEYER, 2008). Ainda segundo Schilit, Adams e Want (1994) as ações tomadas com base nas informações de contexto são simples regras “SE-ENTÃO” previamente definidas.

Para melhor compreensão, o ciclo de operação de um sistema sensível a contexto está ilustrado na Figura 5, elaborada com base em Dobson et al. (2006). O ciclo destaca a aquisição de dados brutos de sensores e perfis de usuário, seguido pela fase de análise e agregação dos dados, que é responsável por processar, inferir e modelar contexto. Como resultado obtém-se dados que permitem a tomada de decisão e por fim a realização de ações e adaptações baseadas nas decisões tomadas (KNAPPMeyer et al., 2013).

Sensibilidade a contexto é um componente fundamental deste trabalho, uma vez que localização é uma das aplicações clássicas de sensibilidade a contexto (DEY, 2001; HOAREAU; SATOH, 2009). Aplicações sensíveis a contexto utilizam localização de pessoas, objetos e dispositivos computacionais como fonte principal de informação contextual (HOAREAU; SATOH, 2009). Atualmente a localização de um dispositivo/usuário é obtida através de sistemas como GPS (*Global Positioning System*), sensores ultrassônicos, tags RFID (*Radio Frequency Identification*), e dispositivos *Bluetooth*, tendo como saída coordenadas, ou localização com uma precisão a nível de cômodos de uma casa no caso das duas últimas alternativas (HOAREAU; SATOH, 2009).

Figura 5: Ciclo de um sistema sensível a contexto.

Fonte: adaptado de (DOBSON et al., 2006).

2.1.3 Consumo de Energia em Dispositivos Móveis

Satyanarayanan (1996) já ressaltava a necessidade das aplicações para dispositivos móveis serem sensíveis ao consumo de energia, apesar de uma provável evolução na tecnologia das baterias. A preocupação do autor abrange diversos níveis tanto em hardware quanto software para ser eficaz na otimização do consumo da bateria.

Pesquisas recentes estão cada vez mais focadas no fornecimento de capacidades cognitivas para telefones móveis (PEJOVIC; MUSOLESI, 2015). Isso permite que o telefone seja treinado para prever eventos futuros a partir de dados atuais e passados, no entanto, a coleta de dados para estas previsões possui um custo elevado. Pejovic e Musolesi (2015) também alertam que em computação móvel preditiva a utilização frequente de diferentes sensores, colaboração entre vários dispositivos e processamento das informações são necessários para a antecipação, o que leva a um elevado consumo de uma fonte de energia limitada (bateria). Este ponto é enfatizado também no trabalho de Chon et al. (2014), onde o autor demonstra que apesar desta limitação da bateria, a utilização de *smartphones* ainda é a mais indicada na tarefa de monitorar o comportamento humano por três motivos: (i) possuir quase 100% de adoção; (ii) são fortemente acoplados ao dia a dia do usuário; e (iii) são capazes de determinar sua localização através de diferentes abordagens.

Visando solucionar este problema, surgem então diversas técnicas para reduzir o consumo de energia destes dispositivos, mantendo as funcionalidades de sensoriamento. Segundo (PEJOVIC; MUSOLESI, 2015) os dois meios populares de reduzir o consumo de energia são: (i) amostragem adaptativa (amostragem com uma menor frequência) e, (ii)

no caso de um dispositivo com vários sensores, ligá-los de forma hierárquica (dos sensores de baixa potência para os de maior potência). EEMSS (*Energy Efficient Mobile Sensing System*) proposto por Wang et al. (2009), por exemplo, liga os sensores de forma hierárquica, ativando os sensores de baixo consumo primeiro, seguido dos sensores com maior consumo de energia caso algum evento de interesse seja detectado.

Com relação a formas de medição de consumo em dispositivos móveis, podem ser citadas três diferentes abordagens:

- Medições físicas com osciloscópios, multímetros ou ferramentas como o Monsoon Power Monitor, fornecendo uma alta precisão, no entanto medindo apenas o aparelho como um todo, e não componentes isolados (sensores) (VERGARA; NADJIM-TEHRANI; PRIHODKO, 2014).
- Criar um perfil de cada sensor utilizando as ferramentas de medição anteriores e estimando o consumo de acordo com a utilização de cada sensor separadamente (LIN et al., 2010);
- Utilizar ferramentas fornecidas pelos fabricantes como por exemplo Trepn (QUALCOMM, 2015) fornecido para processadores da série Snapdragon, Power Tutor (POWER-TUTOR, 2015) que fornece suporte aos *smartphones* da série Nexus, ambos fornecendo análise a nível de componentes isolados, ou ainda a utilização da interface fornecida pela bateria, como é o caso do Sesame (DONG; ZHONG, 2011) que apresenta estatísticas de consumo de todo o dispositivo.

2.2 Computação em Nuvem

Esta seção tem como objetivo introduzir os principais conceitos que foram utilizados envolvendo a computação em nuvem, sua classificação e o consumo de energia neste cenário.

2.2.1 Classificações de Computação em Nuvem

Segundo a definição de Mell e Grance (2011) computação em nuvem é um modelo para permitir acesso ubíquo, conveniente, e sob demanda a um conjunto compartilhado de recursos computacionais configuráveis que podem ser rapidamente alocados e liberados com um esforço mínimo de gerenciamento. Com a introdução da computação em nuvem, tarefas que antes eram executadas localmente agora passam a ser executadas em servidores de localização desconhecida através da Internet (HAYES, 2008). Os modelos oferecidos pela computação em nuvem são classificados em três categorias:

- (i) SaaS - Software como Serviço: A capacidade fornecida ao usuário limita-se aos aplicativos executados em uma infraestrutura de nuvem. As aplicações são acessíveis

a partir de vários dispositivos clientes e o consumidor não gerencia ou controla a infraestrutura onde está hospedado o software, esta responsabilidade é do provedor do serviço (MELL; GRANCE, 2011; PUTHAL et al., 2015).

- (ii) PaaS - Plataforma como Serviço: A capacidade fornecida ao usuário neste caso, está localizada um nível abaixo da SaaS, onde é possível implantar os aplicativos em uma determinada infraestrutura de nuvem onde estão disponíveis linguagens de programação, bibliotecas, serviços e ferramentas de suporte. O consumidor não gerencia ou controla a infraestrutura da nuvem como rede, servidores, sistemas operacionais, ou armazenamento, mas tem controle sobre os aplicativos implementados e configurações (MELL; GRANCE, 2011; PUTHAL et al., 2015).
- (iii) IaaS - Infraestrutura como Serviço: A capacidade fornecida ao usuário vai desde questões como armazenamento, redes e outros recursos computacionais fundamentais, além das previamente mencionadas em PaaS e SaaS.

Com relação a implantação, as infraestruturas de nuvem podem ser mantidas das seguintes formas: (i) Nuvem Privada; (ii) Nuvem Pública; (iii) Nuvem Comunitária; e (iv) Nuvem Híbrida (PUTHAL et al., 2015).

- (i) Nuvem Privada: A infraestrutura de nuvem é provisionada para uso exclusivo de uma única organização que compreende. Pode ser propriedade, gerenciada e operada pela organização, um terceiro, ou alguma combinação deles, e pode ter localização física ou não no local (MELL; GRANCE, 2011; PUTHAL et al., 2015);
- (ii) Nuvem Pública: A infraestrutura de nuvem é provisionada para uso aberto pelo público em geral, sendo esta infraestrutura compartilhada entre diversos usuários. Pode ser propriedade, gerenciados e operados por uma empresa, universidade ou organização governamental, com localização no provedor de nuvem (MELL; GRANCE, 2011; PUTHAL et al., 2015);
- (iii) Nuvem Comunitária: A infraestrutura de nuvem é provisionada para uso exclusivo por uma comunidade específica que têm preocupações comuns. Pode ser propriedade, gerenciada e operada por uma ou mais organizações da comunidade ou ainda um terceiro, podendo ou não existir fisicamente no local (MELL; GRANCE, 2011; PUTHAL et al., 2015);
- (iv) Nuvem Híbrida: A infraestrutura de nuvem é uma composição de duas ou mais infraestruturas de nuvem (privada, comunitária ou pública) que permanecem entidades únicas, mas são unidas por tecnologia padronizada ou proprietária (MELL; GRANCE, 2011; PUTHAL et al., 2015).

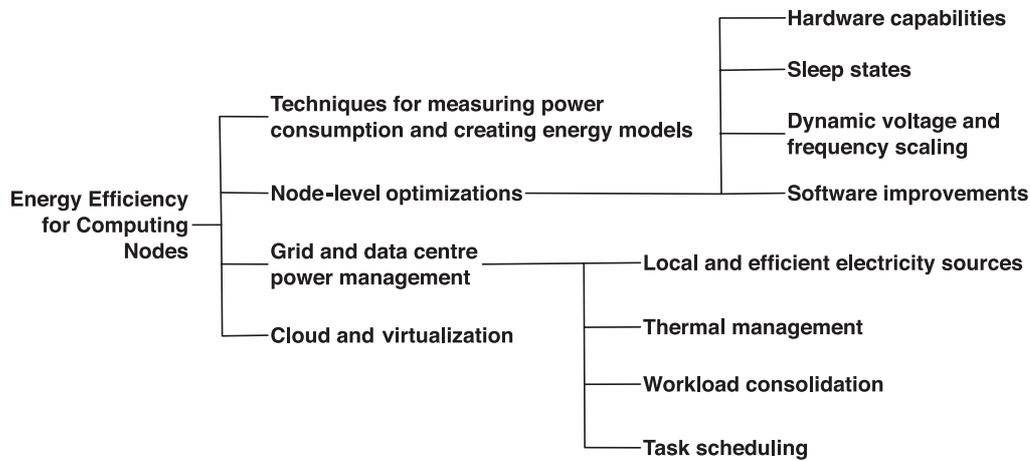
Outra característica chave da computação em nuvem é a elasticidade (LORIDO-BOTRAN; MIGUEL-ALONSO; LOZANO, 2014), que permite que os aplicativos aloquem e liberem recursos de forma dinâmica, ajustando às demandas da aplicação. Todas estas funcionalidades providas pela computação em nuvem, impulsionou sua adoção massiva, fazendo com que provedores deste tipo de serviço necessitem de uma grande quantidade de eletricidade para manter o funcionamento de seus *data centers*, e prover um nível de serviço ideal (GARG et al., 2011). Com o aumento da demanda por *data centers*, o consumo de energia elétrica por parte destes dobrou entre os anos de 2000 e 2005, passando de 70,8 bilhões de kWh para 152,5 bilhões de kWh (MASTELIC et al., 2014). Ainda, o consumo de energia de aplicações do tipo CPU intensivas aumenta drasticamente o consumo de CPU, e conseqüentemente o consumo de energia (CHEN et al., 2014).

2.2.2 Consumo de Energia em Computação em Nuvem

Quando se observa sistemas distribuídos de larga escala, diversas técnicas de economia de energia podem ser utilizadas, como apresentado na taxonomia de Orgerie, Assunção e Lefevre (2014), ilustrada na Figura 6. Observando mais especificamente o cenário da computação em nuvem, uma das suas principais características é a elasticidade, na qual os usuários podem escalar seus recursos computacionais a qualquer momento, de acordo com a demanda ou o tempo de resposta desejado (LORIDO-BOTRAN; MIGUEL-ALONSO; LOZANO, 2014). Considerando uma aplicação paralela de longa execução, um usuário pode querer aumentar o número de instâncias para tentar reduzir o tempo de conclusão da tarefa. Logicamente, o sucesso deste processo vai depender tanto do grão, quanto da modelagem da aplicação. Por outro lado, se a tarefa não escala de forma linear ou perto de uma forma linear, e se o utilizador é flexível com respeito ao tempo de conclusão, o número de instâncias pode ser reduzida. Isso resulta em uma menor quantidade de nós \times horas, e portanto, em um custo mais baixo e melhor uso da energia. Graças aos avanços na área de virtualização (PETRIDES; NICOLAIDES; TRANCOSO, 2012), a elasticidade em computação em nuvem pode ser uma alternativa viável para obter economia de custo significativa quando comparado com o método tradicional de manter uma infraestrutura de TI baseada em *cluster*. Normalmente, neste último caso, há um dimensionamento para o uso de pico, sendo subutilizada quando observamos toda a execução do aplicativo ou ainda, ao analisar o uso real da infraestrutura.

Entretanto, elasticidade pode ser um problema se considerado apenas desempenho, desconsiderando o consumo de energia. Ambos são diretamente relacionados ao consumo de recursos, o que também pode ajudar a medir a qualidade da elasticidade. Embora a elasticidade permita que os aplicativos aloquem e liberem recursos de forma dinâmica, ajustando às demandas da aplicação, estabelecer limites apropriados, medir o desempenho e consumo de energia com precisão neste ambiente não são tarefas fáceis (LORIDO-

Figura 6: Taxonomia de técnicas para melhoria da eficiência energética em sistemas distribuídos de larga escala.



Fonte: obtida em (ORGERIE; ASSUNÇÃO; LEFEVRE, 2014).

BOTRAN; MIGUEL-ALONSO; LOZANO, 2014). Desta forma, um utilizador pode conseguir um bom desempenho considerando o tempo para executar a sua aplicação, mas com uma grande quantidade de recursos, resultando em um desperdício de energia. A ideia de apenas obter um melhor desempenho da aplicação com uma execução elástica, em alguns casos, não é suficiente para usuários e administradores da nuvem. Os usuários acabam pagando por um maior número de recursos, não efetivamente utilizados, de acordo com o paradigma *pay-as-you-go*. A medição do consumo de energia de tais sistemas elásticos não é uma tarefa fácil (LORIDO-BOTRAN; MIGUEL-ALONSO; LOZANO, 2014). Muitos trabalhos se concentram em medição e como estimar o consumo de energia em *data centers* (LUO et al., 2013; CHEN et al., 2014; TIAN; LIN; LI, 2014; PAYA; MARI-NESCU, 2013; ZHANG et al., 2014; YANG et al., 2011; BELOGLAZOV; BUYYA, 2010), no entanto, essas tarefas ainda são desafios ao considerar sistemas elásticos.

A fim de melhorar o consumo de energia, alguns pesquisadores estão analisando estas plataformas e propondo novas formas de alocar recursos com foco em reduzir o consumo de energia. Alguns trabalhos focam em satisfazer os acordos de nível de serviço (SLAs) necessários e alocando os recursos a fim de reduzir ao máximo a energia gasta (FARGO et al., 2013; LUO et al., 2013), outros trabalhos avaliam diferentes perfis de energia de aplicações (*CPU-Bound*, *Memory-Bound*, entre outras) a fim de escalonar as tarefas e reduzir o consumo de energia (BERGEN et al., 2014). Liang et al. (2014) prevê a inicialização de aplicações a fim de alocar recursos de forma eficiente, ou até mesmo trabalhos que propõem o escalonamento de tarefas de acordo com o consumo de energia de cada aplicação (ZHANG et al., 2014).

2.3 Cidades Inteligentes

Esta seção apresenta os principais conceitos que foram utilizados envolvendo cidades inteligentes, as classificações dentro de cidades inteligentes, o planejamento urbano neste cenário, e como funciona o consumo de energia em cidades inteligentes. Neste caso em particular, optou-se por utilizar a definição de cidades inteligentes fornecida por Harrison et al. (2010) que define como uma cidade que conecta a infraestrutura física, tecnologia da informação, a infraestrutura social, e a infraestrutura de negócios para tirar proveito da inteligência coletiva da cidade.

2.3.1 Classificação de Cidades Inteligentes

Existem diversas definições de cidades inteligentes (do inglês *Smart Cities*) além de uma gama de variantes conceituais obtidas pela substituição de “*Smart*” por palavras como “*Intelligent*” ou “*Digital*” (ALBINO; BERARDI; DANGELICO, 2015). O rótulo de “cidade inteligente” é frequentemente distorcido e utilizado de maneira inconsistente, uma vez que não há um único modelo de enquadramento de uma cidade inteligente, nem uma única definição (O’GRADY; O’HARE, 2012). Nesta seção estão presentes algumas das definições, além de componentes de uma cidade inteligente, para facilitar a compreensão de quais aspectos de cidades inteligentes este trabalho está relacionado.

O termo foi usado pela primeira vez na década de 1990 com foco em novas tecnologias e infraestruturas modernas dentro das cidades, sendo o Instituto da Califórnia para comunidades inteligentes um dos primeiros a se concentrar em como as comunidades podem se tornar inteligentes e como uma cidade poderia ser projetada para implementar estas tecnologias (ALAWADHI et al., 2012).

Segundo Zygiaris (2013) cidade inteligente é entendida como uma certa capacidade intelectual que aborda vários aspectos sócio-técnicos e sócio-econômicos de crescimento. Estes aspectos levam a concepções da cidade inteligente como “verde” referindo-se a infraestrutura urbana para a proteção do meio ambiente e redução da emissão de gás carbônico, “interligada” com a revolução da economia de banda larga, “inteligente” declarando a capacidade de produzir informações que agregam valor com dados em tempo real da cidade, de sensores e atuadores, enquanto os termos “cidades inovadoras” e “cidades do conhecimento” referem-se a habilidade da cidade de inovar baseado no conhecimento e na criatividade do capital humano.

Dirks, Gurdgiev e Keeling (2010) ressaltam a importância da integração dos vários sistemas de uma cidade (transporte, energia, educação, saúde, infraestrutura física, alimentação, saneamento e segurança pública) na criação de uma cidade inteligente. Os autores ainda ressaltam que em ambientes densos como os das cidades inteligentes, normalmente os sistemas não operam de forma isolada. A informação gerada por cada sistema

individualmente não é suficiente para inferir alguma inteligência e criar uma cidade inteligente, logo, o sistema deve ser tratado como um todo (DIRKS; GURDGIEV; KEELING, 2010).

Lombardi et al. (2012) elenca seis componentes com diferentes aspectos da vida urbana, a fim de facilitar a identificação dos sistemas que operariam em uma cidade inteligente, apresentados na Tabela 1. A economia inteligente está associada com a presença de indústrias no domínio da tecnologia da informação e comunicação ou que a empregam nos processos de produção. Mobilidade inteligente refere-se à utilização das tecnologia da informação e comunicação no transporte para melhorar o tráfego urbano. Ressaltam-se três componentes na Tabela 1, *Smart Governance*, *Smart Mobility* onde acredita-se que o trabalho esteja atrelado, além de *Smart Environment* onde estaria de forma mais presente a questão da energia e sustentabilidade.

Tabela 1: Componentes de uma cidade inteligente e aspectos relacionados a vida urbana

Componentes	Aspectos
Economia inteligente (<i>Smart Economy</i>)	Indústria
Pessoa inteligente (<i>Smart People</i>)	Educação
Governança inteligente (<i>Smart Governance</i>)	Tomada de decisões
Mobilidade inteligente (<i>Smart Mobility</i>)	Logística e infraestrutura
Ambiente inteligente (<i>Smart Environment</i>)	Eficiência e sustentabilidade
Moradia inteligente (<i>Smart Living</i>)	Segurança e qualidade

Fonte: adaptado de (LOMBARDI et al., 2012).

Com relação aos projetos de cidades inteligentes sendo desenvolvidos ao redor do mundo, pode-se citar que no início de 2013 havia cerca de 143 projetos em andamento ou concluídos (LEE; HANCOCK; HU, 2014). A maior parte destas iniciativas, 50 projetos estavam localizados na Ásia, seguida de 47 projetos na Europa, 35 na América do Norte, 10 na América do Sul e outros 10 distribuídos entre Oriente Médio e África (LEE; HANCOCK; HU, 2014).

2.3.2 Planejamento Urbano

Planejamento urbano refere-se ao planeamento regional em uma área residencial, o qual deve levar em conta vários parâmetros, como o meio ambiente, a população, questões financeiras e transporte (ANTHOPOULOS; VAKALI, 2012). Ou seja, o planejamento urbano é responsável por controlar o desenvolvimento e a organização de uma cidade, determinando as zonas de urbanização e o uso das áreas de terra, a localização de várias redes públicas e espaços comuns levando em conta os parâmetros mencionados anteriormente (ANTHOPOULOS; VAKALI, 2012).

Quando combinados o planejamento urbano e as cidades inteligentes, surge a questão: “como poderiam dados coletados em uma cidade inteligente colaborar no planejamento

urbano?”. Pan et al. (2013a) aponta que informações sobre a mobilidade são úteis para sistemas de tráfego; além disso, aprender o comportamento humano e as relações sociais pode beneficiar a saúde pública, segurança e comércio, caracterização fundamental para o planejamento urbano. Kitchin (2014) também ressalta que a esperança de *Big Data* é uma transformação no conhecimento e governança das cidades através da criação de uma grande massa de dados que ofereça a compreensão em tempo real e controle de urbanidade.

Os processos técnicos e políticos do planejamento urbano podem se beneficiar da análise e mineração de dados (ZHENG et al., 2011). Pan et al. (2013b) aponta em detalhe os dados do rastreamento que poderiam ajudar o planejamento urbano de muitas maneiras, respondendo duas questões fundamentais: (i) quanta infraestrutura é necessária em uma determinada região; e (ii) como distribuí-la. A frequência de visitas está diretamente relacionada com as demandas de infraestrutura. Padrões de deslocamento poderiam responder a segunda questão. Pesquisas anteriores já haviam revelado a potencial relação entre a frequência de visitas e a função social da região (PAN et al., 2013b). De acordo com a frequência de visitas às infraestruturas Zheng et al. (2011) afirma ser possível medir se o atual planejamento da cidade é suficiente ou excessivo para atender as demandas de serviço. Como exemplo mais comum pode ser citada a utilização do número de veículos nas estradas para avaliar o planejamento de tráfego.

Com base nestas pesquisas é possível notar a relevância do levantamento de dados de contexto em cidades inteligentes. Com estes dados, e a aplicação de técnicas de mineração de dados, é possível buscar uma melhoria no convívio e na estrutura de uma cidade além de um dimensionamento apropriado de recursos por parte do órgão administrativo.

2.3.3 Consumo de Energia em Cidades Inteligentes

Os avanços tecnológicos estimulam novos produtos e serviços, que inevitavelmente resultam em um consumo intensivo de recursos, como o consumo de energia (DE PAOLA et al., 2014). Ao mesmo tempo, o aumento dos estudos que observam os custos em termos de pegada ecológica apontam que as atuais taxas de utilização da energia em todo o mundo não são mais sustentáveis (DE PAOLA et al., 2014).

Cidades inteligentes surgem desempenhando um importante papel no desenvolvimento econômico sustentável de uma determinada área, e são consideradas um elemento-chave para a geração de riqueza, conhecimento e diversidade econômica e social (CASTRO; JARA; SKARMETA, 2013). Castro, Jara e Skarmeta (2013) apontam as cidades inteligentes como o “motor” para atingir uma infraestrutura sustentável e facilitar o desenvolvimento de indústrias, edifícios e cidadãos, onde a primeira meta para atingir a sustentabilidade é reduzir o consumo de energia e os níveis dos gases que causam o efeito estufa. Nos últimos anos, portanto, a redução do consumo de energia em cidades tem ganho cada vez mais interesse entre os pesquisadores, visando sustentabilidade ambiental

de longo prazo, possuindo duas principais vertentes (não exclusivas): (i) *Smart Grid*; e (ii) *Smart Buildings*.

Smart Grid é um conceito que combina tecnologia da informação, redes de computadores e de energia elétrica, tendo como principal característica o estabelecimento de uma comunicação de duas vias em todos os nós da rede de energia, utilizando uma infraestrutura de medição avançada (MORVAJ; LUGARIC; KRAJCAR, 2011). Segundo Morvaj, Lugaric e Krajcar (2011) *Smart Grids* permitem o controle e otimização de fontes de energia renováveis além de fornecerem informações em tempo real para controlar os nós, permitindo equilíbrio quase instantâneo da oferta e da procura a nível de consumidores individuais.

Já o termo *Smart Building* (ou construções inteligentes) refere-se a utilização de dispositivos e equipamentos de rede, além de subsistemas adicionais para gerenciamento e controle de fontes renováveis de energia e eletrodomésticos, usando na maioria das vezes tecnologias de comunicação sem fio (MORVAJ; LUGARIC; KRAJCAR, 2011). Segundo Morvaj, Lugaric e Krajcar (2011) *Smart Buildings* possuem: (i) Sensores para o acompanhamento e envio de mensagens em caso de mudanças; (ii) Atuadores para realização de uma ação física; (iii) Controladores que baseiam-se em regras definidas pelo usuário; (iv) Unidade central que permite a programação de unidades no sistema; (v) Interface de comunicação entre usuário e sistema; (vi) Rede para permitir a comunicação entre as unidades; (vii) Medidores inteligentes que oferecem comunicação nos dois sentidos entre o cliente e a empresa de distribuição de energia (MORVAJ; LUGARIC; KRAJCAR, 2011).

Outros autores como Castro, Jara e Skarmeta (2013) analisam o impacto da iluminação que é responsável por 19% do uso global de energia elétrica e por cerca de 6% das emissões totais de gases que provocam o efeito estufa. Os autores apontam que sistemas de iluminação inteligente poderiam, através da integração com um amplo conjunto de sensores e de controladores com tecnologias de informação, alcançar uma maior eficiência e um menor impacto negativo para o usuário final.

O objetivo final de todas as pesquisas envolvendo energia, no âmbito de cidades inteligentes ou construções inteligentes é uma redução significativa das emissões de poluentes e a adoção de estratégias adequadas para reduzir os desperdícios de energia. De Paola et al. (2014) aponta que simplesmente limitar a utilização de novos serviços seria inaceitável para o usuário final, assim, em vez de serviços de corte, a investigação no domínio da eficiência energética deve centrar-se na otimização da utilização dos recursos proporcionando um nível adequado de conforto para os usuários.

2.4 Considerações Parciais

Este capítulo apresentou conceitos pertinentes para a compreensão do restante do trabalho. Estão presentes conceitos sobre cada um dos três níveis que compõe o modelo

ElCity. Primeiramente são apresentados conceitos de computação ubíqua, contexto, bem como problemas relacionados ao consumo de energia em dispositivos móveis, os quais são fundamentais para o nível do cidadão. A seguir, é apresentada uma seção de computação em nuvem, um item fundamental para cidades inteligentes, onde ocorrerá armazenamento e processamento dos dados gerados pelas cidades e seus cidadãos. Por fim, é apresentada a definição e classificação de cidades inteligentes que é seguida, bem como planejamento urbano e algumas melhorias possíveis quando empregado em cidades inteligentes, além de uma subseção apresentando aspectos do consumo de energia em cidades inteligentes. Estes conceitos compõem a base necessária para a compreensão das tecnologias e decisões do modelo proposto.

3 TRABALHOS RELACIONADOS

Este capítulo tem como objetivo apresentar os trabalhos relacionados ao modelo proposto. Nesta análise foram incluídos trabalhos que levam em conta a economia de energia em dispositivos móveis para obtenção de contexto ou localização, economia de energia em computação em nuvem através da elasticidade e economia de energia em cidades inteligentes através do processamento de grandes massas de dados de contexto de usuários.

Os critérios utilizados na seleção são especificados na seção 3.1. O restante do capítulo apresenta os trabalhos divididos em três seções: Seção 3.2 Dispositivos Móveis; Seção 3.3 Computação em Nuvem; e Seção 3.4 Cidades Inteligentes. Por fim, estão presentes as oportunidades de pesquisa e considerações parciais na Seção 3.5.

3.1 Metodologia de Pesquisa e Escolha dos Trabalhos Relacionados

A pesquisa dos trabalhos relacionados foi realizada utilizando o portal Google Scholar, por indexar diversos periódicos como IEEE-Xplorer, ACM - Digital Library e o Portal de Periódicos CAPES. Em uma primeira etapa foi realizada uma busca envolvendo critérios de economia de energia multinível em cidades inteligentes, no entanto nenhum trabalho envolvendo estes critérios foi encontrado, apenas trabalhos que envolvem apenas uma das áreas. Com base nesta pesquisa optou-se pela divisão em três grupos (Dispositivos Móveis, Computação em Nuvem e Cidades Inteligentes), citados anteriormente, sendo que cada um dos grupos foi combinado com os critérios abaixo, a fim de filtrar trabalhos relacionados ao consumo de energia, sensibilidade a contexto e planejamento urbano, o mais próximo possível de cidades inteligentes. Alguns trabalhos, como o de Perera et al. (2014), podem envolver mais de uma área, no entanto possuem maior ênfase no nível em que foram alocados.

- **(i) Dispositivos Móveis:** *Power/Energy Consumption/Saving, Context Sensing/Monitoring/Aware, Smart Cities;*
- **(ii) Computação em Nuvem:** *Smart Cities, Energy/Power Consumption/Saving;*
- **(iii) Cidades Inteligentes:** *Power/Energy Consumption/Saving, Urban Planning, e-Governance.*

3.2 Dispositivos Móveis

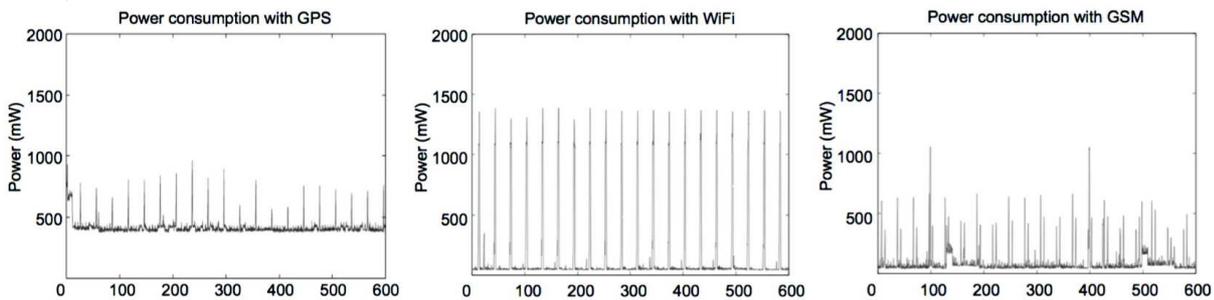
Esta seção tem como objetivo apresentar os trabalhos relacionados a obtenção de contexto utilizando dispositivos móveis com base nos critérios apresentados no início deste capítulo.

3.2.1 EnLoc

Constandache et al. (2009) propõe EnLoc, um sistema de localização energeticamente eficiente para dispositivos móveis. Segundo os autores, a maioria dos aplicativos baseados em localização utilizam recursos de GPS. Infelizmente, GPS possui um custo de energia inaceitável que pode reduzir a vida útil da bateria do telefone para menos de nove horas (tendo como base o *smartphone* Nokia N95 utilizado no trabalho). A proposta tem como base a utilização também de WiFi ou GSM a fim de melhorar a vida da bateria sem penalizar a precisão da localização.

Como contribuição o trabalho quantifica este importante *trade-off* entre precisão da localização e consumo de bateria. O *framework* desenvolvido utiliza uma solução de programação dinâmica para otimizar este *trade-off* tendo como base o percentual de bateria restante. A solução tem como entrada um rastreamento do usuário (leitura de GPS, WiFi e GSM em diversos pontos durante o percurso do usuário) e gera uma programação de leitura do sensor a fim de reduzir o consumo de bateria e o erro médio de localização. Para tal, os autores utilizam traços de consumo de diversos sensores apresentados na Figura 7.

Figura 7: Consumo de energia dos diversos sensores de localização disponíveis (GPS, WiFi e GSM).



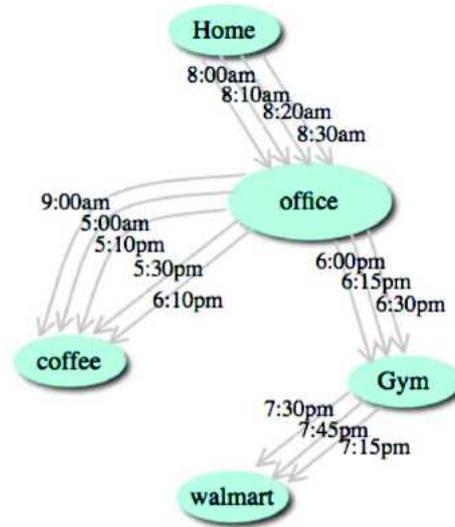
Fonte: obtida em (CONSTANDACHE et al., 2009).

Outra grande oportunidade explorada pelo EnLoc é a previsão do comportamento humano. Seres humanos possuem padrões de atividade habitual, e a amostragem da atividade em alguns pontos pode ser suficiente para prever o restante através de uma árvore lógica de mobilidade, como a apresentada na Figura 8.

Para avaliar a proposta, os autores utilizam um perfil de mobilidade de uma pessoa coletado durante 30 dias, e geram a árvore lógica de mobilidade de forma manual, tendo como resultado a utilização de apenas 25% da bateria durante 24 horas obtendo uma média de erro de localização de 12 metros, bem próximo dos 10 metros obtidos com GPS, e energeticamente mais eficiente.

Usuários podem desviar-se de seus caminhos habituais, mesmo que desvios não sejam um caso comum, eles são importantes. Os autores ressaltam algumas limitações, como a do EnLoc não detectar proativamente os desvios de caminhos, e a geração manual da

Figura 8: Árvore lógica de mobilidade LMT (*logical mobility tree*).



Fonte: obtida em (CONSTANDACHE et al., 2009).

árvore lógica de mobilidade. Ainda pode ser ressaltado que os dados de contexto do usuário não são compartilhados e todo o processamento é feito no próprio *smartphone*.

3.2.2 ACE

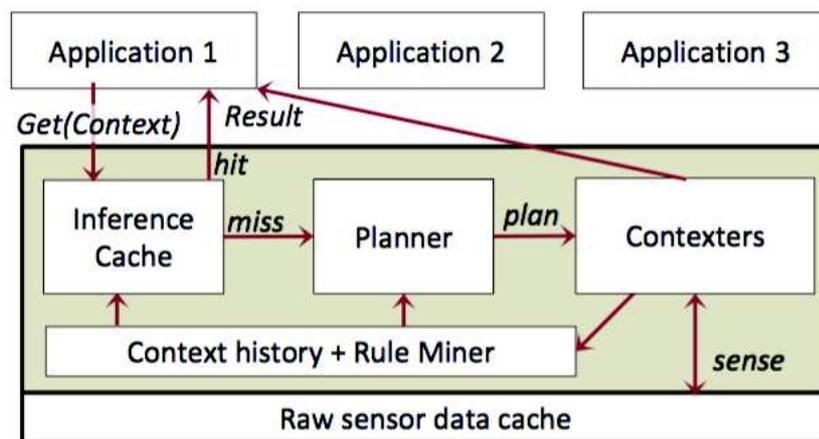
Nath (2012) propõe ACE (*Acquisitional Context Engine*), um *middleware* que suporta aplicações sensíveis a contexto enquanto reduz os custos de energia para inferir contextos. ACE fornece aos usuários o contexto atual para aplicativos executados dentro de um *smartphone*, além de aprender de forma dinâmica as relações entre os vários atributos de um contexto (por exemplo, sempre que o usuário está dirigindo, ele não está em casa). ACE explora essas relações automaticamente aprendidas para: (i) inferir oportunisticamente um atributo de contexto (em casa) de outro atributo já conhecido (dirigindo), sem adquirir dados do sensor; e (ii) detecção especulativa que permite ocasionalmente inferir o valor de um atributo com base em outro de custo menor, como determinar que o usuário não está em casa, através da detecção de atributos que determinam que ele está dirigindo.

Com relação ao fornecimento de informações de contexto para aplicativos, ACE pode alcançar economia de energia significativa. Com três aplicações e traços reais de contexto de 105 usuários, ACE foi capaz de reduzir os custos de detecção em 4,2 vezes quando comparado com um *cache* compartilhado de contexto. Esta economia ocorre utilizando técnicas para correlacionar atributos de contextos, permitindo inferir um atributo contexto desconhecido a partir de um atributo de contexto conhecido. O erro do algoritmo de inferência de contexto utilizado pelos autores ficou abaixo de 4%.

A Figura 9 mostra uma arquitetura de alto nível do ACE. Ele funciona como uma

biblioteca de *middleware* entre aplicações e sensores. Um aplicativo interage com ACE com uma API pedindo o valor atual de um sensor qualquer que ficará armazenado em *cache*. ACE é composto de: (i) *Contexters* que determinam o valor atual de um atributo de contexto através da aquisição de dados utilizando algoritmos de inferência; (ii) *cache* de dados do sensor; (3) *Rule Miner* responsável por aprender regras de contexto e relações entre os atributos de vários contextos; (iv) *Inference Cache* implementa uma memória inteligente dos sensores; (v) *Planner Sensing* responsável por tratar erros de *cache* encontrando a sequência de atributos mais “baratos” a fim de determinar o valor do atributo com problema.

Figura 9: Arquitetura do ACE.



Fonte: obtida em (NATH, 2012).

3.2.3 SmartDC

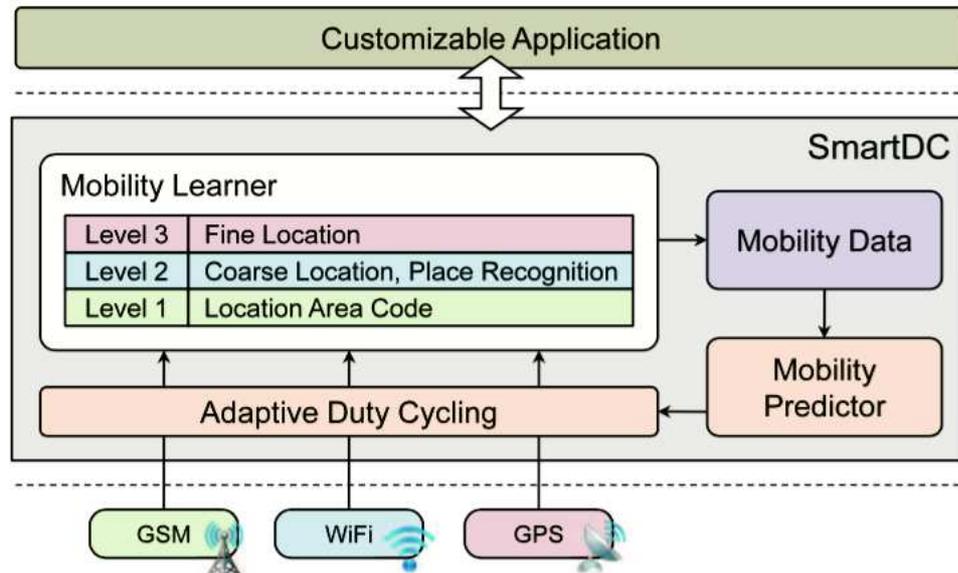
Chon et al. (2014) propõe SmartDC, um previsor de mobilidade adaptativo para monitoramento de localização diária. O objetivo do trabalho é um framework que fornece continuamente dados de localização com consumo mínimo de energia, buscando identificar lugares significativos e prever os locais onde o usuário estará, uma vez que, de acordo com os autores humanos são criaturas de hábitos “tempo-resolvidos” com pequena variação.

Com relação ao primeiro caso, quando um usuário permanece em um lugar por um determinado período de tempo, SmartDC passa a considerá-lo um lugar significativo e gera um nó com localização, conectividade, o tempo de visita e tempo de residência. O problema-chave está em escolher um intervalo de detecção de localização ideal.

Além disso, SmartDC organiza a ordem de ativação dos sensores a fim de minimizar o uso de sensores com consumo intenso de energia. A ideia básica é que a detecção de grão fino seja ativada somente se a detecção de grão grosso não conseguir obter informações precisas. Com relação a previsão de localização os autores utilizam dois tipos de previsores: (i) Cadeias de Markov como um previsor dependente da localização e não linear, e (ii)

Séries temporais como um previsor independente de localização. A arquitetura do sistema proposto está apresentada na Figura 10.

Figura 10: Arquitetura do SmartDC.



Fonte: obtida em (CHON et al., 2014).

Foram coletados os traços reais de 57 usuários ao longo de quatro semanas com SmartDC executando como um serviço de coleta automática. Com relação ao uso de sensores, o sistema proposto consome 81% menos energia do que os sistemas de sensoria-mento periódicos, e 87% menos energia do que a obtenção de dados de contexto utilizada normalmente, mas impondo um atraso de detecção de cerca de 160 segundos.

3.3 Computação em Nuvem

Esta seção tem como objetivo apresentar os trabalhos relacionados a computação em nuvem no âmbito de cidades inteligentes com base nos critérios apresentados no início deste capítulo.

3.3.1 Scallop4SC

Yamamoto, Matsumoto e Nakamura (2012) propõem Scallop4SC, uma plataforma de dados para armazenar e processar dados de serviços de cidades inteligentes. A plataforma se concentra especialmente em dados de casas inteligentes.

Os autores ressaltam dois desafios, o primeiro é como gerenciar dados de casas em larga escala, especialmente o log que tende a tomar grandes proporções, uma vez que dados são coletados periodicamente a partir de vários tipos de aparelhos e sensores. O segundo desafio é a forma de modelar estes dados. Para resolver o primeiro é proposta uma

plataforma de dados escalável para armazenar e processar os dados utilizando Hadoop/-MapReduce. Com relação ao segundo desafio os autores propõem modelos de configuração e log utilizando SQL (Structured Query Language) e HBase.

A Figura 11 apresenta a arquitetura geral do Scallop4SC. Em cada casa inteligente os logs são gerados por um *logger* e enviados para o Scallop4SC. O log da casa é então armazenado, e apenas se necessário, para algum serviços o log é analisado utilizando MapReduce. Os dados de configuração da casa são armazenados em bancos de dados SQL e são fornecidos para serviços externos e aplicações através de uma API.

Os autores implementaram um protótipo com 12 servidores e realizaram uma avaliação experimental calculando o consumo de energia de uma casa, utilizando um log real gravado por um ano em uma casa inteligente com 30 aparelhos. Outra simulação levou em conta 700.000 residências da cidade de Kobe no Japão. A simulação do processamento dos dados estimou que seriam necessárias 12 horas para que Scallop4SC processasse os logs.

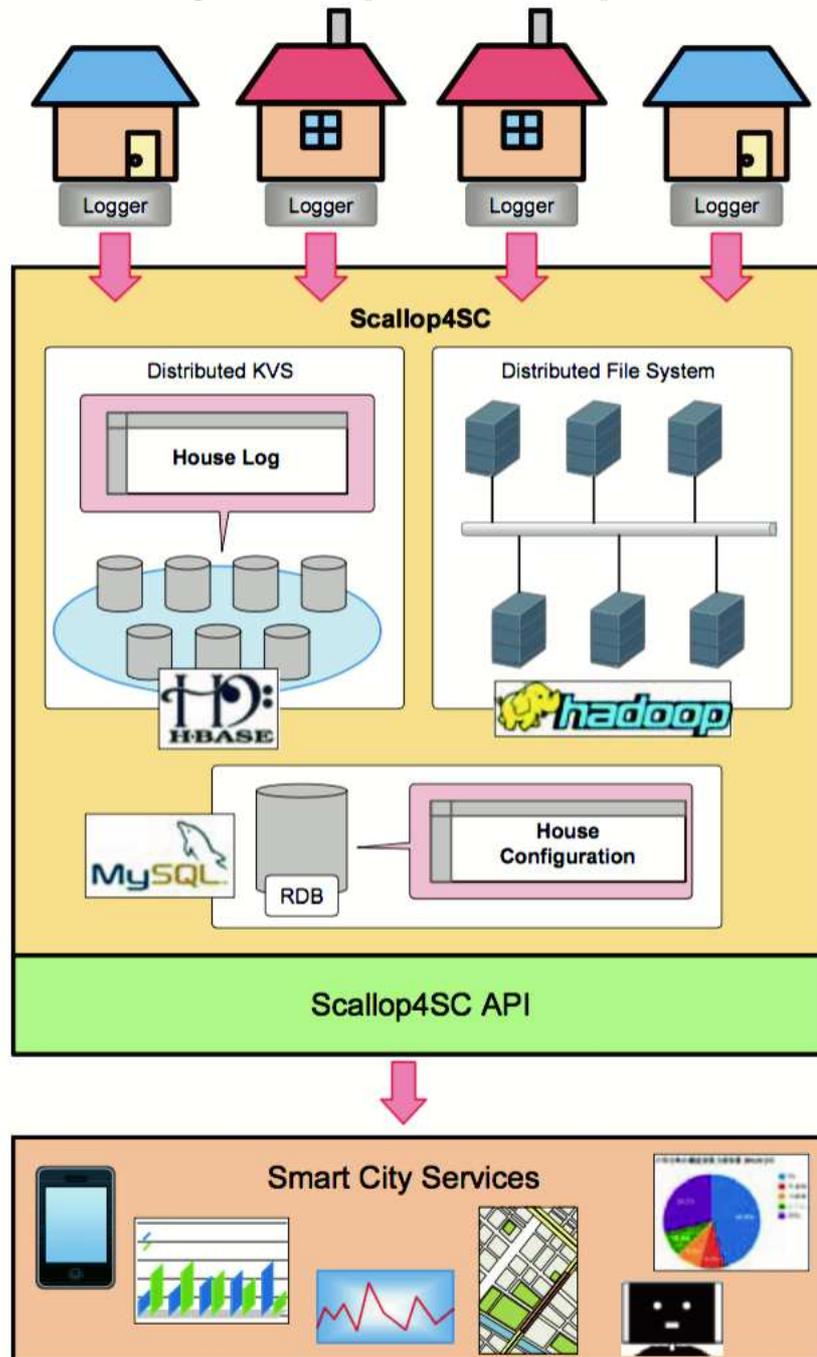
3.3.2 A Cloud-based Architecture for Citizen Services in Smart Cities

Khan e Kiani (2012) apresentam as cidades inteligentes e as interações entre as suas diferentes aplicações para gerar informação inteligente para governança urbana. Além disso, propõem que a computação em nuvem pode fornecer uma infraestrutura computacional adequada para armazenamento de dados e necessidades de processamento de cidades inteligentes. Os autores enfatizam que o usuário além de prover dados de seu ambiente para a cidade deve ter a possibilidade de acessar informações de contexto da cidade inteligente através de um sistema integrado de informações.

As informações apresentadas ao usuário devem ter como base suas preferências, e já devem ter passado por etapas de processamento adicional para a preparação de dados contextuais e visualização da informação. Para tal, os autores propõem a arquitetura em camadas disposta na Figura 12. A arquitetura representada consiste de cinco camadas horizontais e duas camadas verticais. A saída das duas primeiras camadas inferiores é genérica e pode ser adaptada para as necessidades das aplicações nas três camadas acima. Um dos princípios da arquitetura é introduzir componentes sensíveis a contexto em diferentes camadas da arquitetura a fim de coordenar adequadamente o fluxo de informações evitando a captura de dados desnecessários, manuseio e processamento. As camadas são explicadas separadamente abaixo:

- (i) A camada de integração de plataforma retrata uma infraestrutura baseada em um ambiente de nuvem híbrida que garante a acessibilidade multi-plataforma aos dados;
- (ii) A camada de aquisição e análise de dados é utilizada para acessar os dados ambientais de várias fontes incluindo banco de dados remotos;

Figura 11: Arquitetura do Scallop4SC.



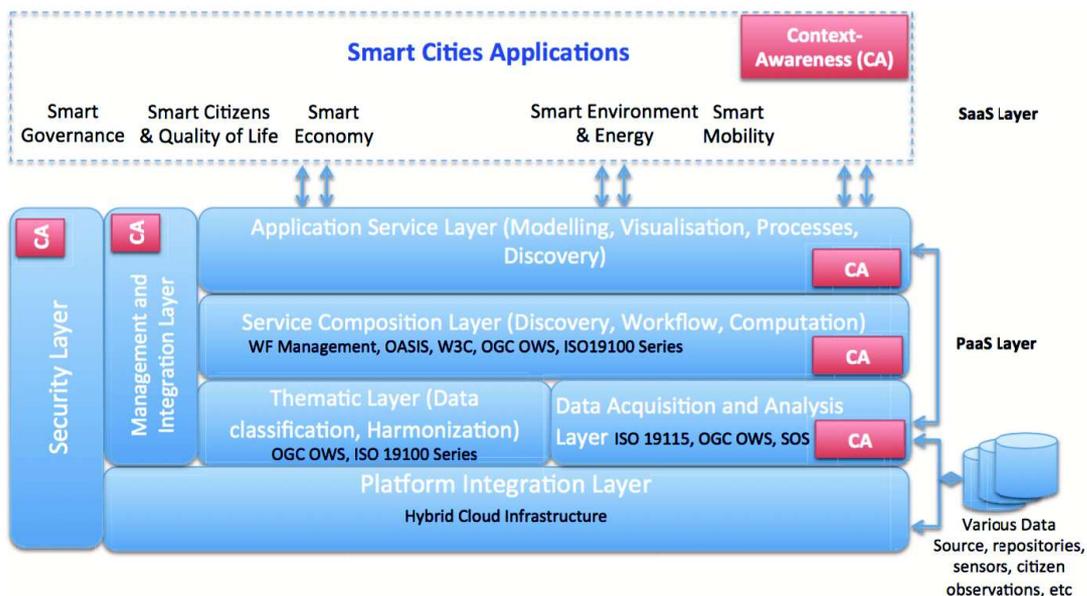
Fonte: obtida em (YAMAMOTO; MATSUMOTO; NAKAMURA, 2012).

- (iii) A camada temática classifica os dados adquiridos em categorias específicas e executa harmonização e atualização dos dados para utilização posterior;
- (iv) A camada de composição de serviços é necessária para projetar fluxos de trabalho, identificar fontes de dados e conectar componentes de processamento;
- (v) A camada de serviços de aplicação usa os resultados da camada de composição de serviços em ferramentas específicas, tais como simulações e mapas para realizar

a análise contextual e a tomada de decisão;

- (vi) A camada de gestão e integração é usada para automatizar o fluxo de dados filtrados e informação entre as camadas horizontais;
- (vii) A camada de segurança cuida da autenticação, autorização e auditoria para a utilização de dados e serviços.

Figura 12: Arquitetura propostas por Khan e Kiani (2012).



Fonte: obtida em (KHAN; KIANI, 2012).

3.3.3 Renewable Energy-aware Data Center Operations for Smart Cities The DC4Cities Approach

Klingert et al. (2015) apresenta um modelo de um gerenciador de *data centers* ciente de energia para cidades inteligentes chamado DC4Cities, apresentado na Figura 13. Segundo os autores o grande desafio das cidades inteligentes, que funcionam em cima de grandes massas de dados, é que todos os dados gerados por uma infinidade de sensores precisam ser coletados, processados e disponibilizados por *data centers*.

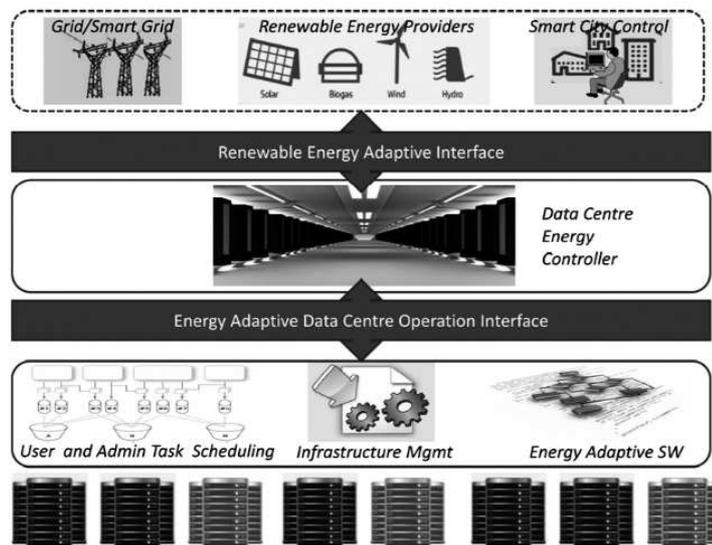
O modelo permite comunicação quase em tempo real entre a cidade inteligente, todos os *data centers* participantes e clientes específicos do *data center*. A cidade inteligente representa uma “Autoridade de Gestão de Energia” e atua como um mediador entre o *data center* e o sistema de energia supervisionando a disponibilidade de recursos energéticos renováveis, ou não, e a partir destes dados calcula um plano de energia ideal.

Para a redução do consumo do *data center* os autores realizam uma reorganização da carga de trabalho a fim de coincidir com o momento onde há uma maior oferta de

energia de fontes renováveis. O resultado esperado é que o consumo de energia real de todo o *data center* seja consideravelmente mais próximo de um plano de energia ideal previamente calculado, satisfazendo assim os objetivos de potência recebidos pela cidade inteligente. Os autores ainda propõem uma API em nível de IaaS, PaaS e aplicações individuais, capaz de permitir que cada nível do *data center* siga as diretivas de energia.

Klingert et al. (2015) apresenta uma simulação do modelo proposto. O modelo considera uma aplicação *batch* que produz 4320 relatórios por dia. Para avaliar os autores definem a métrica RenPercent, que refere-se ao percentual de utilização de energia renovável. Os resultados obtidos indicam que em uma situação real, o RenPercent esperado seja entre 42,20% e 79,41%.

Figura 13: Arquitetura de alto nível do DC4Cities.



Fonte: obtida em (KLINGERT et al., 2015).

3.4 Cidades Inteligentes

Esta seção tem como objetivo apresentar os trabalhos relacionados a cidades inteligentes e planejamento urbano, com base nos critérios apresentados no início deste capítulo.

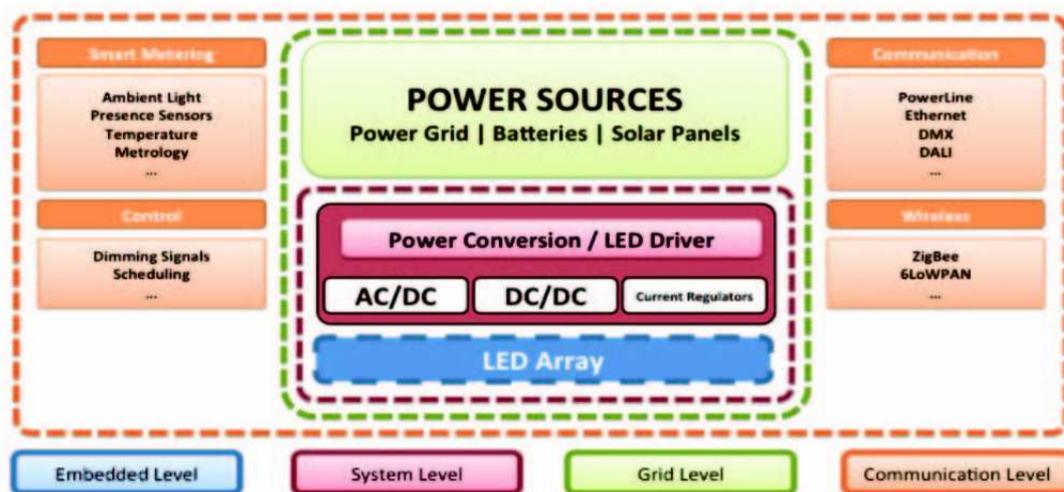
3.4.1 Smart Lighting solutions for Smart Cities

Segundo Castro, Jara e Skarmeta (2013) as cidades inteligentes são o meio de atingir sustentabilidade tanto em nível de infraestrutura, como em desenvolvimento sustentável industrial e residencial. Para os autores, o primeiro passo rumo a este objetivo é reduzir o consumo de energia e os níveis de gases do efeito estufa, e para tal, os autores propõem uma solução de iluminação para cidades inteligentes. Este sistema compreende uma área heterogênea e multi-disciplinar no âmbito da gestão de iluminação, com a possibilidade

de integrar um amplo conjunto de tecnologias de sensores e de controles, em conjunto com as tecnologias de informação e comunicação, com o objetivo de alcançar uma maior eficiência nos domínios comercial e público.

O sistema proposto compreende a integração de funcionalidades e interfaces inteligentes em quatro níveis complementares apresentados na Figura 14. (i) O primeiro nível de integração é no mecanismo de iluminação ou na própria fonte de luz; (ii) O segundo nível é em luminárias e sistemas de iluminação; (iii) O terceiro nível compreende gestão e monitoramento de fontes de energia, usinas e equipamentos de distribuição; (iv) O quarto nível compreende soluções de iluminação completas, com aplicações de monitoramento, controle e gestão.

Figura 14: Níveis de integração do sistema de iluminação inteligente.



Fonte: obtida em (CASTRO; JARA; SKARMETA, 2013).

O sistema proposto fornece recursos para operar automaticamente um dispositivo ou um conjunto deles em um amplo conjunto de situações:

- (i) Cronológica e agendamento astronômico: por programação cronológica e astronômica é possível definir temporizadores para as luzes durante um período de tempo. Também inclui a capacidade de modificar a operação do sistema dependendo do horário de nascer e pôr do sol;
- (ii) Comportamento ambiental e humano: adaptando o comportamento do sistema de acordo com o comportamento das entidades monitoradas e variáveis de interesse;
- (iii) Acontecimentos concretos de programação, é possível utilizar configurações especiais para determinadas situações, tais como festas, noites escuras, ou datas em que é necessário algum tipo de iluminação especial;
- (iv) Condições, ou alarme: segurança poderia ser melhorada através integração deste tipo de sistema inteligente, que pode incluir operações como abrir portas e iluminação

pela detecção de alguma situação de alerta;

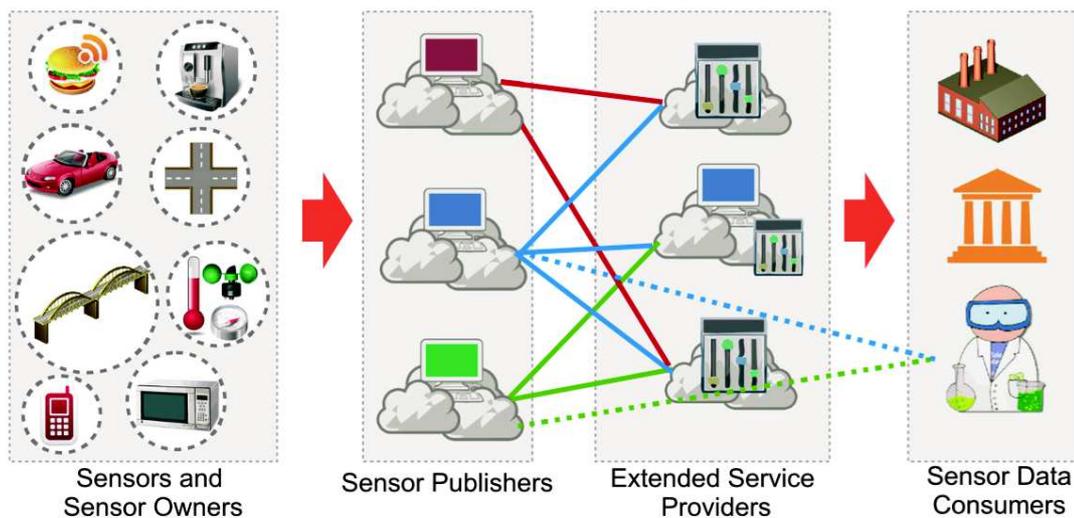
- (v) Programa complexo ou inferência: configurações baseadas em eventos e reconhecimento de padrões.

O sistema de iluminação inteligente proposto apresenta uma redução de 50% a 70% quando comparado com tecnologias convencionais, tanto no consumo de energia quanto nas emissões de gases do efeito estufa.

3.4.2 Sensing as a Service Model for Smart Cities Supported by Internet of Things

Perera et al. (2014) propõe um modelo de sensores como serviço para cidades inteligentes suportado pela Internet das Coisas (IoT). O modelo, apresentado na Figura 15, é constituído por quatro camadas: (i) os sensores e os proprietários do sensor; (ii) os editores de sensores; (iii) os prestadores de serviços estendidos; e (iv) os consumidores de dados do sensor.

Figura 15: Arquitetura do Sensing as a Service.



Fonte: obtida em (PERERA et al., 2014).

A primeira camada, composta por sensores e proprietários dos sensores, tem como objetivo capturar informação que pode ser usada para entender o comportamento de usuários e suas preferências, ou ainda pode detectar as condições meteorológicas e de tráfego. A segunda camada tem como principal função detectar sensores disponíveis, comunicar com os proprietários dos sensores e obter permissão para publicar os dados dos sensores na nuvem. A terceira camada pode ser considerada como a mais inteligente entre todas as quatro camadas, devendo fornecer serviços que agreguem valor para os consumidores de dados de sensores. Por fim, a quarta camada é composta por todos os consumidores de dados de sensores. Estes consumidores precisam registrar-se e obter um certificado digital válido de uma autoridade, a fim de consumir dados dos sensores.

Os autores discutem o modelo de três perspectivas diferentes: (i) econômica, (ii) tecnológica e (iii) social, analisando características como sustentabilidade e escalabilidade, apresentando diversos casos de uso em cidades inteligentes. O modelo também permite utilizar eficientemente recursos e acomodando um grande número de consumidores, contribuindo significativamente para enfrentar desafios das cidades inteligente.

3.4.3 An urban control center for the energy governance of a smart city

Carli et al. (2014) aborda a necessidade de ferramentas de planejamento urbano que forneçam controle de energia em cidades inteligentes. Os autores apresentam a arquitetura de um sistema de apoio à decisão chamado Urban Control Center (UCC). UCC mede o desempenho energético da cidade e apoia o tomador de decisão na determinação de um plano de ação. O UCC pode ser usado, por exemplo, pela administração pública para analisar o mapa energético da cidade e pelos cidadãos para obter informações sobre o uso de energia e como reduzir os impactos ambientais.

O artigo apresenta a arquitetura UCC e as principais ferramentas de avaliação de desempenho da cidade, a coleta de dados, análise e elaboração de relatórios. É proposto um modelo de tomada de decisão com base em um modelo de programação de dois níveis e integração de diferentes unidades de tomada de decisão (painéis de decisão). Cada painel é dedicado à governança estratégica de energia de um subsistema específico de energia urbana (edifícios privados, edifícios públicos, iluminação pública da cidade, etc).

Figura 16: Arquitetura de alto nível do UCC.



Fonte: obtida em (CARLI et al., 2014).

O modelo integra sistemas diferentes, proporcionando uma visão completa da situação energética da cidade, podendo gerar alertas, lançar fluxos de trabalho, além de fornecer uma análise situacional em tempo real, e acompanha questões diárias. O núcleo do modelo UCC, responsável pelos dados dos vários sistemas urbanos, é apresentado na Figura 16. A esquerda da Figura está presente a ferramenta de avaliação de desempenho e a direita,

a ferramenta de apoio à decisão e gestão estratégica da cidade inteligente.

3.5 Análise e Oportunidades de Pesquisa

Os trabalhos apresentados foram avaliados de acordo com seis quesitos definidos como desejáveis em um modelo sustentável de gerência de cidades inteligentes. Os quesitos avaliados estão descritos abaixo e dispostos na Tabela 2 a fim de classificar e facilitar a análise dos trabalhos apresentados.

- (i) **Contexto:** O trabalho leva em conta dados de contexto de usuários, a fim de possibilitar a gerência ou melhoria da eficiência energética com base nos usuários e suas atividades?
- (ii) **Energia:** O trabalho realiza algum tipo de análise ou melhoria no consumo de energia, ou eficiência energética?
- (iii) **Dados:** O trabalho proposto realiza algum tipo de análise ou agregação dos dados coletados a fim de disponibilizá-los em uma plataforma para que sejam utilizados em algum tipo de tomada de decisão?
- (iv) **Elasticidade:** Os recursos do trabalho proposto adaptam-se de acordo com a variação de demanda?
- (v) **Governança:** O trabalho visa a melhoria das condições da cidade ou possui foco em auxiliar na governança de cidades inteligentes?
- (vi) **Multinível:** O trabalho integra os níveis cidadão, nuvem e dispositivos da cidade para oferecer melhorias no âmbito de cidades inteligentes?

Tabela 2: Avaliação dos trabalhos relacionados com base nos quesitos definidos.

Trabalho	Contexto	Energia	Dados	Elasticidade	Governança	Multinível
Constandache et al. (2009)	✓	✓		✓		
Nath (2012)	✓	✓		✓		
Chon et al. (2014)	✓	✓		✓		
Yamamoto, Matsumoto e Nakamura (2012)		✓	✓	✓	✓	
Khan e Kiani (2012)	✓		✓	✓	✓	
Klingert et al. (2015)		✓	✓	✓		
Castro, Jara e Skarmeta (2013)	✓	✓		✓		
Perera et al. (2014)	✓		✓	✓		
Carli et al. (2014)		✓	✓		✓	

Fonte: elaborado pelo autor.

Enloc (proposto por Constandache et al. (2009)), ACE (proposto por Nath (2012)) e SmartDC (proposto por Chon et al. (2014)) apresentam um *framework* para coleta de dados e contexto através dos *smartphones* dos usuários, além de adaptarem as configurações dos sensores para reduzir o consumo de energia, e consequentemente reduzindo a necessidade de recarga. Se analisados de forma conjunta, com outros dispositivos da cidade, estes dados poderiam ser úteis na gerência de cidades inteligentes. Entretanto isto não é feito, nem os dados são disponibilizados para que um serviço terceiro realize esta análise.

Yamamoto, Matsumoto e Nakamura (2012) propõem Scallop4SC que consiste em uma plataforma para o armazenamento e processamento de dados de serviços da cidade inteligentes. No entanto, este trabalho tem um foco particular em casas inteligentes (*smart homes*) não fornecendo uma visão geral da cidade nem coletando dados de contexto usuários, de modo a contribuir apenas de forma limitada para o planejamento urbano. A arquitetura proposta por (KHAN; KIANI, 2012) tem ênfase na arquitetura de nuvem para cidades inteligentes, mas desconsidera totalmente o consumo de energia e a sustentabilidade. Já o modelo de nuvem proposto por Klingert et al. (2015) (DC4cities) possui foco em energia, sustentabilidade e elasticidade do *data center*, mas apesar de possuir um componente chamado “*smart city control*” ele refere-se apenas ao controle dos *data centers* e suas fontes de energia, não da cidade como um todo.

Castro, Jara e Skarmeta (2013) propõe Smart Lightning, que apresenta uma das possíveis formas de reduzir o consumo de energia em cidades inteligentes através da utilização de dados de contexto de usuários. Entretanto, é uma solução pontual, que não considera governança nem disponibilização dos dados coletados. Sensing as a Service proposto por Perera et al. (2014) utiliza a computação em nuvem para disponibilizar os dados de contexto de usuários, além de possibilitar entradas de grandes volumes de dados e diversas formas de compartilhamento, mas não leva em consideração o consumo de energia/sustentabilidade do sistema, e não visa a melhoria na governança da cidade inteligente. UCC proposto por Carli et al. (2014) apresenta ferramentas de gestão da cidade e energia, com foco na análise de dados e gerenciamento das fontes de energia, mas sem considerar dados de contexto dos usuários da cidade, nem a adaptação do sistema de acordo com a demanda.

Considerando os trabalhos analisados, é possível encontrar soluções que focam em obtenção de contexto dos usuários com um reduzido consumo energético (CONSTANDACHE et al., 2009; NATH, 2012; CHON et al., 2014) mas desconsiderando a utilização destes dados para auxiliar a tomada de decisões ou melhoria da qualidade de vida nas cidades inteligentes. Outros trabalhos propõem estruturas de computação em nuvem para cidades inteligentes, no entanto com foco específico na infraestrutura, desconsiderando os usuários (YAMAMOTO; MATSUMOTO; NAKAMURA, 2012; KLINGERT et al., 2015; KHAN; KIANI, 2012). Por fim, foram analisados trabalhos com ênfase em redução de

consumo e gerência para cidades inteligentes (CASTRO; JARA; SKARMETA, 2013; PERERA et al., 2014; CARLI et al., 2014), mas que em alguns casos negligenciam o consumo de energia, sustentabilidade, escalabilidade e dados de contexto dos usuários. Com base nas limitações encontradas é possível identificar a oportunidade para **um modelo elástico de economia de energia multinível (cidadão, computação em nuvem e cidade) para cidades inteligentes**.

Se observados separadamente, cada nível possui uma capacidade limitada de economia de energia, no entanto, se forem combinados acredita-se que um potencial maior de economia de energia pode ser obtido. Assim, esta pesquisa tem como objetivo propor um modelo de gerência de energia multinível, que coleta dados de contexto dos usuários, torna-os disponíveis para serem processados na nuvem e usa os resultados obtidos para propor um plano de redução do consumo de energia na cidade, sem afetar os serviços disponíveis, ou seja, sem alterar a percepção/qualidade oferecida aos cidadãos.

3.6 Considerações Parciais

Este capítulo apresentou um levantamento de trabalhos relacionados à questão de pesquisa desta dissertação. Considerando os trabalhos analisados, é possível encontrar soluções pontuais em cada um dos níveis contemplados pelo modelo ElCity. No nível de usuários encontra-se trabalhos relacionados a obtenção de dados de contexto e análise da mobilidade dos cidadãos com reduzido consumo de energia (CONSTANDACHE et al., 2009; NATH, 2012; CHON et al., 2014). No nível do *data center* observou-se trabalhos que propõem plataformas de armazenamento e processamento de dados, além de diversos serviços para cidades inteligentes (YAMAMOTO; MATSUMOTO; NAKAMURA, 2012; KLINGERT et al., 2015; KHAN; KIANI, 2012). Já no nível das cidades inteligentes, alguns trabalhos abordam a redução de consumo de energia, contexto dos usuários e gerenciamento de fontes de energia (CASTRO; JARA; SKARMETA, 2013; PERERA et al., 2014; CARLI et al., 2014). Entretanto, em uma busca extensiva, não foi possível encontrar trabalhos que propusessem integração dos níveis, propondo melhoras na sustentabilidade ou economia de energia em cidades inteligentes. Desta forma, identificou-se a oportunidade para um modelo elástico de economia multinível (cidadão, computação em nuvem e dispositivos da cidade) de energia para cidades inteligentes.

4 MODELO ELCITY

Este capítulo descreve o modelo ElCity, um modelo elástico de economia multinível de energia para cidades inteligentes. Para facilitar a apresentação e compreensão do modelo este capítulo está dividido em 7 seções. A Seção 4.1 apresenta as decisões de projeto, seguida da arquitetura na Seção 4.2, os detalhes das decisões de projeto nos diferentes níveis na Seção 4.3, o monitor de energia na Seção 4.4, o gerenciadores de elasticidade na Seção 4.5, a aplicação de recomendação de economia de energia na Seção 4.6 e por fim as considerações parciais do modelo na Seção 4.7.

4.1 Decisões de Projeto

O modelo ElCity busca combinar os dados dos cidadãos e dispositivos da cidade para serem processados em um *data center*, a fim de gerar planos de economia de energia sem afetar os serviços já oferecidos na cidade. A ideia principal consiste em estender o conceito de elasticidade, largamente utilizado no âmbito de computação em nuvem, neste caso para gerenciar o consumo de energia em cidades inteligentes, de acordo com a demanda em relação ao uso dos recursos. Em primeiro lugar, os recursos provenientes de fontes diferentes de energia (smartphones do cidadão, dispositivos da cidade e nós de um *data center* baseado em nuvem) podem ser combinados para posteriormente criar planos a fim de ligar e desligar recursos de cada um dos consumidores de energia mencionados. Para o desenvolvimento do modelo ElCity foram consideradas as seguintes decisões de projeto:

- (i) No nível dos cidadãos (também apresentados como usuários neste trabalho), optou-se pela utilização de smartphones para coletar dados de contexto (em particular, a localização do usuário) devido ao seu forte acoplamento ao dia a dia das pessoas (CHON et al., 2014), além de possuir uma diversidade de sensores, ampliando o leque de opções de conectividade;
- (ii) Considerando o armazenamento e processamento de dados, optou-se pela utilização de computação em nuvem, uma vez que esta plataforma oferece a capacidade de elasticidade, adaptando automaticamente sua capacidade de armazenamento e processamento de dados de acordo com as necessidades do usuário e cidade;
- (iii) A administração pública fornece uma infraestrutura de nuvem para armazenamento e processamento de dados da cidade e usuários;
- (iv) O modelo utiliza elasticidade reativa (RIGHI et al., 2015), que baseia-se em limites (*thresholds*) inferiores e superiores de carga de CPU, onde a gestão de recursos ocorre quando um destes limites é ultrapassado;

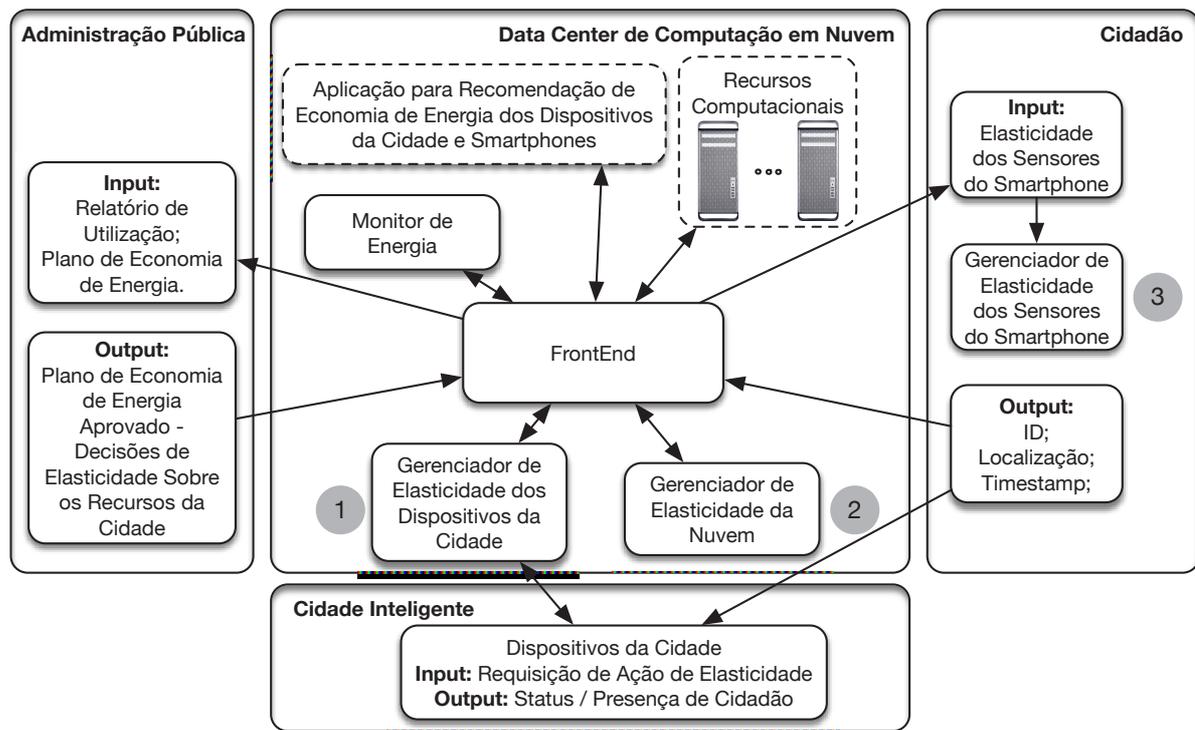
- (v) O modelo deve ser genérico o suficiente para cobrir o maior número possível de locais e dispositivos da cidade, embora o foco da avaliação neste momento seja em iluminação pública como dispositivos da cidade;
- (vi) Os dispositivos da cidade atuam como “coisas” (IoT) com sensores e atuadores. Eles capturam dados de cidadãos em seu entorno, passam estes dados para a nuvem e recebem planos de economia de energia a partir dos dados enviados e dos dados fornecidos pelo cidadão;
- (vii) Utilizando a nuvem como um *middleware*, a administração pública é responsável por aprovar ou recusar os planos de economia de energia nos dispositivos da cidade.

4.2 Arquitetura

ElCity possui foco na integração de dados de várias fontes, proporcionando assim um quadro de economia de energia multinível para gerir a energia em cidades inteligentes. O modelo trabalha com três níveis de energia: (i) os usuários, ou cidadãos que utilizam smartphones equipados com sensores, que exigem bateria como fonte de alimentação (e conseqüentemente necessita de recarga); (ii) os dispositivos da cidade, em especial a iluminação pública onde as lâmpadas são comumente programadas para serem ativadas durante toda a noite, com ou sem cidadãos presentes; e (iii) *data centers* de computação em nuvem que apresentam gerenciadores de energia diferentes que analisam e combinam dados do próprio *data center*, cidadãos e dispositivos da cidade, para em seguida gerar planos de economia de energia para os diferentes níveis. A Figura 4.2 ilustra a arquitetura ElCity, com destaque aos gestores de energia.

Analisando a Figura 4.2 a partir do cidadão, é possível observar três módulos. O cidadão é responsável por fornecer sua localização, junto com um *timestamp* e seu identificador próprio para a nuvem. Esta informação pode ser repassada indiretamente através do próprio dispositivo de cidade, que detecta os utilizadores na sua área de cobertura e solicita estes dados, podendo nesse caso enviar a localização do próprio dispositivo, sem a necessidade de utilização de sensores GPS por exemplo, e transmite para a nuvem. Ou ainda, o usuário pode usar uma aplicação instalada e o sinal WiFi para enviar diretamente todos os dados históricos de uma só vez para a nuvem sempre que seu dispositivo for conectado ao carregador. De acordo com Constandache et al. (2009), os seres humanos têm padrões de atividade habituais, e a amostragem em alguns pontos pode ser suficiente para prever o restante da atividade através de uma árvore mobilidade lógica. Com os dados de localização de cidadãos na nuvem, uma aplicação para recomendação de economia de energia, que conhece todos os dispositivos da cidade, analisa os dados recebidos para definir os horários em que estes dispositivos podem ser desligados. Além disso, como retorno a nuvem também fornece aos smartphones recomendações, a fim de desativar alguns sen-

Figura 17: Arquitetura ElCity, destacando os três pontos que realizam as ações de economia de energia: (1) gerenciador responsável por ligar e desligar os dispositivos da cidade de acordo com a demanda dos cidadãos; (2) gerenciador responsável por controlar a aplicação de recomendação de economia de energia na nuvem, propondo a adição ou remoção de máquinas virtuais para apoiar a execução; (3) gerenciador responsável pela redução do consumo de energia nos smartphones ligando e desligando os sensores de acordo com a demanda e consumo de energia de cada sensor.



Fonte: elaborado pelo autor.

sores durante algumas partes do dia, sendo que essas recomendações não devem afetar a conectividade, e sim reduzir o consumo de energia. Ou seja, os sensores do smartphone ligam de forma a conectar-se aos dispositivos na cidade com o sensor de menor consumo de energia, o que afeta diretamente o consumo de bateria do smartphone.

O componente principal do modelo ElCity é o FrontEnd na nuvem, que é responsável pela gestão de todas as ligações decorrentes de módulos externos, interligando os dados recebidos com os sistemas que estão sendo executados. Ou seja, este FrontEnd que recebe os dados dos demais níveis e centraliza antes de enviar para a aplicação de recomendação de elasticidade. Após processar estes dados, um relatório de movimentação dos usuários através de vários pontos com controle (dispositivos da cidade), bem como as possíveis ações de ligar e desligar os dispositivos da cidade é enviado para a administração pública. Este relatório deve ser examinado pela administração pública que pode sugerir possíveis mudanças, retornando o plano de economia de energia para a nuvem. Por fim, a nuvem irá entrar em contato com os dispositivos da cidade para programar as ações de liga e desliga de seus atuadores nos períodos definidos. Estas ações nos dispositivos da cidade

são realizadas pelo gerente de dispositivo localizado na nuvem.

Considerando-se as descrições apresentadas, ElCity adiciona na literatura o conceito de “Economia de energia elástica multinível” para cidades inteligentes, que pode ser definida da seguinte forma:

- Definição 1: **Economia de energia elástica multinível** é uma extensão do conceito de elasticidade de nuvens computacionais (RIGHI et al., 2015) para gerenciar o consumo de energia em diferentes níveis de consumidores de energia em cidades inteligentes, de modo que os recursos em cada nível são automaticamente ligados e desligados de acordo com o padrão de uso dos cidadãos. Economia de energia elástica multinível visa gerar planos de economia de energia com base na demanda imposta sobre cada recurso, mas sempre considerando a qualidade dos serviços oferecidos atualmente pelos recursos, em cada um dos níveis considerados.

ElCity usa o termo elasticidade de maneira diferente dependendo do nível considerado: (i) no *data center*, para ampliar ou reduzir o número de nós computacionais que processam a aplicação de recomendação de economia de energia; (ii) em dispositivos da cidade, transformando o estado do que está conectado ao atuador (neste caso lâmpadas) entre ligado e desligado de acordo com a localização dos usuários; (iii) nos smartphones dos usuários, ativando e desativando sensores quando os cidadãos estão se movendo ao longo da cidade¹, agindo em conformidade com os receptores de sinal disponíveis na cidade. Por exemplo, no caso (ii) os dispositivos da cidade, se ninguém está perto de um determinado dispositivo da cidade durante a noite, ele pode ser desligado para economizar energia. Logicamente, o mesmo dispositivo deve ter a capacidade de alterar para o estado ligado, se ocorrer um evento inesperado no seu raio de cobertura. Caso o dispositivo detecte um novo usuário que não era esperado em seu raio de alcance, o atuador se manterá ligado por um intervalo de tempo definido. É importante também ressaltar que poderiam ter sido utilizados detectores de movimento independentes nos dispositivos da cidade para controlar quando ele deve ser ligado ou desligado. No entanto, optou-se por não utilizar esta abordagem, porque ela não passa pelo controle da administração pública. Entende-se que as cidades têm bairros com exigências diferentes (incluindo segurança e índices de criminalidade), por isso é importante ter planos de energia que são aprovados pela administração pública.

4.3 Detalhe das Decisões de Projeto nos Diferentes Níveis

Esta seção apresenta como cada nível trata o consumo de energia. Em cada uma das subseções a seguir é destacado um nível, sendo o primeiro os cidadãos, seguido do *data center* de computação em nuvem e por fim os dispositivos da cidade.

¹Constandache et al. (2009) afirmam que os sensores de smartphones, incluindo GPS, Bluetooth e Wi-Fi, quando ativados são os principais responsáveis pelo consumo de bateria.

4.3.1 Cidadãos

As técnicas para economia de energia relativa a utilização de sensores em smartphones encontradas nos trabalhos relacionados (CONSTANDACHE et al., 2009; NATH, 2012; CHON et al., 2014) se resumem a duas: (i) amostragem adaptativa (amostragem com uma menor frequência) e (ii) no caso de um dispositivo com vários sensores, ligá-los de forma hierárquica (dos sensores de baixa potência para os de maior potência). Como o modelo ElCity é genérico e contempla a utilização de diversos sensores, a técnica de ligação hierárquica é utilizada, ou seja, o smartphone sempre se conectará aos dispositivos utilizando o sensor de menor consumo possível.

Além disso, será utilizada uma árvore de mobilidade lógica (CONSTANDACHE et al., 2009), que leva em consideração o horário em que um cidadão costuma passar por um determinado ponto para definir o momento em que serão ativados os sensores. Estas técnicas são implementadas através do gerenciador de elasticidade dos Dispositivos Móveis conforme a Figura 4.2. A sincronização com a nuvem só ocorre quando o aparelho está ligado a uma fonte de energia, a fim de maximizar o tempo de vida útil da bateria. A localização do usuário pode ser determinada de duas formas: (i) usando sensores no smartphone como GPS ou trilateração de antenas 3G ou 4G; (ii) através da localização de dispositivos da cidade fixos que detectam um usuário que está próximo. Cada dispositivo da cidade apresenta uma coordenada de latitude e longitude que são salvas previamente na nuvem.

4.3.2 Data Center

Com relação a nuvem, são utilizadas algumas das técnicas apresentadas na taxonomia de Orgerie, Assunção e Lefevre (2014). Optou-se pela utilização de “*Cloud and Virtualization*” e “*Workload Consolidation*”, a primeira por ser a utilização da nuvem em si, que se apresenta mais econômica do que a utilização de um cluster (ROLOFF et al., 2012), por exemplo, alocando apenas os recursos necessários, além da consolidação de cargas de trabalho.

Além disto, na nuvem é utilizada elasticidade horizontal baseada em *thresholds* (limites) utilizando a métrica CPU para adicionar ou remover recursos (RIGHI et al., 2015). A utilização da métrica de CPU ocorre porque esta é apresentada como a principal responsável pelo consumo de energia em nós computacionais por vários autores (ORGERIE; ASSUNÇÃO; LEFEVRE, 2014; CHEN et al., 2014; LUO et al., 2013). O modelo também permite a definição de SLAs (*Service Level Agreement*) que podem ser utilizadas para limitar o número máximo de nós que são atribuídos num determinado período de tempo, reduzindo assim o consumo de energia oriundo dos *data centers* em horários de pico.

4.3.3 Dispositivos da Cidade

Na cidade, o foco principal deste trabalho é o consumo de energia do sistema de iluminação pública. Este sistema é atualmente responsável por 19% do consumo global de eletricidade e 6% da emissão total de gases do efeito estufa (CASTRO; JARA; SKARMETA, 2013). As medidas de economia de energia podem ser bastante diferentes dependendo dos pontos onde os dispositivos da cidade estão instalados e qual é o sensor/atuador instalado em cada dispositivo. A estratégia utilizada é a de desligar as lâmpadas em locais onde não há movimento em um determinado momento, e então ligá-las somente se um usuário estiver na área de cobertura do dispositivo da cidade determinada pela administração pública. Assim, lugares que os usuários não realizam nenhum tipo de interrupção irão economizar mais energia do que outros.

Hoje, as lâmpadas na cidade apenas apresentam foto-sensores ou relógios que determinam horários fixos em que elas permanecerão ligadas ou desligadas, independente da presença ou não de usuários. Para atingir a visão de cidade inteligente, cada lâmpada deve ter uma placa com sensores e atuadores que permita que os usuários e a nuvem conectem-se. Deve-se tomar cuidado porque este novo aparelho irá permanecer ligado 24 horas por dia, a fim de receber e fornecer dados para a nuvem. Deste modo, a economia de energia gerada pelo modelo ElCity precisa superar a energia consumida por estas placas. Por fim, a administração pública tem a responsabilidade de decidir sobre o consumo de energia no sistema de iluminação pública. Em outras palavras, embora possível, ElCity não desencadeia ações de elasticidade ao longo dos dispositivos da cidade sem antes enviar relatórios para a administração pública pedindo a sua aprovação. Isto ocorre, pois em locais com alto índice de criminalidade, por exemplo, pode ser desejável manter as luzes sempre ligadas.

4.4 Monitor de Energia

Neste trabalho está sendo proposta uma forma mais sustentável para gerir o consumo de energia em cidades inteligentes. Para conseguir isto, são necessárias formas de medir o consumo de energia, tanto em uma cidade atual como em cidades inteligentes. Desta forma, a arquitetura ElCity apresenta um monitor de energia, que é responsável por medir o consumo de energia tanto dos recursos do *data center* como nos dispositivos da cidade (responsáveis por uma parcela maior do consumo de energia, se comparados com smartphones). Ao contrário das cidades atuais, a iluminação pública nas cidades inteligentes compreende lâmpadas que são controladas através de uma placa com sensores e atuadores. Esta placa é responsável pela detecção de usuários, coleta seus dados e os envia para o *data center* de computação em nuvem. No entanto, para alcançar os objetivos propostos, esta placa deve estar ligada 24 horas por dia. Para medir seu consumo de

energia, é realizada a multiplicação do seu tempo de funcionamento (em segundos) pelo consumo em Watts obtendo o consumo de energia em Joules, de acordo com a fórmula $J = W \times s$. O mesmo método é também aplicado aos dispositivos ligados nos atuadores da placa, considerando o seu consumo nominal de energia.

Ainda no contexto deste modelo de cidades inteligentes, é necessário medir o consumo de energia do *data center* de computação em nuvem. Nesta área, a implantação de sensores de corrente ou Wattímetros pode ser caro se não for feito no momento em que toda a infraestrutura (*i.e.*, *cluster* ou *data center*) é instalada, além de ser custosa tanto em questões financeiras como em tempo conforme a infraestrutura cresce. Uma solução alternativa e menos dispendiosa é a utilização de modelos de energia para estimar o consumo de componentes ou de um *data center* inteiro (ORGERIE; ASSUNÇÃO; LEFEVRE, 2014). Modelos de energia devem ser leves (em relação ao consumo de recursos computacionais) e não interferir no consumo de energia que eles tentam estimar. Tendo em vista estes requisitos, propõe-se um monitor de energia que explora dados de energia capturados em um pequeno conjunto de nós, a fim de formular uma equação que estende os resultados para um conjunto arbitrário de nós homogêneos. Mais precisamente, a metodologia utilizada é similar a de Luo et al. (2013) que consiste em três etapas:

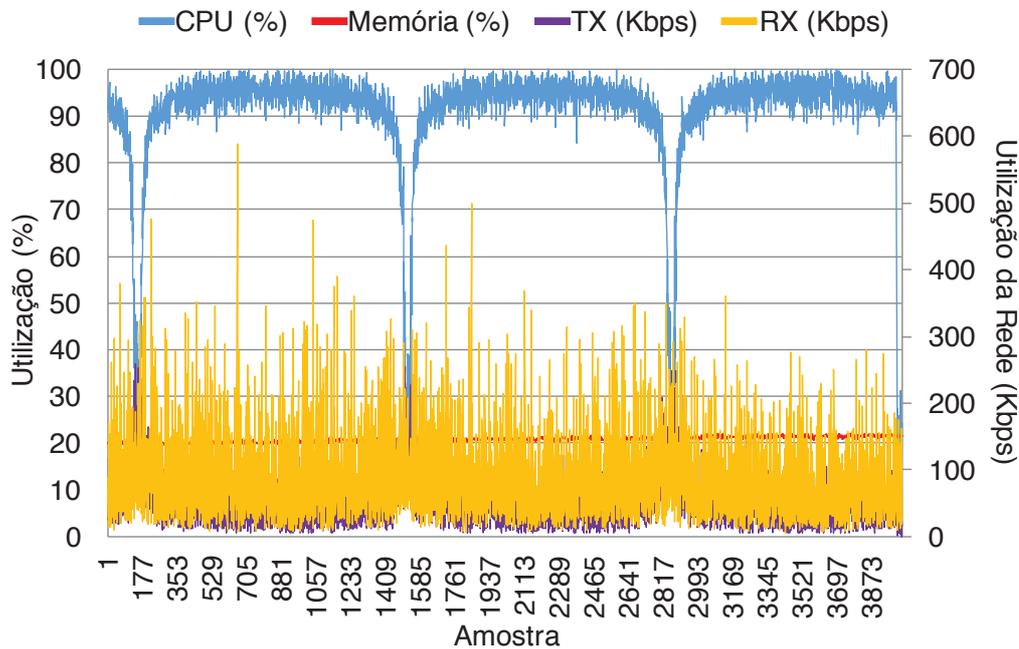
- (i) Coletar amostras de uso de recursos, bem como o consumo de energia da máquina utilizando um medidor de consumo. Neste caso, utilizou-se um medidor Minipa ET-4090² que coletou mais de 8000 amostras usando uma carga composta que pode consumir diversos tipos de recursos dos nós, a fim de representar aplicações reais em ambiente de nuvem (CHEN et al., 2014);
- (ii) Executar métodos de regressão para gerar o modelo de energia a ser utilizado posteriormente;
- (iii) Testar o modelo em um conjunto diferente de dados, coletados com o medidor de diferentes máquinas homogêneas, a fim de validar se o modelo representa corretamente o consumo de energia das demais máquinas.

Para testar o monitor de energia foram analisados vários recursos de um único nó, incluindo CPU, memória principal, cache, uso de disco e utilização de rede, alguns deles apresentados na Figura 18. Após a coleta dos dados foi aplicado o critério de Kaiser (KEITH A. YEOMANS, 1982) para determinar quais fatores têm impacto significativo no consumo de energia, obtendo assim CPU e memória principal como os recursos mais relevantes. A fim de analisar a precisão do modelo gerado foram coletados dados de CPU, memória principal e consumo de energia instantâneos, aplicando posteriormente PCR (Regressão de Componentes Principais) (JOLLIFFE, 1982) em mais de 8.000 amostras obtidas a partir de um único nó. Os dados recolhidos estão alinhados com estudos

²<http://www.minipa.com.br/2/87/132-Minipa-Alicates-Wattmetro-ET-4091>

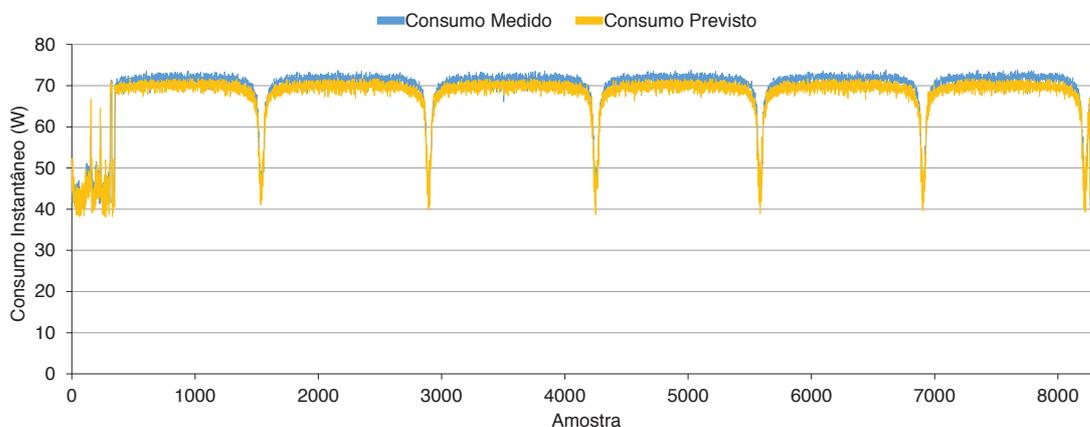
anteriores (ORGERIE; ASSUNÇÃO; LEFEVRE, 2014), que apresentam a CPU como o principal vilão do consumo de energia. Após a geração deste modelo foi realizada a predição da mesma quantidade de amostras de energia baseada em amostras coletadas de CPU e memória de outro nó com mesma configuração de hardware. Comparando estas amostras geradas pela predição de consumo, com as amostras coletadas com o medidor, obteve-se uma precisão média e mediana de 97,15% e 97,72% respectivamente, como pode ser visto na Figura 19.

Figura 18: Utilização de rede, CPU e memória principal durante a execução de uma carga composta.



Fonte: elaborado pelo autor.

Figura 19: Comparativo do consumo instantâneo entre o consumo previsto e o consumo medido.

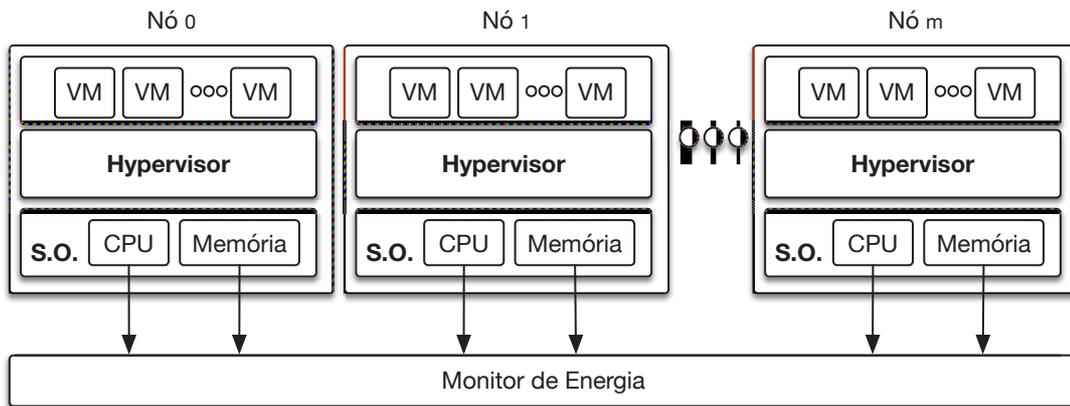


Fonte: elaborado pelo autor.

Após a execução da aplicação, os dados de CPU e memória principal são utilizados

como entrada no modelo gerado, a fim de se obter o consumo de energia instantânea, medido em Watts (W). A grande vantagem deste modelo é o fato de considerar a elasticidade da nuvem, em outras palavras, o modelo leva em conta apenas o consumo de energia dos recursos que foram efetivamente utilizados, e não o consumo total do *data center*, ou um nó específico. O uso de recursos é coletado de todos os nós durante o tempo de execução da aplicação, e através de um arquivo de log que informa o intervalo de tempo que cada máquina é utilizada, apenas as amostras relativas a execução da aplicação são consideradas para o cálculo do consumo de energia. Este processamento de registro é executado *post-mortem* e permite uma análise mais precisa do consumo de energia da aplicação, e não apenas o consumo de energia de toda a infraestrutura. Esta granularidade mais fina permite a utilização de funções de custo, por exemplo, a fim de determinar a viabilidade da utilização da elasticidade em nuvem para executar uma determinada aplicação.

Figura 20: Medição de consumo em ambiente de nuvem utilizando o monitor de energia.



Fonte: elaborado pelo autor.

Seguindo as abordagens tradicionais de monitoramento de nuvem, o modelo utiliza monitoramento periódico: durante a execução do aplicativo, há um total de o observações, onde t ($1 \leq t \leq o$) representa o índice de uma observação em particular. Com o objetivo de apresentar estimativas de consumo de energia, em Joules e Watts, o modelo propõe um conjunto de equações numeradas de 4.1 a 4.5. As equações estão dispostas de maneira “*top-down*”, da que agrega o consumo de vários nós, até a que mede apenas um nó. A Equação 4.1 como etapa final, estima o consumo total de uma aplicação em Joules (Watts \times segundo). Esta equação representa a soma do consumo de energia em cada observação t ($1 \leq t \leq o$) quando se considera o intervalo de tempo entre t e $t - 1$. Dois pontos devem ser ressaltados aqui: (i) se o intervalo entre duas observações é 1s, a última parte da equação pode ser desconsiderada; (ii) quando t é 1, $T(t - 1)$ é assumida como 0 representando o ponto de partida da execução. *ETC*, como apresentado na Equação 4.2, é utilizado para calcular o consumo total de energia de todas as máquinas alocadas na *tésima* observação t , retornando o consumo em Watts.

As Equações 4.3 e 4.4 são utilizadas para estimar o consumo de energia de uma

única máquina em um momento particular. $CPU(i, t)$ e $Memória(i, t)$ são medidos em porcentagem, indicando quanto de CPU e de memória é utilizado em uma máquina i no instante t . Na equação 4.4, α representa o consumo de energia quando a máquina está ociosa e β e δ são multiplicadores que atuam sobre os dados de CPU e memória, respectivamente. Esta equação informa o consumo de energia instantânea em Watts. O modelo aborda recursos homogêneos, mas a única adaptação necessária para contemplar ambientes heterogêneos seria a alteração dos multiplicadores da Equação 4.4 para cada tipo de máquina presente no *data center*. Por fim, a Equação 4.5 apresenta o consumo de energia da aplicação quando se emprega uma quantidade específica de nós, aqui representada como z . Este cálculo resulta no consumo total de energia, também representado em Joules, gastos por esta quantidade de nós.

$$TC = \sum_{t=1}^o ETC(t) \times (tempo(t) - tempo(t - 1)) \quad (4.1)$$

$$ETC(t) = \sum_{i=0}^{Maquina} MC(i, t) \times x \begin{cases} x = 0 & \text{se a maquina } i \text{ não está ativa no instante } t; \\ x = 1 & \text{se a maquina } i \text{ está ativa no instante } t. \end{cases} \quad (4.2)$$

$$MC(i, t) = f(CPU(i, t), Memória(i, t)) \quad (4.3)$$

$$f(a, b) = \alpha + \beta \times a + \delta \times b \quad (4.4)$$

$$NEC(z) = \sum_{i=0}^{TempoAplicação} ETC(i) \times y \begin{cases} y = 0 & \text{se em } i, \text{ o total de maquinas ativas é } \neq z; \\ y = 1 & \text{se em } i, \text{ o total de maquinas ativas } = z. \end{cases} \quad (4.5)$$

4.5 Gerenciadores de Elasticidade

ElCity utiliza o conceito de elasticidade para ligar e desligar os recursos de acordo com a demanda imposta sobre tais recursos. Esta seção detalha os gerentes de elasticidade que são utilizados em cada um dos níveis.

4.5.1 Gerenciador de Elasticidade dos Dispositivos Móveis

O gerenciador de elasticidade dos dispositivos móveis é responsável por coordenar quais sensores permanecerão ativos em um determinado intervalo de tempo. A interação entre

os dispositivos móveis e a aplicação na nuvem fornece recomendações a um aplicativo localizado dentro dos dispositivos móveis, indicando quais sensores serão utilizados. Este escalonamento de sensores de acordo com Chon et al. (2014), onde os sensores são ligados do que possui menor consumo ao com maior consumo de acordo com a necessidade. Neste caso, os sensores serão ligados de acordo com o que estiver disponível para conectar a um determinado dispositivo na cidade. Ou seja, se um dispositivo possui Bluetooth Low Energy (BLE) e WiFi, é preferível que o sensor Bluetooth do smartphone esteja ativo ao invés do WiFi. Estas sugestões são obtidas através do processamento dos dados de movimentação do usuário, juntamente com os dados fornecidos pelos dispositivos localizados na cidade. Os dados do usuário são enviados ao FrontEnd, que retorna um plano de escalonamento dos sensores, toda vez que o celular é conectado a um carregador e possui conexão com o FrontEnd.

4.5.2 Gerenciador de Elasticidade dos Dispositivos da Cidade

O gerenciador de elasticidade dos dispositivos da cidade é responsável por coordenar a interação entre os dispositivos da cidade e a aplicação na nuvem que fornece recomendações de economia de energia. De forma detalhada, cada dispositivo envia dados para o gerenciador periodicamente informando os usuários que passaram próximo a ele desde a sua última comunicação com a nuvem. Utilizando os dados coletados a partir dos dispositivos da cidade e dos cidadãos, os recursos computacionais na nuvem irão processar um plano de recomendação para gerar economia de energia baseado em seu padrão de movimentação. Estes planos são transmitidos para a administração pública, que vai aprová-los ou não, ou ainda, sugerir modificações sobre eles. Assim, a administração pública envia o plano de energia final para FrontEnd na nuvem, que irá passar esses dados para o gerenciador de elasticidade dos dispositivos da cidade. Por fim, este gerenciador entrará em contato com os dispositivos listados no plano, atualizando-os com os novos horários para ligar e desligar os dispositivos conectados em seus atuadores.

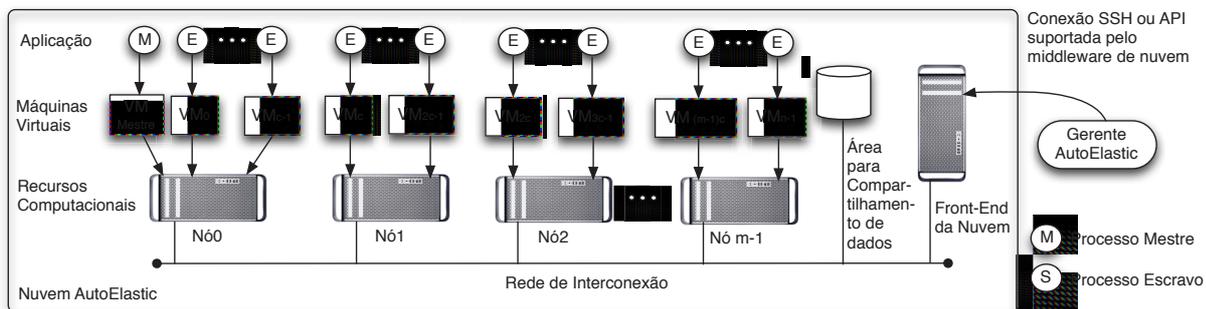
4.5.3 Gerenciador de Elasticidade do Data Center de Nuvem

Na nuvem, é executada uma aplicação que analisa dados de dispositivos da cidade e dos usuários para gerar planos de economia de energia. O objetivo é proporcionar elasticidade de recursos, utilizados para executar esta aplicação, ou seja, aumentando ou diminuindo a quantidade de recursos de acordo com a carga da aplicação. Desta forma, ElCity adota o mesmo modelo de nuvem publicado em Righi et al. (2015), chamado AutoElastic, um trabalho anterior do grupo e utiliza-o como gerenciador de elasticidade do *data center* de computação em nuvem.

AutoElastic é um modelo de elasticidade em nuvem que opera no nível PaaS de uma

plataforma de nuvem, agindo como um *middleware* que permite a transformação de uma aplicação paralela não elástica em uma elástica. O modelo funciona com elasticidade automática e reativa de forma horizontal (gestão de réplicas de VMs), proporcionando alocação e consolidação de nós de computação e máquinas virtuais. Como uma proposta PaaS, AutoElastic propõe um *middleware* para compilar uma aplicação mestre-escravo iterativa, além de um gerenciador de elasticidade.

Figura 21: Distribuição de nós, VMs e processos usando a infraestrutura de nuvem do AutoElastic, na qual cada VM engloba um único processo da aplicação e cada nó executa c VMs, em que c denota o número de CPUs do nó.



Fonte: obtida em (RIGHI et al., 2015).

A Figura 21 ilustra a relação entre os processos, máquinas virtuais e nós computacionais. A nuvem é modelada com m recursos computacionais homogêneos e distribuídos, em que no mínimo um deles (Nó 0) está sempre ativo. Esse nó é encarregado de executar uma VM com o processo mestre e outras c VMs com processos escravos, em que c significa o número de unidades de processamento (cores ou CPU) dentro de um nó em particular. O grão de elasticidade para cada ação se refere a um simples nó. Por fim, a qualquer momento, o número de VMs executando processos escravos é igual a $n = c \times m$. Essa abordagem é baseada no trabalho de Lee et al. (2011), em que os autores procuram explorar uma melhor eficiência em aplicações paralelas.

O usuário pode informar um SLA com a quantidade mínima e máxima de VMs permitida. Caso esse arquivo não for informado, assume-se que a quantidade máxima de VMs é o dobro da quantidade de VMs observada no lançamento da aplicação. O fato de que o gerenciador, e não a aplicação, incrementa ou decrementa o número de recursos, provê o benefício da elasticidade assíncrona. Aqui, elasticidade assíncrona significa que a execução da aplicação e as ações de elasticidade ocorrem simultaneamente, não penalizando a aplicação com a sobrecarga da reconfiguração de recursos (alocação e desalocação). Contudo, esse assincronismo leva ao seguinte questionamento: Como podemos notificar a aplicação sobre a reconfiguração de recursos? Para isso, AutoElastic provê a comunicação entre as VMs e o Gerente utilizando uma área de memória compartilhada. Outras opções de comunicação também podem ser possíveis, incluindo NFS, *middleware* orientado a mensa-

gens (tais como JMS ou AMQP) ou também espaço de tuplas (JavaSpaces, por exemplo). O uso de uma área compartilhada para interação de dados entre VMs é uma abordagem comum em nuvens privadas (CAI et al., 2012; MILOJICIC; LLORENTE; MONTERO, 2011; WEN et al., 2012). AutoElastic utiliza essa ideia para disparar ações como segue:

- Gerente escreve na área compartilhada, enquanto a aplicação lê:
 - Ação1: há um novo nó computacional com c máquinas virtuais, cada uma com um novo processo da aplicação.
 - Ação2: solicita permissão para consolidação de um nó computacional e suas máquinas virtuais.
- Um único processo da aplicação escreve na área compartilhada, enquanto o Gerente lê:
 - Ação3: dando permissão para consolidar o nó requisitado previamente.

Baseado na Ação1, os processos correntes podem iniciar trabalhando com a nova configuração de recursos (um único nó com c VMs, cada uma com um novo processo). A Ação2 é relevante pelas seguintes razões: (i) não parando a execução do processo enquanto procedimentos ou (comunicação ou computação) estão ocorrendo; (ii) garantindo que a aplicação não será abortada com a súbita interrupção de um ou mais processos. Em particular, a segunda razão é importante para aplicações MPI que executam sobre redes TCP/IP, pois elas comumente interrompem com o término prematuro de algum de seus processos. A Ação3 é normalmente tomada pelo processo mestre, que garante que a aplicação está em um estado global consistente em que processos podem ser desconectados apropriadamente. Em seguida, os processos restantes não trocam nenhuma mensagem com o nó dado. É utilizada uma área compartilhada pois isso facilita a notificação de todos os processos sobre a adição ou remoção de recursos, realizando a reconfiguração do canal de comunicação de maneira simples.

AutoElastic oferece elasticidade em nuvem utilizando a técnica de replicação. Na ação de aumentar a infraestrutura, o Gerente aloca um novo nó e lança novas máquinas virtuais nele utilizando um modelo da aplicação. A inicialização de uma VM é finalizada com a execução de um processo escravo que irá realizar requisições para o processo mestre. Essa instanciação de VMs é controlada pelo Gerente e somente após sua inicialização estar completa, o Gerente notifica os outros processos através da Ação1. O procedimento de consolidação aumenta a eficiência de utilização de recursos e também provê um melhor gerenciamento do consumo de energia. Particularmente, Baliga et al. (2011) afirmam que o número de VMs em um nó não é um fator influente para o consumo de energia, mas sim o fato de um nó estar ligado ou não.

Como em (CHIU; AGRAWAL, 2010) e (IMAI; CHESTNA; VARELA, 2012), o monitoramento de dados é realizado de forma periódica. Assim, o Gerente AutoElastic

obtem a métrica CPU, aplica séries temporais baseando-se em valores passados e compara a métrica final com os *thresholds* superior e inferior. Mais precisamente, é empregada a técnica de Média Móvel de acordo com as Equações 4.6 e 4.7. $PC(i)$ retorna uma previsão de carga de CPU quando considerando a execução de n VMs com processos escravo na observação número i do Gerente. Para realizar isso, $MM(i, j)$ informa a carga de CPU de uma máquina virtual j na observação i . A Equação 4.7 utiliza média móvel considerando as z observações mais recentes da carga de CPU $Carga(k, j)$ da VM j , em que $i - z \leq k \leq i$. Por fim, a Ação1 é disparada se PC for maior que o *threshold* superior, enquanto a Ação2 é acionada quando PC for menor que o *threshold* inferior.

$$PC(i) = \frac{1}{n} \cdot \sum_{j=0}^{n-1} MM(i, j) \quad (4.6)$$

em que

$$MM(i, j) = \frac{\sum_{k=i-z+1}^i Carga(k, j)}{z} \quad (4.7)$$

para $i \geq z$.

4.6 Aplicação de Recomendação de Economia de Energia

Esta seção apresenta como foi projetada a aplicação que analisa os traços dos cidadãos. A aplicação de recomendação de economia de energia recebe o histórico de contexto do cidadãos (ROSA et al., 2015) contendo localização, data/hora coletada e identificação do usuário. Os mesmos dados, mas desta vez coletados pelos dispositivos na cidade também são enviados ao FrontEnd na nuvem. Isto ocorre pois caso algum dispositivo ainda não tenha sincronizado com a nuvem, estes dados podem estar disponíveis de alguma outra fonte. Após o processamento, é gerado um plano de economia de energia de acordo com os dispositivos da cidade que os cidadãos passaram e os respectivos horários. Esta informação é enviada para a administração pública que aprova ou não as recomendações para ligar e desligar os dispositivos da cidade.

A aplicação segue o modelo iterativo de passagem de mensagens. As aplicações paralelas que executam na nuvem proposta devem ser projetadas segundo o modelo MPMD (*Multiple Program Multiple Data*) (WILKINSON; ALLEN, 2005), no qual o mestre tem um executável e os escravos outro. Além disso, as diretivas de comunicação suportam a criação dinâmica de processos e a reorganização da topologia de comunicação durante a execução da aplicação, similar a aplicações que utilizam MPI 2.0.

O Algoritmo 1 apresenta um pseudocódigo da aplicação desenvolvida para analisar os dados fornecidos pelos cidadãos. O aplicativo recebe como entrada os dados dos cidadãos e dispositivos, os pontos de interesse que representam a localização onde os dispositivos da cidade estão, a distância mínima em que o cidadão deve estar para ligar o dispositivo,

bem como a quantidade inicial de escravos para iniciar o processamento. Esta abordagem contendo a distância mínima torna-se mais eficiente do que a utilização de sensores de presença, por exemplo, devido a possibilidade de poder acionar as lâmpadas de forma proativa, e de forma transparente ao usuário, e não apenas quando este está a poucos metros do dispositivo.

Após a entrada dos dados, o Algoritmo 1 lê uma certa quantidade de dados (de tamanho configurável) e subsequentemente, distribui entre os escravos ativos para processá-los. A qualquer momento novos escravos podem ser instanciados em novos recursos, e estes novos recursos serão utilizados no ciclo de processamento seguinte. O mestre, no final de um ciclo, executa a agregação dos resultados parciais, armazenando-os em uma estrutura de dados local. Ainda, seguindo as diretivas de execução da aplicação, a inicialização de uma nova máquina virtual deve também iniciar a execução de um processo escravo, o qual requisita conexão com o processo mestre automaticamente, como apresentado no Algoritmo 2.

Algoritmo 1: Pseudo-código do processo mestre da aplicação paralela que gera planos de economia de energia baseado nos dados do cidadão e dispositivos da cidade.

```

input : dados a serem processados, pontos de interesse, distância mínima e quantidade inicial de
          escravos
output: plano de economia de energia
1 início
2   enquanto houver_dados faça
3     para (j=0; j < tamanho_chunk; j++) faça
4       | le_dados();
5     fim
6     atualiza_recursos(socket);
7     para (j=0; j < tamanho; j++) faça
8       | aceita_conexão(escravos[j], portas[j]);
9     fim
10    para (j=0; j < qt_escravos; j++) faça
11      | tarefa = cria_tarefa(dados[tamanho_chunk/qt_escravos], pontos_interesse,
12      | dist_minima);
12      | envia_assincrono(escravos[j], tarefa);
13    fim
14    para (j=0; j < qt_escravos; j++) faça
15      | recebe_sincrono(escravos[j], resultados[j]);
16      | guarda_resultados(resultados[j], todos_resultados);
17    fim
18  fim
19  gera_plano_energia(todos_resultados);
20  envia_plano_energia();
21 fim

```

Por fim, o código requerido para que a elasticidade ocorra é apresentado no Algoritmo 3. Em primeiro lugar, é necessário verificar se existe alguma notificação do gerenciador de elasticidade na área de dados compartilhada. Se a Notificação 1 é recebida, o processo mestre lê as informações referentes aos novos escravos e sabe que deve aguardar a conexão deles. Em caso de receber uma Notificação 2, o processo mestre remove do

Algoritmo 2: Pseudo-código do processo escravo da aplicação paralela que gera planos de economia de energia baseado nos dados do cidadão e dispositivos da cidade.

```

input : void
output: void
1 início
2   mestre = procura(endereço_mestre);
3   porta = cria_porta(endereço_IP, VM_id);
4   requisita_conexão(mestre, porta);
5   enquanto (verdadeiro) faça
6     recebe_sincrono(mestre, tarefa);
7     para (j=0; j < dados; j++) faça
8       se (distancia(ponto_interesse, dados[j]) > dist_minima) então
9         resultado += dados[j];
10      fim
11     fim
12     envia_assincrono(mestre, resultado);
13     desconecta(mestre);
14   fim
15 fim

```

seu grupo de processos aqueles pertencentes à máquina física que será consolidada. Após isso, o processo gera uma Notificação 3, liberando a remoção de recursos por parte do gerenciador de elasticidade da nuvem.

Algoritmo 3: Pseudo-código para gerenciar a elasticidade no processo mestre, seguindo as notificações apresentadas por (RIGHI et al., 2015).

```

input : recursos atuais e ações de elasticidade
output: recursos atualizados
1 início
2   alterações = 0;
3   se (Ação == 1) então
4     alterações += adiciona_VMs();
5   fim
6   senão se (Ação == 2) então
7     alterações -= remove_VMs();
8     permissão_consolidação(); //Notificação 3
9   fim
10  se (Ação == 1 ou Ação == 2) então
11  | reorganiza_portas(portas);
12  fim
13  tamanho += alterações;
14 fim

```

Quando o mestre termina a agregação e processamento de todos os dados dos cidadãos, um plano de recomendação de economia de energia é gerado. Este plano informa em cada ponto em um determinado intervalo de tempo se um ou mais cidadãos estão próximos (de acordo com a distância mínima estabelecida), e o tempo de permanência neste ponto. Com base nesses dados, a administração pública pode determinar se este ponto vai agir de uma maneira elástica ou não. O relatório verificado é devolvido para a nuvem e só então as ações nos dispositivos serão enviadas a eles. Ainda, este relatório também informa aos

dispositivos móveis de cada usuário quando e quais sensores devem estar ativos, com base nos disponíveis em cada dispositivo da cidade que ele acionou.

4.7 Considerações Parciais

Este capítulo apresentou o modelo ElCity, um modelo elástico de economia multinível de energia para cidades inteligentes. Esta elasticidade deve ser transparente e não deve afetar os serviços já oferecidos aos cidadãos. O modelo trabalha com três níveis de energia: (i) os usuários, ou cidadãos que utilizam smartphones equipados com sensores, que exigem bateria como fonte de alimentação (e conseqüentemente necessita de recarga); (ii) os dispositivos da cidade, em especial a iluminação pública onde as lâmpadas são comumente programadas para serem ativadas durante toda a noite, com ou sem cidadãos presentes; e (iii) *data centers* de computação em nuvem que apresentam gerenciadores de energia diferentes que analisam e combinam dados do próprio *data center*, cidadãos e dispositivos da cidade, para em seguida gerar planos de economia de energia para os diferentes níveis.

Está presente também a descrição de cada um dos componentes do modelo, bem como suas interligações e os detalhes das decisões de projeto em cada um dos níveis. Dentro dos componentes do ElCity, destacam-se os mecanismos de economia de energia em cada um dos níveis (Cidadão, Nuvem e Cidade), onde propõe-se uma extensão do conceito de elasticidade de nuvens computacionais (RIGHI et al., 2015) para gerenciar o consumo de energia em diferentes níveis de consumidores de energia em cidades inteligentes, de modo que os recursos em cada nível são automaticamente ligados e desligados de acordo com o padrão de uso dos cidadãos.

Ainda, é proposta uma forma de monitorar o consumo de energia, especialmente no nível de dispositivos e computação em nuvem (responsáveis por uma parcela maior do consumo de energia, se comparados com smartphones). De forma resumida, a placa que é responsável pela detecção de usuários, coleta seus dados e os envia para o *data center* de computação em nuvem. Para medir seu consumo de energia, é realizada a multiplicação do seu tempo de funcionamento (em segundos) pelo consumo em Watts obtendo o consumo de energia em Joules. O mesmo método é também aplicado aos dispositivos ligados nos atuadores da placa, considerando o seu consumo nominal de energia.

Por fim, está presente uma descrição detalhada da aplicação de recomendação de economia de energia que gera os planos para cada dispositivo na cidade, baseada no padrão de movimentação dos cidadãos. A aplicação de recomendação de economia de energia recebe o histórico de contexto do cidadãos (ROSA et al., 2015) contendo localização, data/hora coletada e identificação do usuário. Após o processamento, é gerado um plano de economia de energia de acordo com os dispositivos da cidade que os cidadãos passaram e os respectivos horários. Após a aprovação deste plano pela administração pública, o gerenciador de elasticidade dos dispositivos da cidade envia as atualizações necessárias

para cada dispositivo da cidade. Da mesma forma, o gerenciador de elasticidade dos sensores do smartphone enviará quais são os sensores do smartphone que podem ser desligados em determinado momento.

5 AVALIAÇÃO DOS RESULTADOS

Este capítulo apresenta os parâmetros, cenários e métricas utilizados nos testes além dos resultados obtidos. Em primeiro lugar é apresentado o ambiente de testes, seguido dos resultados para o monitor de energia, detalhando sua precisão e comportamento durante a execução de uma aplicação com e sem elasticidade na nuvem. Por fim, é apresentado o consumo de energia com base na cidade de Roma em duas situações: cidade atual (sem elasticidade) e cidade inteligente baseada em ElCity.

5.1 Descrevendo o Ambiente de Testes

Para realização dos testes foi utilizado um ambiente composto por dois tipos de dados, reais e sintéticos, da seguinte forma:

- **Dados Reais:** Foi desenvolvida uma aplicação para gerar planos de economia de energia, que é executada utilizando a infraestrutura de nuvem privada OpenNebula. Os recursos computacionais consistem em 6 nós homogêneos (1 FrontEnd e 5 nós de computação), cada um com 4 GB de RAM e dois núcleos de processamento de 2,9 GHz. O gerenciador de elasticidade da nuvem utiliza limites de CPU para prover elasticidade reativa, onde os valores utilizados são 30% e 50% para os limites inferiores e 70% e 90% para os limites superiores. Eles são considerados representativos para validar elasticidade em ambientes de nuvem, de acordo com trabalhos anteriores (RIGHI et al., 2015).
- **Dados Sintéticos:** os dados que representam os dispositivos da cidade e os usuários são sintéticos, vindo de um banco de dados específico disponibilizado por Bracciale et al. (2014) que contém a movimentação de 316 taxistas na cidade de Roma, Itália coletados durante o período de 30 dias. O número de dispositivos da cidade e a sua localização é selecionado com base nestes dados.

5.2 Avaliação do Monitor de Energia

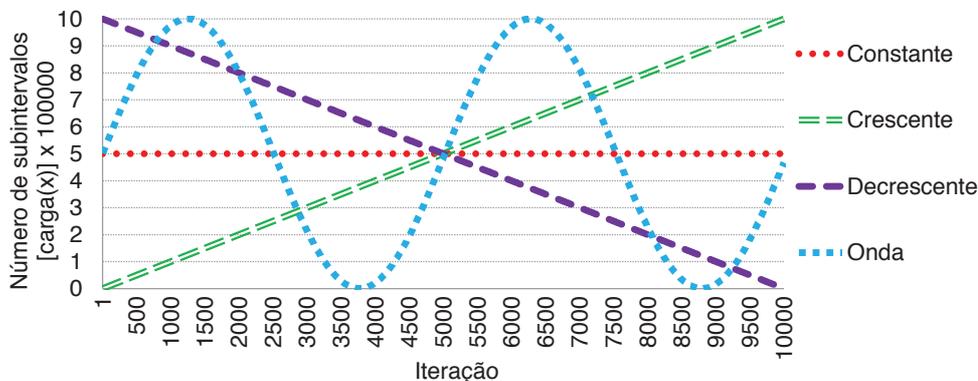
Esta seção apresenta resultados do monitor de energia que compõe o modelo ElCity. Primeiramente está presente a metodologia utilizada, seguida dos resultados do consumo de energia em um único nó, sem elasticidade. Por fim, são apresentados os resultados dos experimentos com elasticidade ativa.

5.2.1 Metodologia de Avaliação do Monitor de Energia

A aplicação utilizada nos testes do monitor de energia calcula a aproximação para a integral do polinômio $f(x)$ num intervalo fechado $[a, b]$. Para tal, foi implementado o método de Newton-Cotes para intervalos fechados conhecido como Regra do Trapézio Repetida (COMANESCU, 2012). A fórmula de Newton-Cotes pode ser útil se o valor do integrando é dada em pontos igualmente espaçados. Considere a partição do intervalo $[a, b]$ em n subintervalos iguais, cada qual de comprimento h ($[x_i, x_{i+1}]$, para $i = 0, 1, 2, \dots, n-1$). Assim, $x_{i+1} - x_i = h = \frac{b-a}{n}$. Dessa forma, pode-se escrever a integral de $f(x)$ como sendo a soma das áreas dos n trapézios contidos dentro do intervalo $[a, b]$. Está sendo reutilizada a mesma aplicação apresentado em Righi et al. (2015), em que a carga de trabalho recebidas pelo processo mestre é uma lista de equações e os seus parâmetros (intervalos e número de subintervalos), obtendo como retorno o resultado para cada equação na lista.

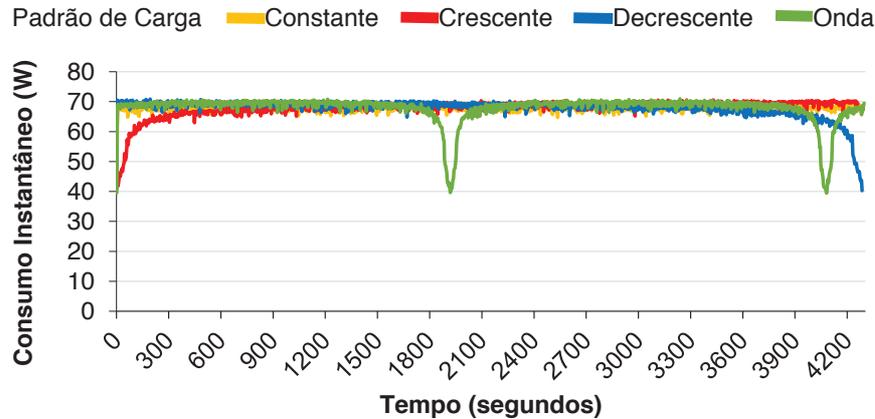
Os parâmetros acima mencionados foram utilizados para modelar quatro cargas de trabalho: Crescente, Constante, Decrescente e Onda. Cargas de trabalho sintéticas foram escolhidos pois são consideradas uma forma representativa para avaliar a elasticidade em nuvens computacionais (ISLAM et al., 2012). A Figura 22 apresenta graficamente uma representação de cada padrão de carga. O eixo x expressa a iteração (cada iteração representa uma equação que será calculada, dividida e distribuída pelo processo mestre), enquanto o eixo y representa a respectiva carga de processamento para aquela iteração. Novamente, a carga é definida pela quantidade de subintervalos s entre os limites a e b , que nos experimentos são 1 e 10, respectivamente. Quanto maior a quantidade de subintervalos, maior é a quantidade de equações a serem calculadas pelos processos escravos, e consequentemente maior é a carga de processamento.

Figura 22: Representação gráfica dos padrões de carga utilizados na avaliação do monitor de energia.



Fonte: adaptado de (RIGHI et al., 2015).

Figura 23: Consumo de energia (em Watts) dos diferentes padrões de carga, sem elasticidade.



Fonte: elaborado pelo autor.

5.2.2 Consumo de Energia Sem Elasticidade

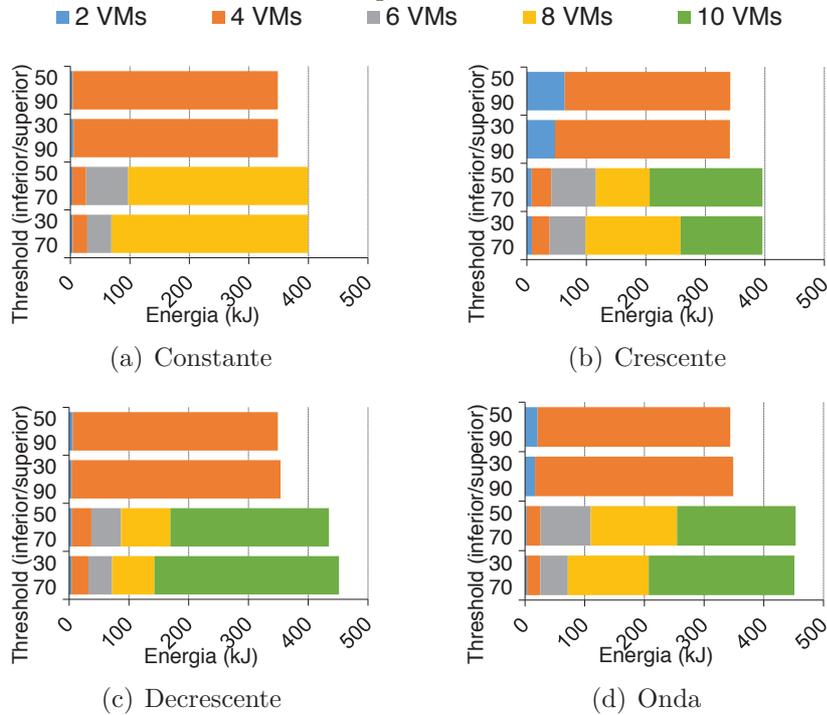
A Figura 23 ilustra o consumo de energia em Watts de acordo com o monitor de energia quando as ações de elasticidade estão desativadas. Neste contexto, um único nó com duas VMs está sendo utilizado para executar os processos escravos. Observa-se que o simples fato de ligar o nó (com Sistema Operacional Ubuntu Linux) consome cerca de 40 Watts. Qualquer atividade de cálculo ocasiona uma elevação desse índice ao intervalo entre 40 e 71 Watts. Embora a função Crescente tenha um crescimento lento com relação a carga da CPU, o consumo de energia aumenta rapidamente até o limite superior do intervalo mencionado. O mesmo comportamento aparece nas funções Decrescente e Onda. Na função de Onda, especialmente, por possuir uma queda de processamento na metade de sua execução de acordo com uma função senoidal.

5.2.3 Consumo de Energia Com Elasticidade Ativa

Quando a elasticidade está ativa é possível observar uma grande variação no número de VMs alocadas para executar a aplicação paralela, impactando diretamente sobre o tempo e o consumo de energia da execução. A Figura 24 apresenta um perfil do consumo de energia da aplicação utilizando o monitor de energia, considerando os quatro padrões de carga. Em particular, este perfil apresenta os resultados da Equação 4.5, demonstrando mais claramente a distribuição e uso de VMs para cada combinação de *thresholds*. Observou-se que o consumo de energia cresce juntamente com o número de máquinas virtuais utilizadas para cada carga, confirmando a relação entre a utilização de recursos e consumo de energia. Iniciando a execução com único nó (2 VMs), a aplicação com carga Crescente, Decrescente e Onda alocam até 5 nós (10 VMs), enquanto a Constante utiliza no máximo 4 nós (8

VMs).

Figura 24: Consumo energético para diferentes quantidades de máquinas virtuais e cargas de trabalho variando os thresholds inferior e superior.



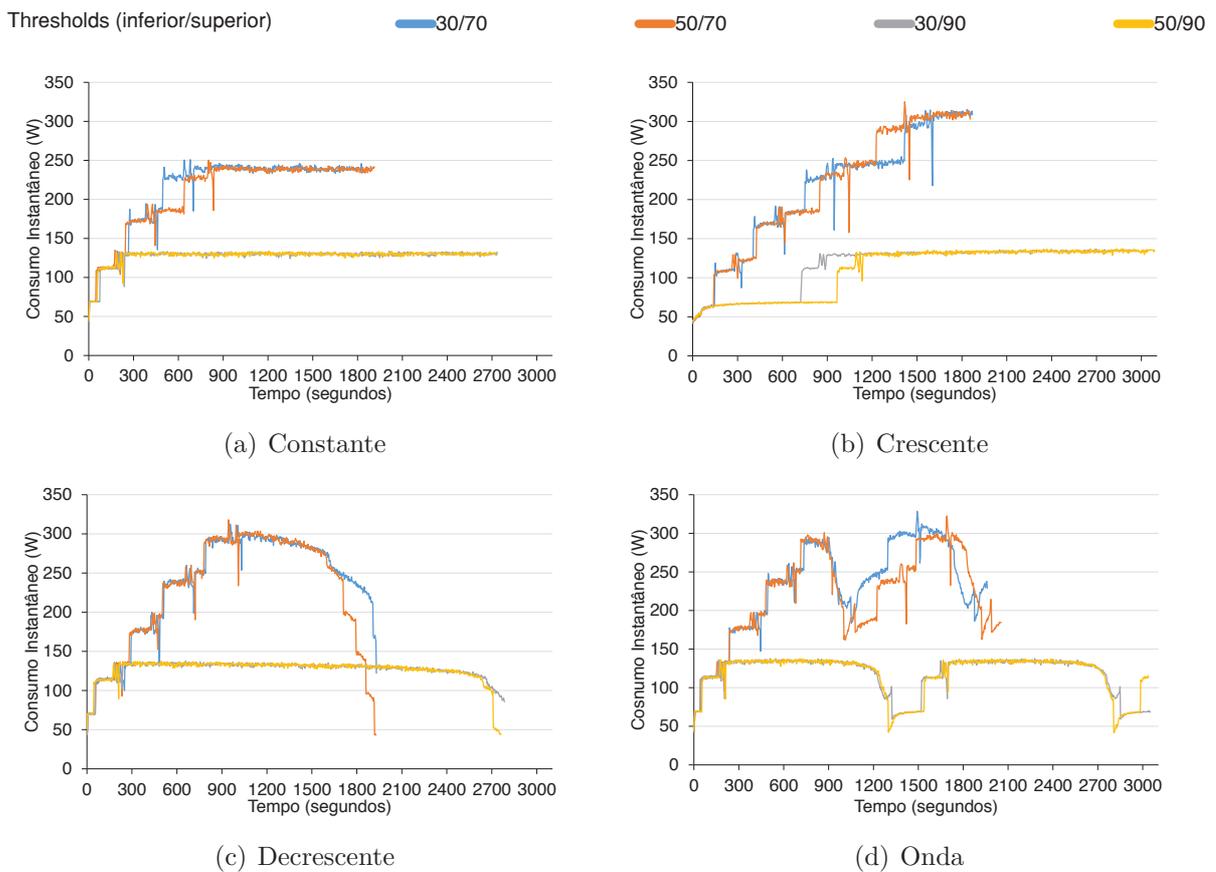
Fonte: elaborado pelo autor.

Na Figura 24 também é possível observar que a variação do *threshold* superior impacta diretamente no consumo total de energia. Neste caso, o número de nós foi traduzido para VMs, pois a aplicação sempre aloca duas máquinas virtuais por nó. O valor de 70% implica no desencadeamento de alocações de VMs de forma mais reativa, uma vez que a carga do sistema excede este limite com maior frequência. A mesma situação sob uma perspectiva diferente por ser observada com um *threshold* superior maior, o que aumenta o tempo de execução, mas fornece um melhor consumo de energia. Em outras palavras, um valor próximo de 100% para o limite superior adia a reconfiguração dos recursos, mantendo o estado de sobrecarga por mais tempo. Por exemplo, o valor de 70% e 90% são responsáveis pela alocação de 10 e 4 VMs no padrão de carga Crescente, respectivamente. Apesar de alocar 6 VMs a mais no primeiro caso, a diferença no consumo de energia não segue esta proporção: mesmo executando com menos VMs, o segundo caso mantém a CPU sobrecarregada (cerca de 90%) por mais tempo, impactando diretamente sobre o consumo energético. O limite inferior não apresenta impacto significativo sobre a execução.

Considerando um modelo de consumo de energia sem elasticidade, o limite inferior do consumo instantâneo seria de 200 W uma vez que o consumo de energia de cada máquina ociosa é de 40 W conforme apresentado na Figura 23 e destacado pelo α na Equação 4.4. A Figura 25 apresenta o gráfico de execução destacando picos e quedas bruscas de consumo de energia quando se analisa o consumo de energia de forma elástica,

utilizando a Equação 4.2 durante o tempo total de execução da aplicação. Neste gráfico podem ser observadas alocações e consolidações de *hosts*, além de oscilações durante a inicialização das VMs. Estes gráficos apresentam as vantagens em analisar a aplicação utilizando um modelo elástico, pois considera apenas o consumo de energia das máquinas que executam computação, e representa de forma mais fiel o consumo energético de uma aplicação que faz uso da elasticidade.

Figura 25: Comportamento do consumo energético das diferentes cargas de trabalho variando os thresholds inferior e superior.



Fonte: elaborado pelo autor.

Além dos resultados obtidos utilizando o monitor de energia ilustrados na Figura 25, a Tabela 3 apresenta os tempos de execução e consumo de energia com e sem elasticidade. A execução sem elasticidade sempre têm os menores valores de consumo de energia, apesar de ter o tempo de execução mais elevado. Considerando o tempo de execução, esta tabela mostra que, quanto menor o valor do threshold superior (70) melhor o tempo de execução da aplicação. O limite inferior não tem um impacto significativo na execução da aplicação, com exceção do padrão de carga decrescente. Nesta situação, o uso de 30% como threshold inferior é responsável por uma consolidação de recursos menos reativa, fazendo com que a aplicação utilize mais recursos por mais tempo. Por outro lado, o valor de 50% é responsável por antecipar a consolidação dos recursos, elevando a carga da CPU em cada instância, mas reduzindo o consumo de energia.

Tabela 3: Tempo e energia medida através do monitor de energia com e sem a elasticidade, variando os thresholds superior e inferior (quando a elasticidade está ativa).

Carga	Elasticidade	Threshold		Tempo	Energia (kJ)
		Superior	Inferior		
Constante		-	-	4277	291.37
	✓		30	1883	399.28
	✓	70	50	1914	399.21
	✓		30	2730	348.53
	✓	90	50	2737	348.79
Crescente		-	-	4261	289.43
	✓		30	1869	395.88
	✓	70	50	1858	395.96
	✓		30	2965	341.28
	✓	90	50	3088	341.93
Decrescente		-	-	4286	290.86
	✓		30	1929	451.49
	✓	70	50	2787	353.75
	✓		30	1926	434.64
	✓	90	50	2761	349.19
Onda		-	-	4296	291.30
	✓		30	1959	451.17
	✓	70	50	2053	453.26
	✓		30	3050	362.04
	✓	90	50	3037	359.48

Fonte: elaborado pelo autor.

5.3 Analisando Consumo de Energia na Cidade Inteligente

Nesta seção, primeiramente é apresentada a metodologia utilizada, seguido dos resultados da aplicação para a recomendação de economia de energia em dois cenários: (i) sem elasticidade; (ii) com a elasticidade ativa, quando utilizados os thresholds 30 % ou 50 % para o limite inferior e 70 % e 90 % para o limite superior. Após a análise do consumo da aplicação, é apresentado o impacto do consumo de energia ao variar tanto o número de cidadãos como número de dispositivos da cidade em duas situações: (i) cidade padrão considerando apenas o consumo das lâmpadas como dispositivos da cidade; (ii) cidade inteligente, considerando o consumo de energia do *data center* e das lâmpadas (ao ativar o plano de energia proposto pela aplicação de recomendação). Na situação (ii), os dispositivos da cidade além de uma lâmpada, também é considerado um *single-board computer*, neste caso uma placa Cubietruck para controlar a presença do cidadão e comunicação com a nuvem. Nessas duas situações, estão sendo considerados dois tipos de lâmpadas: (a) lâmpadas de vapor de sódio e (b) lâmpadas LED.

5.3.1 Metodologia de Avaliação da Aplicação de Recomendação de Economia de Energia

Na nuvem, ElCity utiliza um aplicativo que se encarrega de computar um plano de economia de energia, levando em consideração os dados de entrada a partir de dispositivos da cidade e usuários. Para isso, foi utilizado um conjunto de dados (BRACCIALE et al., 2014) com mais de 21 milhões de registros com a seguinte estrutura: identificação do usuário, localização e um *timestamp* (data/hora). Cada registro descreve a localização precisa de um dos 316 motoristas de táxi na cidade de Roma em um determinado instante de tempo, durante o período: 1 de fevereiro de 2014 e 2 de março de 2014. O conjunto de registros é enviado a partir dos dispositivos da cidade e dos smartphones dos cidadãos ao FrontEnd da nuvem, que irá repassá-los para a aplicação de recomendação de economia de energia. O resultado final é um plano de economia de energia que informa se um dispositivo da cidade tem de ser ligado ou desligado em um determinado momento.

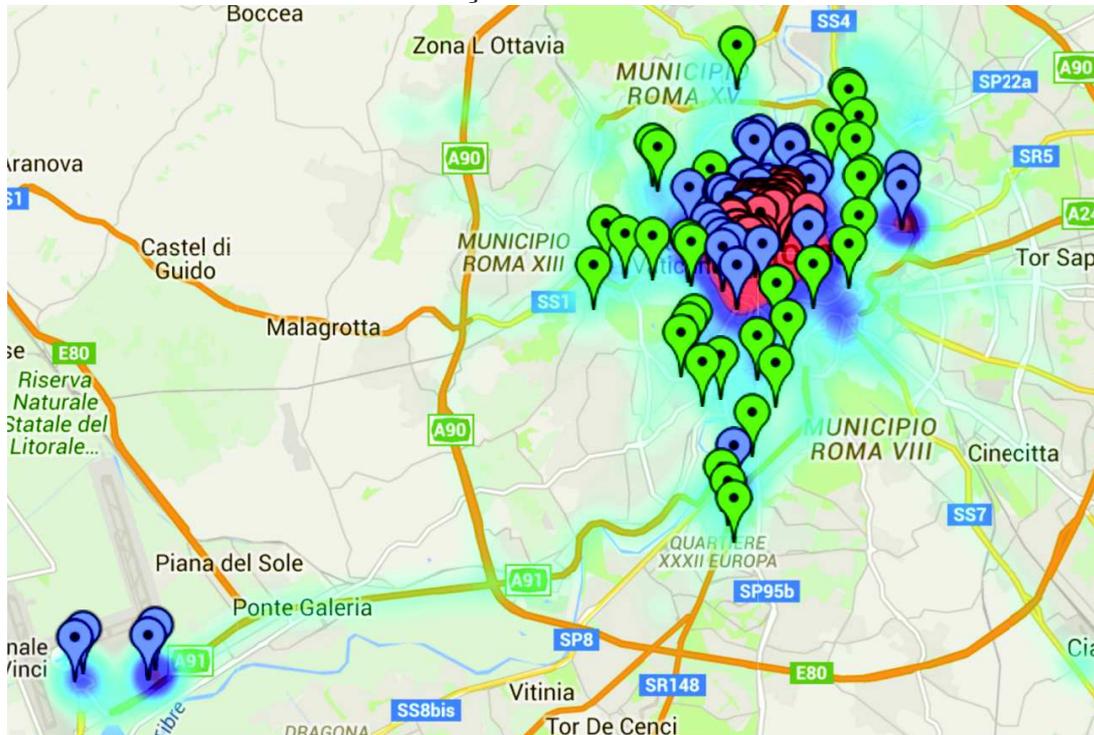
Utilizou-se o intervalo de 5 minutos antes e depois de um usuário passa em um determinado ponto (dispositivo da cidade) para manter o dispositivo ligado. Este tempo foi utilizado para permitir uma sensação de transparência para os cidadãos, ou seja, para o usuário perceber o mínimo possível a ação de liga e desliga das lâmpadas. Além disso, o dispositivo da cidade será ligado sempre que um utilizador é detectado a uma distância de 20 metros a partir deste ponto. A fim de avaliar também o impacto do número de dispositivos e dos cidadãos considerou-se uma variação de 100, 200 e 316 cidadãos (conjunto de dados completo) e uma cidade com 25, 50 e 100 dispositivos. Nesta avaliação não será contemplada a avaliação da economia que poderia ser obtida através do escalonamento dos sensores do smartphone para obtenção de dados de contexto, uma vez que esta avaliação pode ser vista em trabalhos como o de Chon et al. (2014).

Para definir os pontos onde seriam alocados os dispositivos da cidade, foi realizada a plotagem de um mapa de calor usando a API do Google Maps¹ com base nos dados utilizados. Utilizando o mapa de calor foram definidos 100 pontos no mapa como representado na Figura 26. Os pontos de controle (dispositivos da cidade) são sempre igualmente distribuídos de modo que quando utilizados 100 pontos, 33 pontos estão localizados na área mais clara (que possui menos movimento dos usuários) representada pela cor verde, 34 pontos estão na região azul que possui um nível médio de movimentos e os 33 pontos restantes estão localizados na região mais utilizada representada em vermelho na figura. A mesma lógica de distribuição segue quando utilizados 50 e 25 pontos. Destaca-se também, que estes dados de intensidade de movimentação também poderiam ser utilizados pela administração pública a fim de distribuir de forma a otimizar os investimentos com infraestrutura.

Para avaliar a eficiência dos resultados processados pela aplicação os dados serão avaliados segundo dois cenários, conforme apresentado na Figura 27. O primeiro cenário

¹<https://developers.google.com/maps/>

Figura 26: Mapa da cidade de Roma, com rotas em forma de mapa de calor e com pontos indicando dispositivos da cidade. Em verde os com menor movimentação, azul com movimentação média e vermelho com maior movimentação.



Fonte: elaborado pelo autor.

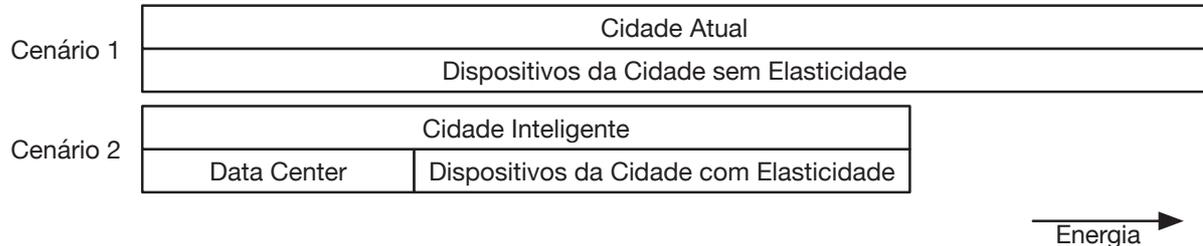
considera a ligação dos dispositivos em um horário específico, uma forma de simplificar o que ocorre atualmente com foto-sensores, neste caso ligando as 19:00 e desligando as 06:00. No segundo cenário, é considerado o consumo de energia dos dispositivos e seus controladores, acrescido do consumo de energia gasto no *data center* para processar os dados dos usuários. Como dispositivos foram consideradas lâmpadas utilizadas para iluminação pública, utilizando os valores de consumo nominal de lâmpadas de vapor de sódio e LED que são de 150 W e 90 W, respectivamente (GE, 2012), por serem os dispositivos utilizados atualmente no cenário brasileiro. Além disso, na cidade inteligente, cada dispositivo da cidade também possui uma placa com atuadores e sensores. Neste caso, foi considerada uma placa Cubietruck² devido seus sensores já incorporados (Bluetooth e WiFi), e seu baixo consumo de energia (cerca de 2,5 W no pico de processamento).

5.3.2 Consumo de Energia do Data Center

A Tabela 4 apresenta o tempo de execução e consumo de energia da aplicação de recomendação de economia de energia. Nestes testes, estão sendo consideradas todas as variações do número de cidadãos e dispositivos da cidade sem elasticidade na nuvem (utilizando um único nó). Esta informação será utilizada como base para avaliar os

²<http://cubieboard.org/model/cb3/>

Figura 27: Cenários avaliados e situação desejável: cidade inteligente, considerando o consumo do *data center* e dispositivos da cidade (lâmpadas com uma placa de sensores/atuadores), apresentando um consumo de energia mais baixo quando comparado com o obtido em uma cidade padrão (somente lâmpadas ligando em horários fixos).



Fonte: elaborado pelo autor.

ganhos que podem ser obtidos utilizando a elasticidade da nuvem para esta aplicação, considerando o tempo de execução e o consumo de energia.

Tabela 4: Tempo e consumo de energia da execução sequencial da aplicação de recomendação de economia de energia variando o número de cidadãos e dispositivos.

Cidadãos Dispositivos	100			200			316		
	25	50	100	25	50	100	25	50	100
Tempo (segundos)	160	259	499	395	591	990	468	721	1420
Energia (kJoules)	10.78	17.82	34.30	25.93	39.91	68.17	31.69	51.04	98.65

Fonte: elaborado pelo autor.

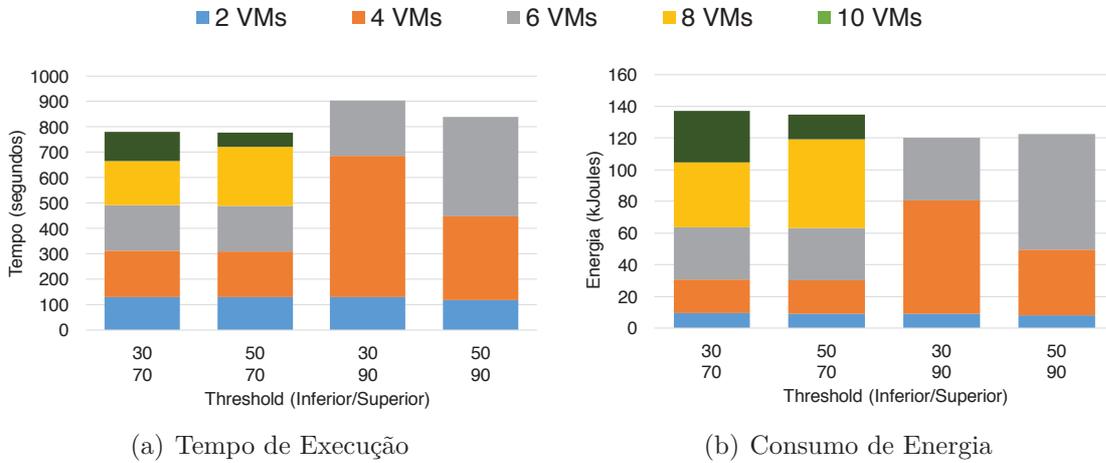
Na Figura 28, bem como na Tabela 5 é possível observar que a variação dos *thresholds* impacta diretamente no consumo total de energia. Mais uma vez, o valor de 70% para o limite superior implica no desencadeamento de alocações de VMs de uma forma mais reativa reduzindo o tempo de execução, no entanto causando um aumento no consumo de energia. Isto também confirma que as cargas de trabalho sintéticas não são válidas apenas para avaliar o desempenho de elasticidade em nuvem, como dito por Islam et al. (2012), mas também para avaliar o consumo de energia de uma aplicação de acordo com um conjunto específico de *thresholds*, uma vez que os resultados obtidos corroboram com os avaliadores anteriormente.

Tabela 5: Tempo de execução e consumo de energia variando os limites inferior e superior da aplicação de recomendação de economia de energia na nuvem com 316 usuários e 100 dispositivos.

	Threshold		Tempo (segundos)	Energia (kJoules)
	Inferior	Superior		
30	70		778	137.03
	90		904	120.02
50	70		779	134.98
	90		840	122.45

Fonte: elaborado pelo autor.

Figura 28: Perfis de tempo de execução (a) e consumo de energia (b) quando considerada a alocação de VMs e variação dos limites inferiores e superiores.



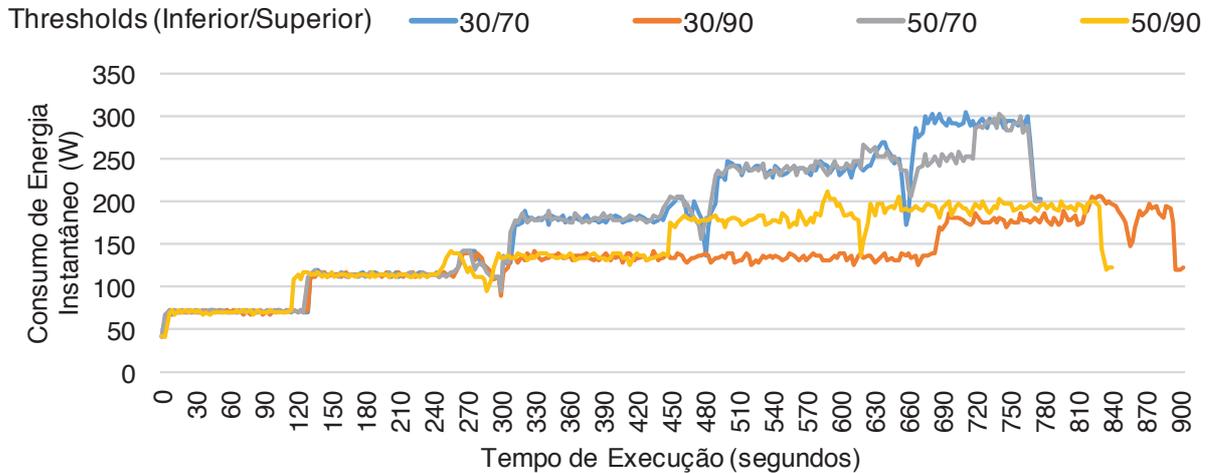
Fonte: elaborado pelo autor.

A Figura 29 apresenta o comportamento da execução da aplicação de recomendação de economia de energia. Esta figura apresenta os picos e quedas no consumo de energia quando se analisa o consumo de energia de uma forma elástica, utilizando a Equação 2 do monitor de energia, durante toda a execução. Nesta figura também é possível observar a alocação e desalocação dos nós, e oscilações durante a inicialização de uma nova VM (lembrando que estão sendo utilizados nós *dual-core*, por isso sempre são consideradas duas VMs por nó). Os gráficos mostram as vantagens em analisar a aplicação utilizando um modelo elástico, considerando apenas o consumo de energia das máquinas que estão envolvidas no processamento de dados. Em comparação, o uso de um *cluster* deve sempre considerar o consumo de energia de todos os nós que estão ligados, de forma independente, se eles estão alocados ou não para uma determinada aplicação.

5.3.3 Consumo de Energia da Iluminação Pública

Com relação ao consumo de energia das lâmpadas controladas por dispositivos da cidade, pode-se observar que os valores são significativamente mais elevados do que o consumo de energia do *data center* utilizado para processar os dados dos cidadãos (menos de 1% para processar dados de 316 cidadãos com dispositivos de 100 cidades). A Figura 30 apresenta a comparação entre o consumo de energia das lâmpadas de vapor de sódio com e sem o uso da elasticidade, variando o número de cidadãos entre 100 e 316 e dispositivos da cidade entre 25 e 100. Considerando o consumo de lâmpadas de vapor de sódio ligadas e desligadas em horários fixos, o consumo de energia observado varia de 4.461.185 kJoules com 100 cidadãos e 25 dispositivos para 17.844.743 kJoules com 316 cidadãos e 100 dispositivos. Quando as luzes são ativadas de acordo com o padrão de circulação dos cidadãos (com elasticidade) o consumo de energia varia entre 188.567 kJoules com 100

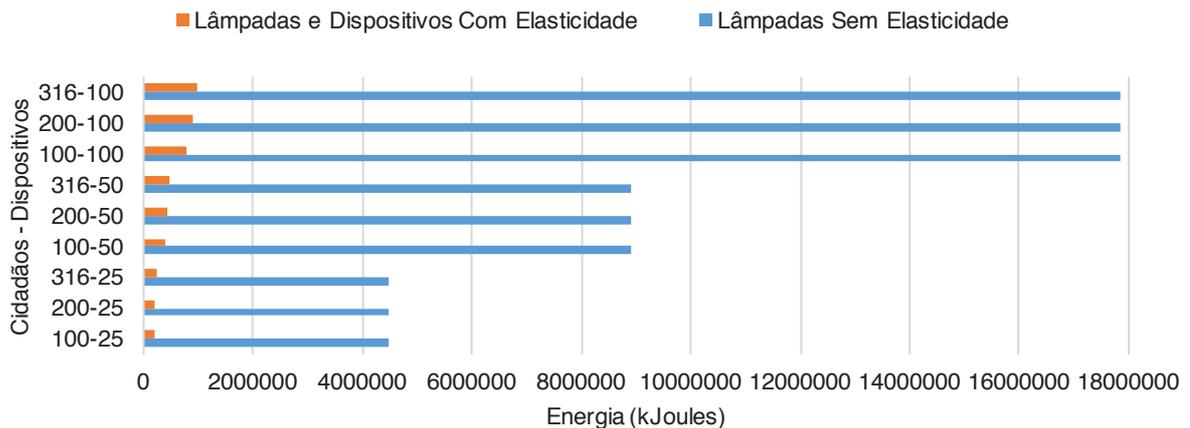
Figura 29: Comportamento da aplicação para recomendação de economia de energia quando executado com elasticidade no ambiente de nuvem, variando os limites superiores e inferiores.



Fonte: elaborado pelo autor.

cidadãos e 25 dispositivos da cidade e 979.379 kJoules com 316 cidadãos e 100 dispositivos da cidade (apenas 4% e 5% do valor anterior com o mesmo número de cidadãos e dispositivos, respectivamente). Destacamos também que o cenário elástico considera a energia de uma placa com sensores e atuador para controlar a lâmpada em questão. Esta placa é mantida sempre ligada para receber as ações do gerente de elasticidade dos dispositivos da cidade e enviar as informações necessárias para a tomada de decisões.

Figura 30: Consumo de energia das lâmpadas de vapor de sódio com e sem elasticidade nos dispositivos da cidade.

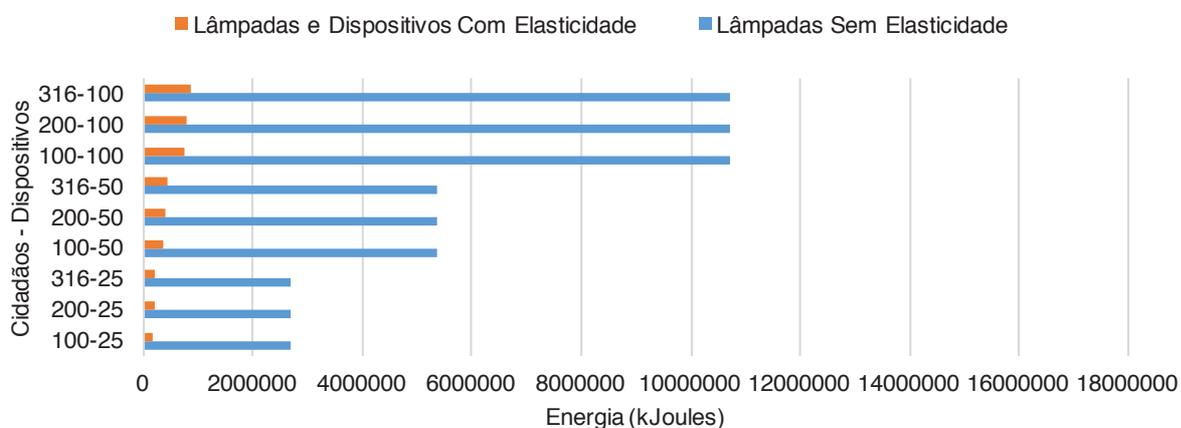


Fonte: elaborado pelo autor.

A Figura 31 apresenta a comparação entre o consumo de energia de lâmpadas LED com e sem o uso das recomendações de elasticidade variando o número de cidadãos e dispositivos. Considerando o consumo de lâmpadas LED ligando e desligando em horários fixos, o consumo de energia do período observado pode variar de 2.676.711 kJoules com 100 cidadãos e 25 dispositivos para 10.706.845 kJoules com 316 cidadãos e 100 disposi-

tivos, o que já representa apenas 60% do consumo de energia das lâmpadas de vapor de sódio. Quando as luzes são ligadas e desligadas de acordo com o padrão de movimentação dos cidadãos (com elasticidade) o consumo de energia cai para 178.030 kJoules com 100 cidadãos e 25 dispositivos e 847.187 kJoules com 316 cidadãos e 100 dispositivos, o que representam apenas 7% e 8% do valor anterior com o mesmo número de cidadãos e dispositivos. Este percentual é maior do que o observado com as lâmpadas de vapor de sódio, devido ao fato do consumo das placas que realizam o controle serem mais representativos sobre o consumo de lâmpadas de LED.

Figura 31: Consumo de energia das lâmpadas de LED com e sem elasticidade nos dispositivos da cidade.



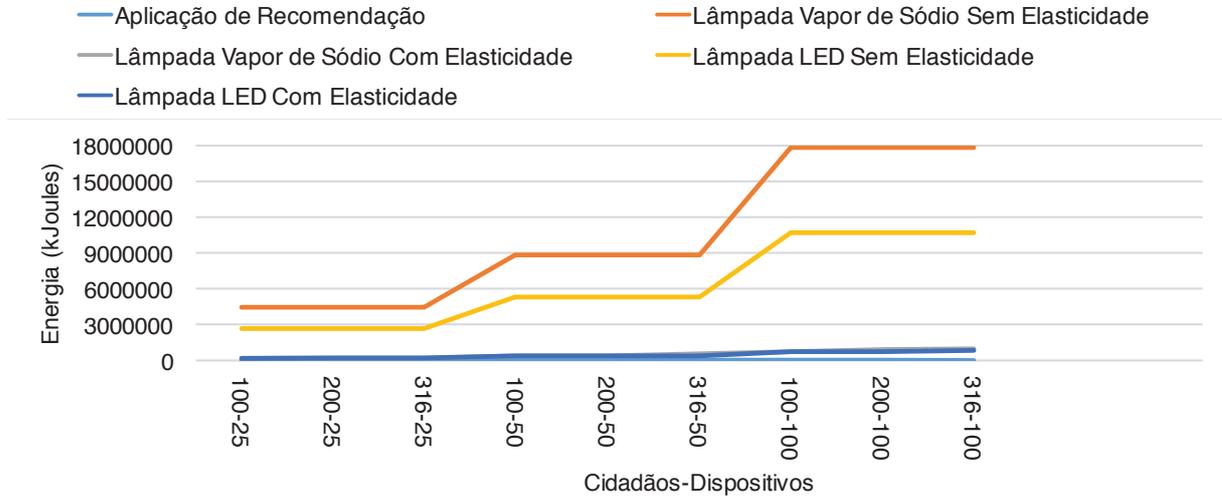
Fonte: elaborado pelo autor.

5.3.4 Consumo de Energia da Cidade Inteligente Com e Sem Elasticidade

A Figura 32 e a Tabela 6 apresentam o consumo de energia de todos os cenários e dispositivos avaliados, bem como o tempo para processar os dados dos cidadãos na nuvem. A aplicação para recomendação de elasticidade foi executada na nuvem com os *thresholds* de 30% e 70% para os limites inferior e superior. Eles foram os responsáveis pelo menor tempo de processamento, o que poderia ser um fator decisivo para o uso destas recomendações em cenários de maior escala. Analisando os dados da Tabela 6, podemos observar que os dispositivos da cidade representam o nível onde é possível obter a maior economia de energia para a cidade. Desta forma, mais dispositivos sendo controlados de forma elástica representam uma economia de energia superior. Como pode ser observado o consumo de energia de 25 dispositivos da cidade sem elasticidade é superior ao consumo de energia de 100 dispositivos da cidade com elasticidade em todos os cenários. Também é possível observar que quanto maior o número de cidadãos e dispositivos, maior o tempo de processamento em nuvem e energia gasta. Além disso, observa-se que a soma dos consumos de energia do *data center* e dispositivos da cidade controlados de forma elástica

é menor do que a energia gasta nas lâmpadas de uma cidade sem elasticidade.

Figura 32: Consumo de energia dos dispositivos da cidade com e sem elasticidade.



Fonte: elaborado pelo autor.

Tabela 6: Tempo de execução da aplicação de recomendação de economia de energia na nuvem com os limiares 30/70 e consumo de energia de diferentes dispositivos da cidade (lâmpadas), variando o número de cidadãos e dispositivos. O consumo de energia na cidade padrão é o consumo dos dispositivos cidade sem elasticidade. O consumo de energia na cidade inteligente é obtido através da soma dos dispositivos com elasticidade e da aplicação.

Cidadãos	Dispositivos	Tempo de Execução da Aplic. (segundos)	Aplicação na Nuvem	Energia (kJoules)			
				Lâmpada de Vapor de Sódio		Lampada de LED	
				Com Elasticidade	Sem Elasticidade	Com Elasticidade	Sem Elasticidade
100	25	199	13	4.461.185	188.567	2.676.711	178.030
100	50	250	26	8.922.371	391.769	5.353.422	364.841
100	100	443	50	17.844.743	786.140	10.706.845	731.244
200	25	339	34	4.461.185	206.315	2.676.711	188.679
200	50	484	53	8.922.371	438.767	5.353.422	393.040
200	100	683	102	17.844.743	889.037	10.706.845	792.982
316	25	547	56	4.461.185	222.614	2.676.711	198.458
316	50	701	92	8.922.371	480.716	5.353.422	418.209
316	100	778	137	17.844.743	979.379	10.706.845	847.187

Fonte: elaborado pelo autor.

5.4 Considerações Parciais

Este capítulo apresentou a avaliação do modelo ElCity considerando diferentes cenários de testes e diferentes aplicações para avaliar seus componentes principais, dentre as

análises realizadas destacam-se:

- (i) Análise do monitor de energia na parte de computação em nuvem, utilizando 2 aplicações com elasticidade, em um primeiro momento cargas sintéticas, e posteriormente a aplicação de recomendação de economia de energia;
- (ii) Análise do consumo de energia variando os thresholds superiores e inferiores da aplicação para recomendação de economia de energia, bem como da aplicação com cargas sintéticas;
- (iii) Análise do impacto dos dados gerados pela aplicação de recomendação de economia de energia em uma cidade que segue o modelo ElCity, variando o número de pontos e cidadãos e testando dois tipos de lâmpadas como dispositivos.

Com relação ao item (i), foi introduzido um monitor de consumo de energia para *data centers* de computação em nuvem. O monitor proposto estima o consumo de energia com base em amostras de CPU e memória com precisão média e mediana 97,15% e 97,72%, respectivamente. Este monitor foi utilizado em conjunto com o *middleware* AutoElastic, alocando e desalocando recursos de acordo com as demandas das duas aplicações analisadas. Focando na reprodutibilidade dos resultados, foi apresentado um conjunto de equações que permite que outros pesquisadores possam empregar este monitor para medir o consumo de energia em suas aplicações elásticas.

Considerando o item (ii), foi possível observar que a variação dos *thresholds* impacta diretamente no consumo total de energia. Quanto menor o threshold superior, maior é o consumo de energia da aplicação, uma vez que as novas máquinas são alocadas de forma mais reativa. Isto faz com que seja incluindo o consumo de energia não só do processamento, mas sim da máquina sem processar, variação apresentada no início do Capítulo, quando avaliado o consumo de energia sem elasticidade. Este comportamento é o mesmo para ambas as aplicações, demonstrando que cargas de trabalho sintéticas não são apenas válidas para analisar a elasticidade de aplicações em nuvem, como dito por Islam et al. (2012), mas também o seu consumo de energia.

Por fim, considerando o item (iii), foi possível obter uma economia de energia superior a 90% desligando os dispositivos da cidade com utilização das recomendações geradas. Esta economia considera a soma do consumo dos dispositivos que recebem as ações do gerenciador na nuvem, os quais permanecem sempre ligados, além do consumo do *data center* para processar os dados dos cidadãos, e as lâmpadas acionadas de forma elástica. Com 316 cidadãos e 100 dispositivos, o consumo de lâmpadas de LED e vapor de sódio com elasticidade representam apenas 5% e 8% do valor sem elasticidade.

6 CONCLUSÃO

Este trabalho apresentou ElCity, um modelo que gerencia diferentes fontes de dados (cidadãos, dispositivos da cidade e *data center* de computação em nuvem) para alcançar uma economia de energia multinível no âmbito de cidade inteligente. Foi introduzido o conceito de economia de energia elástica multinível, que baseia-se na extensão da ideia de elasticidade em nuvem para controlar a energia ao longo dos três níveis mencionados. Esta extensão permite que os gerenciadores de elasticidade liguem e desliguem os recursos em cada nível de acordo com a demanda. O modelo proposto reúne dados de usuários e dispositivos da cidade, processando-os no *data center* para fornecer planos de economia de energia.

Neste trabalho considerou-se unicamente a iluminação pública como dispositivos da cidade, para fins de validação do modelo. Em uma cidade padrão (sem elasticidade) a iluminação pública envolve apenas o consumo de energia das lâmpadas, que são acionadas por foto-sensores, ou em horários específicos. Portanto, o principal desafio do ElCity consiste na inclusão do processamento gerado pelo *data center*, e seu consequente consumo de energia, dispositivos para controlar as lâmpadas, além das próprias lâmpadas, e gerir estes recursos obtendo um consumo de energia total menor do que a abordagem padrão.

O modelo ElCity foi avaliado através de duas estratégias: (i) análise da precisão do monitor de energia com relação a previsão da energia consumida no *data center*; (ii) execução de uma aplicação recomendação de que gera um plano de economia de energia para ser implantado em dispositivos da cidade. Neste caso, em particular, foi utilizado conjunto de dados da cidade de Roma, que apresenta dados de movimentação e localização de cidadãos ao longo de um período de 30 dias.

No primeiro caso, a avaliação do monitor de energia considerando o *data centers* de computação em nuvem, foi possível estimar o consumo de energia com base em amostras de CPU e memória com precisão média e mediana 97,15% e 97,72%, respectivamente. Já a avaliação dos dispositivos da cidade resultou em cerca de 90% de economia de energia quando comparado com a estratégia padrão de iluminação pública (foto-sensores, ou ligando em horários fixos). Também foi realizada a avaliação de quais são os níveis de consumo de energia mais decisivos, concluindo que os dispositivos da cidadão são os responsáveis pela maior parcela do consumo de energia. Por fim, observou-se que o futuro da economia de energia nas cidades inteligentes deve levar em conta a combinação de diversas fontes de dados, e não tratá-las apenas de forma isolada. Entretanto, as novas abordagens não devem afetar qualidade dos serviço já oferecido.

6.1 Contribuições

O modelo ElCity buscou atender as lacunas identificadas no estado da arte através da avaliação dos trabalhos relacionados. Neste sentido, o desenvolvimento deste trabalho resultou em contribuições tanto científicas, como contribuições para a sociedade em cidades inteligentes que utilizam o modelo ElCity. As três contribuições do modelo ElCity no âmbito científico estão listadas nos itens abaixo:

- (i) Com o objetivo de desenvolver uma cidade mais sustentável, é introduzido o conceito chamado “economia de energia elástica multinível” para cidades inteligentes, que combina dados de cidadãos, dispositivos da cidade e os nós de *data centers* para ligar e desligar os recursos provenientes destes consumidores de energia;
- (ii) ElCity fornece um *framework* com as configurações de entrada e saída para os cidadãos, dispositivos da cidade e componentes do *data center*, integrando também a administração pública que pode analisar, modificar e aprovar planos de economia de energia;
- (iii) Foi desenvolvido monitor de energia que estima o consumo de energia para aplicações em nuvem com elasticidade com base em dados de CPU e memória com precisão média e mediana 97,15% e 97,72%.

Já com relação as contribuições para a sociedade e o meio ambiente, podem ser destacadas:

- (i) Integração entre usuários e administração pública;
- (ii) Economia de energia nos smartphones, *data center* e cidade com a utilização dos conceitos de elasticidade;
- (iii) Aumento da economia de energia proveniente da iluminação pública, e consequentemente colaborando para a redução da emissão de gases do efeito estufa;
- (iv) Possibilidade de melhoria do planejamento urbano com base nos dados fornecidos pelos cidadãos.

Ainda, a Tabela 7 ressalta as contribuições do modelo ElCity em comparação com os demais trabalhos analisados. Pode ser observado que o modelo utiliza os dados de contexto de usuários através de uma aplicação móvel, buscando a economia de energia multinível, os dados são armazenados e analisados com o auxílio da computação em nuvem, há o conceito de elasticidade tanto na nuvem quanto na utilização dos sensores no smartphone, e nos dispositivos da cidade. Além disto, o modelo também busca uma melhoria no planejamento urbano das cidades, através de uma abordagem multinível.

Tabela 7: Contribuições do modelo ElCity comparado com os trabalhos relacionados com base nos quesitos previamente definidos.

Trabalho	Contexto	Energia	Dados	Elasticidade	Governança	Multinível
Constandache et al. (2009)	✓	✓		✓		
Nath (2012)	✓	✓		✓		
Chon et al. (2014)	✓	✓		✓		
Yamamoto, Matsu- moto e Nakamura (2012)		✓	✓	✓	✓	
Khan e Kiani (2012)	✓		✓	✓	✓	
Klingert et al. (2015)		✓	✓	✓		
Castro, Jara e Skar- meta (2013)	✓	✓		✓		
Perera et al. (2014)	✓		✓	✓		
Carli et al. (2014)		✓	✓		✓	
ElCity	✓	✓	✓	✓	✓	✓

Fonte: elaborado pelo autor.

6.2 Limitações

Nesta seção serão elencadas algumas das limitações que podem ser encontrados durante a implementação do modelo em escala real.

- Antes de implantar ElCity, é necessária uma profunda discussão sobre a privacidade dos usuários e da utilização dos seus dados, bem como uma análise de protocolos de segurança e autenticação entre cada um dos três níveis;
- A aplicação de recomendação de economia de energia desenvolvida utiliza apenas os dados da atividade dos cidadãos para gerar o plano de economia de energia baseado em um histórico e prever a economia que poderia ter sido obtida. Desta forma, é pertinente o desenvolvimento de uma aplicação de mineração de dados para avaliar os dados históricos de modo que permitam previsões mais precisas sobre o uso de elasticidade nos dispositivos da cidade;
- Alguns cidadãos podem recusar-se a fornecer os seus dados de localização durante todas suas atividades na cidade, devido a preocupações de privacidade, o que pode reduzir a acurácia de uma aplicação de previsão de movimentação;
- A instalação de um computador de placa única (Cubietruck, Arduino, Raspberry PI, entre outros) em cada dispositivo da cidade em um cenário real pode ser custoso tanto em tempo como financeiramente.

6.3 Trabalhos Futuros

Trabalho futuros contemplam testes do modelo ElCity, de uma forma controlada, mas com base em dados em tempo real. Desta forma, pretende-se utilizar Cubietrucks em dispositivos da cidade para controlar tanto a economia de energia, como monitorar a proximidade de usuários através de sensores Bluetooth e WiFi. Além disso, embora presentes na arquitetura ElCity, não foi realizada uma análise da economia de energia no ponto de vista do usuário. As pesquisas futuras terão foco em modelar e avaliar como economizar energia em smartphones desativando alguns sensores quando não são úteis, em determinados momentos, ao longo das rotas dos usuários na cidade e observar o impacto na conectividade do usuário. Além disso, pretende-se caracterizar diferentes tipos de cidadãos e o meio de transporte que esta sendo utilizado. Isto pode ser feito utilizando a velocidade em que o cidadão estaria se movendo, por exemplo, a fim de prover um controle mais preciso de dispositivos pertinentes para cada caso. Por fim, além da iluminação pública considerada neste trabalho, pretende-se aplicar e avaliar as ideias do modelo ElCity ao longo de diferentes dispositivos da cidade, incluindo semáforos de trânsito, bombas de água e fontes.

6.4 Publicações

Ao longo de todo o período desta pesquisa foram produzidos diversos artigos para publicação em revistas e eventos. Além dos artigos referentes a esta dissertação, também foram produzidos outros artigos através da colaboração com outros grupos de pesquisa e universidades. A seguir são listados os artigos publicados durante este período.

- Artigos publicados em revistas internacionais:

1. DA ROSA RIGHI, RODRIGO ; GUERREIRO, VLADIMIR MAGALHAES ; ROSTIROLLA, GUSTAVO ; FACCO RODRIGUES, VINICIUS ; DA COSTA, CRISTIANO ANDRE ; CHIWIACOWSKY, LEONARDO DAGNINO . Using Computational Geometry to Improve Process Rescheduling on Round-Based Parallel Applications. *Scalable Computing. Practice and Experience*, v. 17, p. 13-32, 2016.
2. RODRIGUES, VINICIUS FACCO ; ROSTIROLLA, GUSTAVO ; RIGHI, RODRIGO DA ROSA ; DA COSTA, CRISTIANO ANDRÉ ; BARBOSA, JORGE LUIS VICTÓRIA . Impact of Thresholds and Load Patterns when Executing HPC Applications with Cloud Elasticity. *CLEI Electronic Journal*, v. 19, p. 1:1-1:22, 2016.
3. DA ROSA RIGHI, RODRIGO ; DA COSTA, CRISTIANO ANDRÉ ; RODRIGUES, VINICIUS FACCO ; ROSTIROLLA, GUSTAVO . Joint-analysis

of performance and energy consumption when enabling cloud elasticity for synchronous HPC applications. *Concurrency and Computation*, v. na, p. n/a-n/a, 2015.

- Artigos publicados em eventos internacionais:

1. RIGHI, RODRIGO DA ROSA ; ROSTIROLLA, GUSTAVO ; REIS, EDUARDO SOUZA DOS ; FISCHER, GABRIEL ; CHANG, VICTOR ; RAMACHANDRAN, MUTHU . IDAC: A Sensor-based Model for Presence Control and Idleness Detection in Brazilian Companies. In: Special Session on Recent Advancement in IoT, Big Data and Security, 2016, Rome. Proceedings of the International Conference on Internet of Things and Big Data, 2016. p. 479.
2. RIGHI, R. R. ; REIS, E. S. ; ROSTIROLLA, G. ; COSTA, C. A. ; ALBERTI, A. M. . Exploring Cloud Elasticity on Developing an EPCGlobal-Compliant Middleware. In: 10th Annual IEEE International Conference on RFID (RFID 2016), 2016, Orlando. Proceedings of the 10th Annual IEEE International Conference on RFID (RFID 2016). Los Alamitos, CA, USA: IEEE, 2016. v. 10. p. 53-57.
3. ROSTIROLLA, GUSTAVO; DA ROSA RIGHI, RODRIGO ; RODRIGUES, VINICIUS FACCO ; VELHO, PEDRO ; PADOIN, EDSON LUIZ . GreenHPC: a novel framework to measure energy consumption on HPC applications. In: 2015 Sustainable Internet and ICT for Sustainability (SustainIT), 2015, Madrid. 2015 Sustainable Internet and ICT for Sustainability (SustainIT). p. 1.
4. DA ROSA RIGHI, RODRIGO ; VEITH, ALEXANDRE ; RODRIGUES, VINICIUS FACCO ; ROSTIROLLA, GUSTAVO ; DA COSTA, CRISTIANO ANDRÉ ; FARIAS, KLEINNER ; ALBERTI, ANTONIO MARCOS . Rescheduling and checkpointing as strategies to run synchronous parallel programs on P2P desktop grids. In: the 30th Annual ACM Symposium, 2015, Salamanca. Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC '15. v. 14. p. 501-504.
5. RIGHI, R. R. ; VEITH, A. ; ROSTIROLLA, G. ; RODRIGUES, V. F. ; COSTA, C. A. . BSPonP2P: Towards Running Bulk-Synchronous Parallel Applications on P2P Desktop Grids. In: The 2015 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2015), 2015, Las Vegas, USA. Proceedings of the 21st International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2015). San Diego, California, USA: CSREA Press, 2015. p. 374-380.
6. RABUSKE, F. ; COSTA, C. A. ; RIGHI, R. R. ; ROSTIROLLA, G. ; ALBERTI, A. M. ; BUSSE, A. ; HEISS, H. . GetLB++: Improving Transaction

Load Balancing on the Electronic Funds Transfer Landscape. In: 21st International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2015), 2015, Las Vegas, USA. Proceedings of 21st International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2015). San Diego, California, USA: CSREA Press, 2015. p. 13-19.

7. MARQUES, B. S. ; NARDIN, I. F. ; ROSTIROLLA, G. ; COSTA, C. A. ; RIGHI, R. R. . On Evaluating Internet of Things In Centralised and Distributed Architectures. In: 14th International Conference WWW/Internet (ICWI 2015), 2015, Dublin, Ireland. Proceedings of the 14th International Conference WWW/Internet (ICWI 2015). Lisboa, Portugal: IADIS Press, 2015. v. 14. p. 95-102.
8. RODRIGUES, V. F. ; ROSTIROLLA, G. ; RIGHI, R. R. ; COSTA, C. A. . Cloud Elasticity for HPC Applications: Observing Energy, Performance and Cost. In: XLI Conferência Latinoamericana de Informática (CLEI 2015), 2015, Arequipa, Peru. Proceeding of the XLI Conferência Latinoamericana de Informática (CLEI 2015), 2015, 2015. v. 41. p. 189-200.
9. RODRIGUES, VINICIUS FACCO ; ROSTIROLLA, GUSTAVO ; DA ROSA RIGHI, RODRIGO ; ANDRE DA COSTA, CRISTIANO ; BARBOSA, JORGE LUIS VICTORIA . Cloud elasticity for HPC applications: Observing energy, performance and cost. In: 2015 XLI Latin American Computing Conference (CLEI), 2015, Arequipa. 2015 Latin American Computing Conference (CLEI), 2015. p. 1.

- Artigos publicados em eventos regionais/nacionais:

1. ROSTIROLLA, G.; WENDT, I. G. ; COSTA, C. A. ; RIGHI, R. R. . Proposição de um Modelo de Consumo de Energia para Aplicações Elásticas em Nuvem. In: XIV Workshop de Computação em Clouds e Aplicações (WCGA 2016), 2016, Salvador, BA. Anais do Simpósio Brasileiro em Redes de Computadores (SBRC 2016). Porto Alegre, RS: Sociedade Brasileira de Computação, 2016. v. 14. p. 56-59.
2. ROSTIROLLA, G.; RODRIGUES, V. F. ; RIGHI, R. R. ; COSTA, C. A. . Analyzing Energy Consumption of Elastic HPC Applications in the Cloud. In: XIII Workshop de Processamento Paralelo e Distribuído (WSPPD 2015), 2015, Porto Alegre, RS. Anais do XIII Workshop de Processamento Paralelo e Distribuído (WSPPD 2015), 2015. p. 1-4.
3. RODRIGUES, V. F. ; ROSTIROLLA, G. ; RIGHI, R. R. ; COSTA, C. A. . Analyzing Performance and Efficiency of HPC Applications in the Cloud.

- In: XIII Workshop de Processamento Paralelo e Distribuído (WSPPD 2015), 2015, Porto Alegre, RS. Anais do XIII Workshop de Processamento Paralelo e Distribuído (WSPPD 2015), 2015. p. 25-28.
4. ROSTIROLLA, G.; RODRIGUES, V. F. ; RIGHI, R. R. . Um Modelo de Consumo de Energia para Ambientes de Nuvem com Elasticidade. In: 13ª Escola Regional de Redes de Computadores (ERRC 2015), 2015, Passo Fundo. Anais da 13ª Escola Regional de Redes de Computadores. Porto Alegre - RS: Sociedade Brasileira de Computação SBC, 2015. p. 37-44.
 5. RODRIGUES, V. F. ; ROSTIROLLA, G. ; RIGHI, R. R. . Elasticidade Assíncrona: Transferência Não Bloqueante de VMs para Viabilizar a Reorganização de Aplicações HPC em Cloud Computing. In: 13ª Escola Regional de Redes de Computadores, 2015, Passo Fundo. Anais da 13ª Escola Regional de Redes de Computadores. Porto Alegre - RS: Sociedade Brasileira de Computação SBC, 2015. p. 45-52.
 6. ROSTIROLLA, G.; RODRIGUES, V. F. ; RIGHI, R. R. ; VELHO, P. . Análise de Desempenho e Consumo de um Cluster Baseado em Computadores de Placa Única. In: 12ª Escola Regional de Redes de Computadores, 2014, Canoas. 12ª Escola Regional de Redes de Computadores, 2014.
 7. RODRIGUES, V. F. ; ROSTIROLLA, G. ; RIGHI, R. R. . Elasticidade Reativa em Nuvem para Aplicações de Alto Desempenho. In: 12ª Escola Regional de Redes de Computadores, 2014, Canoas. 12ª Escola Regional de Redes de Computadores, 2014.
 8. RODRIGUES, V. F. ; ROSTIROLLA, G. ; COSTA, C. A. ; RIGHI, R. R. . Combinando Elasticidade Reativa e Preditiva para a Execução de Aplicações Paralelas Iterativas em Nuvem. In: XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2016), 2016, São Leopoldo. Anais da XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2016). Porto Alegre: Sociedade Brasileira de Computação, 2016. v. 1. p. 159-160.
 9. ROSTIROLLA, G.; RODRIGUES, V. F. ; COSTA, C. A. ; RIGHI, R. R. . Correlacionando Modelos de Energia para Aplicações Paralelas em Nuvem. In: XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2016), 2016, São Leopoldo. Anais da XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2016). Porto Alegre: Sociedade Brasileira de Computação, 2016. v. 1. p. 163-164.
 10. REIS, E. S. ; ROSTIROLLA, G. ; RIGHI, R. R. . Análise do Consumo de Energia de um Cluster ARM Utilizando o Benchmark NAS. In: XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS

2016), 2016, São Leopoldo. Anais da XVI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2016). Porto Alegre: Sociedade Brasileira de Computação SBC, 2016. v. 1. p. 227-230.

11. RODRIGUES, V. F. ; ROSTIROLLA, G. ; RIGHI, R. R. . Elasticidade Automática Baseada em Thresholds para Aplicações Paralelas Iterativas em Nuvem Computacional. In: 15a Escola Regional de Alto Desempenho (ERAD/RS 2015), 2015, Gramado, RS. Anais da 15a Escola Regional de Alto Desempenho, 2015. v. 15. p. 107-108.
12. ROSTIROLLA, G.; VELHO, P. ; RODRIGUES, V. F. ; RIGHI, R. R. . Análise da Economia de Energia Através do Desligamento de Nós Ociosos em um Cluster com Processadores ARM. In: 15a Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2015), 2015, Gramado, RS. Anais da 15a Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul, 2015. v. 15. p. 97-98.

- Artigos submetidos para avaliação:

1. RIGHI, RODRIGO DA ROSA; ROSTIROLLA, GUSTAVO; COSTA CRISTIANO ANDRÉ DA; FISCHER, GABRIEL SOUTO; WENDT, GUILHERME IVAM; REIS, EDUARDO SOUZA DOS. Automatic Clocking and Idleness Management in Enterprise Environments Using Wireless Sensors. 2016 Latin American Computing Conference (CLEI).
2. RIGHI, RODRIGO DA ROSA, ROSTIROLLA, GUSTAVO; RODRIGUES, VINICIUS FACCO; BARBOSA, JORGE LUIS VICTÓRIA; COSTA CRISTIANO ANDRÉ DA; ALBERTI, ANTONIO MARCOS; CHANG, VICTOR . Towards Enabling Live Thresholding as Utility to Manage Elastic HPC Applications in the Cloud. IEEE Transactions on Cloud Computing.
3. PACHECO, FABIANA BEAL; KLEIN AMAROLINDA ZANELA; RIGHI, RODRIGO DA ROSA; ROSTIROLLA, GUSTAVO. Business Model creation for Products Based on the Internet of Things. Electronic Markets – The International Journal on Networked Business.
4. RIGHI, RODRIGO DA ROSA; RODRIGUES, VINICIUS FACCO; ROSTIROLLA, GUSTAVO; COSTA CRISTIANO ANDRÉ DA; ROLOFF, EDUARDO; NAVAUX, PHILIPPE OLIVER ALEXANDRE. A Lightweight Plug-and-Play Elasticity Service for Self-Organizing Resource Provisioning on Parallel Applications. IEEE Transactions on Services Computing.
5. ROSTIROLLA, GUSTAVO, RODRIGUES, VINICIUS FACCO; RIGHI, RODRIGO DA ROSA; COSTA CRISTIANO ANDRÉ DA, KREUTZ; DIEGO

- LUIS; SINGH, DHANANJAY. EME: an Energy Model for Elastic Cloud-based HPC Applications. *Revista Brasileira de Computação Aplicada*.
6. ROSTIROLLA, GUSTAVO; RIGHI, RODRIGO DA ROSA; BARBOSA, JORGE LUIS VICTÓRIA; COSTA CRISTIANO ANDRÉ DA. EICity: an Elastic Multilevel Energy Saving Model for Smart Cities. *Simulation Modelling Practice and Theory*.

REFERÊNCIAS

- ALAWADHI, S.; ALDAMA-NALDA, A.; CHOURABI, H.; GIL-GARCIA, J.; LEUNG, S.; MELLOULI, S.; NAM, T.; PARDO, T.; SCHOLL, H.; WALKER, S. Building Understanding of Smart City Initiatives. In: SCHOLL, H.; JANSSEN, M.; WIMMER, M.; MOE, C.; FLAK, L. (Ed.). **Electronic Government**. : Springer Berlin Heidelberg, 2012. p. 40–53. (Lecture Notes in Computer Science, v. 7443).
- ALBINO, V.; BERARDI, U.; DANGELICO, R. M. Smart Cities: definitions, dimensions, performance, and initiatives. **Journal of Urban Technology**, v. 22, n. 1, p. 3–21, 2015.
- ANTHOPOULOS, L.; VAKALI, A. Urban Planning and Smart Cities: interrelations and reciprocities. In: ÁLVAREZ, F.; CLEARY, F.; DARAS, P.; DOMINGUE, J.; GALIS, A.; GARCIA, A.; GAVRAS, A.; KARNOURSKOS, S.; KRKO, S.; LI, M.-S.; LOTZ, V.; MÜLLER, H.; SALVADORI, E.; SASSEN, A.-M.; SCHAFFERS, H.; STILLER, B.; TSELENTIS, G.; TURKAMA, P.; ZAHARIADIS, T. (Ed.). **The Future Internet**. : Springer Berlin Heidelberg, 2012. p. 178–189. (Lecture Notes in Computer Science, v. 7281).
- BALIGA, J.; AYRE, R.; HINTON, K.; TUCKER, R. Green Cloud Computing: balancing energy in processing, storage, and transport. **Proceedings of the IEEE**, v. 99, n. 1, p. 149–167, 2011.
- BELOGLAZOV, A.; BUYYA, R. Energy Efficient Resource Management in Virtualized Cloud Data Centers. In: IEEE/ACM INTERNATIONAL CONFERENCE ON CLUSTER, CLOUD AND GRID COMPUTING, 2010., 2010, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2010. p. 826–831. (CCGRID '10).
- BERGEN, A.; DESMARAIS, R.; GANTI, S.; STEGE, U. Towards software-adaptive green computing based on server power consumption. **Proceedings of the 3rd International Workshop on Green and Sustainable Software - GREENS 2014**, New York, New York, USA, p. 9–16, 2014.
- BRACCIALE, L.; BONOLA, M.; LORETI, P.; BIANCHI, G.; AMICI, R.; RABUFFI, A. **CRAWDAD dataset roma/taxi (v. 2014-07-17)**. 2014.
- CAI, B.; XU, F.; YE, F.; ZHOU, W. Research and application of migrating legacy systems to the private cloud platform with cloudstack. In: AUTOMATION AND LOGISTICS (ICAL), 2012 IEEE INTERNATIONAL CONFERENCE ON, 2012. **Anais...** , 2012. p. 400–404.
- CARLI, R.; DEIDDA, P.; DOTOLI, M.; PELLEGRINO, R. An urban control center for the energy governance of a smart city. In: EMERGING TECHNOLOGY AND FACTORY AUTOMATION (ETFA), 2014 IEEE, 2014. **Anais...** , 2014. p. 1–7.
- CASTRO, M.; JARA, A.; SKARMETA, A. Smart Lighting Solutions for Smart Cities. In: ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS (WAINA), 2013 27TH INTERNATIONAL CONFERENCE ON, 2013. **Anais...** , 2013. p. 1374–1379.

CHEN, F.; GRUNDY, J.; SCHNEIDER, J.-G.; YANG, Y.; HE, Q. Automated Analysis of Performance and Energy Consumption for Cloud Applications. In: ACM/SPEC INTERNATIONAL CONFERENCE ON PERFORMANCE ENGINEERING, 5., 2014, New York, NY, USA. **Proceedings...** ACM, 2014. p. 39–50. (ICPE '14).

CHIU, D.; AGRAWAL, G. Evaluating caching and storage options on the Amazon Web Services Cloud. In: GRID COMPUTING (GRID), 2010 11TH IEEE/ACM INTERNATIONAL CONFERENCE ON, 2010. **Anais...**, 2010. p. 17–24.

CHON, Y.; TALIPOV, E.; SHIN, H.; CHA, H. SmartDC: mobility prediction-based adaptive duty cycling for everyday location monitoring. **Mobile Computing, IEEE Transactions on**, v. 13, n. 3, p. 512–525, March 2014.

COMANESCU, M. Implementation of time-varying observers used in direct field orientation of motor drives by trapezoidal integration. In: POWER ELECTRONICS, MACHINES AND DRIVES (PEMD 2012), 6TH IET INTERNATIONAL CONFERENCE ON, 2012. **Anais...**, 2012. p. 1–6.

CONSTANDACHE, I.; GAONKAR, S.; SAYLER, M.; CHOUDHURY, R.; COX, L. EnLoc: energy-efficient localization for mobile phones. In: INFOCOM 2009, IEEE, 2009. **Anais...**, 2009. p. 2716–2720.

COSTA, C. da; YAMIN, A.; GEYER, C. Toward a General Software Infrastructure for Ubiquitous Computing. **Pervasive Computing, IEEE**, v. 7, n. 1, p. 64–73, Jan 2008.

DE PAOLA, A.; ORTOLANI, M.; LO RE, G.; ANASTASI, G.; DAS, S. K. Intelligent Management Systems for Energy Efficiency in Buildings: a survey. **ACM Comput. Surv.**, New York, NY, USA, v. 47, n. 1, p. 13:1–13:38, June 2014.

DEY, A. K. Understanding and Using Context. **Personal Ubiquitous Comput.**, London, UK, UK, v. 5, n. 1, p. 4–7, Jan. 2001.

DIRKS, S.; GURDGIEV, C.; KEELING, M. Smarter cities for smarter growth. **IBM Global Business Services**, p. 24, 2010.

DOBSON, S.; DENAZIS, S.; FERNÁNDEZ, A.; GAÏTI, D.; GELENBE, E.; MASSACCI, F.; NIXON, P.; SAFFRE, F.; SCHMIDT, N.; ZAMBONELLI, F. A Survey of Autonomic Communications. **ACM Trans. Auton. Adapt. Syst.**, New York, NY, USA, v. 1, n. 2, p. 223–259, Dec. 2006.

DONG, M.; ZHONG, L. Self-constructive High-rate System Energy Modeling for Battery-powered Mobile Systems. In: INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 9., 2011, New York, NY, USA. **Proceedings...** ACM, 2011. p. 335–348. (MobiSys '11).

European Environment Agency. **Urban sprawl in Europe - The ignored challenge.**, 2006. 56 p. v. 10, n. 10.

FARGO, F.; TUNC, C.; AL-NASHIF, Y.; HARIRI, S. Autonomic performance-per-watt management (APM) of cloud resources and services. **Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference on - CAC '13**, New York, New York, USA, p. 1, 2013.

- GARG, S. K.; YEO, C. S.; ANANDASIVAM, A.; BUYYA, R. Environment-conscious scheduling of {HPC} applications on distributed Cloud-oriented data centers. **Journal of Parallel and Distributed Computing**, v. 71, n. 6, p. 732 – 749, 2011. Special Issue on Cloud Computing.
- GE. **Iluminacao publica eficiente torna cidades brasileiras mais sustentaveis**. 2012.
- HARRISON, C.; ECKMAN, B.; HAMILTON, R.; HARTSWICK, P.; KALAGNANAM, J.; PARASZCZAK, J.; WILLIAMS, P. Foundations for Smarter Cities. **IBM Journal of Research and Development**, v. 54, n. 4, p. 1–16, July 2010.
- HAYES, B. Cloud Computing. **Commun. ACM**, New York, NY, USA, v. 51, n. 7, p. 9–11, July 2008.
- HOAREAU, C.; SATOH, I. Modeling and Processing Information for Context-Aware Computing: a survey. **New Generation Computing**, v. 27, n. 3, p. 177–196, 2009.
- IMAI, S.; CHESTNA, T.; VARELA, C. A. Elastic Scalable Cloud Computing Using Application-Level Migration. In: IEEE/ACM FIFTH INTERNATIONAL CONFERENCE ON UTILITY AND CLOUD COMPUTING, 2012., 2012, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2012. p. 91–98. (UCC '12).
- ISLAM, S.; LEE, K.; FEKETE, A.; LIU, A. How a Consumer Can Measure Elasticity for Cloud Platforms. In: ACM/SPEC INTERNATIONAL CONFERENCE ON PERFORMANCE ENGINEERING, 3., 2012, New York, NY, USA. **Proceedings...** ACM, 2012. p. 85–96. (ICPE '12).
- ISMAIL, A. Utilizing big data analytics as a solution for smart cities. In: MEC INTERNATIONAL CONFERENCE ON BIG DATA AND SMART CITY (ICBDSC), 2016., 2016. **Anais...** , 2016. p. 1–5.
- JOLLIFFE, I. T. A Note on the Use of Principal Components in Regression. **Journal of the Royal Statistical Society. Series C (Applied Statistics)**, v. 31, n. 3, p. 300–303, 1982.
- KEITH A. YEOMANS, P. A. G. The Guttman-Kaiser Criterion as a Predictor of the Number of Common Factors. **Journal of the Royal Statistical Society. Series D (The Statistician)**, v. 31, n. 3, p. 221–229, 1982.
- KHAN, Z.; KIANI, S. A Cloud-Based Architecture for Citizen Services in Smart Cities. In: UTILITY AND CLOUD COMPUTING (UCC), 2012 IEEE FIFTH INTERNATIONAL CONFERENCE ON, 2012. **Anais...** , 2012. p. 315–320.
- KITCHIN, R. The real-time city? Big data and smart urbanism. **GeoJournal**, v. 79, n. 1, p. 1–14, 2014.
- KLINGERT, S.; NIEDERMEIER, F.; DUPONT, C.; GIULIANI, G.; SCHULZE, T.; MEER, H. de. Renewable energy-aware data centre operations for smart cities the DC4Cities approach. In: SMART CITIES AND GREEN ICT SYSTEMS (SMARTGREENS), 2015 INTERNATIONAL CONFERENCE ON, 2015. **Anais...** , 2015. p. 1–9.

KNAPPMAYER, M.; KIANI, S.; REETZ, E.; BAKER, N.; TONJES, R. Survey of Context Provisioning Middleware. **Communications Surveys Tutorials, IEEE**, v. 15, n. 3, p. 1492–1519, Third 2013.

KOOMEY, J. Growth in data center electricity use 2005 to 2010. **A report by Analytical Press, completed at the request of The New York Times**, p. 9, 2011.

LE, K.; BILGIR, O.; BIANCHINI, R.; MARTONOSI, M.; NGUYEN, T. D. Managing the Cost, Energy Consumption, and Carbon Footprint of Internet Services. **SIGMETRICS Perform. Eval. Rev.**, New York, NY, USA, v. 38, n. 1, p. 357–358, June 2010.

LEE, J. H.; HANCOCK, M. G.; HU, M.-C. Towards an effective framework for building smart cities: lessons from seoul and san francisco. **Technological Forecasting and Social Change**, v. 89, p. 80 – 99, 2014.

LEE, Y.; AVIZIENIS, R.; BISHARA, A.; XIA, R.; LOCKHART, D.; BATTEN, C.; ASANOVIC, K. Exploring the tradeoffs between programmability and efficiency in data-parallel accelerators. In: **COMPUTER ARCHITECTURE (ISCA), 2011 38TH ANNUAL INTERNATIONAL SYMPOSIUM ON, 2011. Anais...**, 2011. p. 129–140.

LIANG, Q.; ZHANG, J.; ZHANG, Y.-h.; LIANG, J.-m. The placement method of resources and applications based on request prediction in cloud data center. **Information Sciences**, v. 279, p. 735–745, Sep 2014.

LIN, K.; KANSAL, A.; LYMBEROPOULOS, D.; ZHAO, F. Energy-accuracy Trade-off for Continuous Mobile Device Location. In: **INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 8., 2010, New York, NY, USA. Proceedings...** ACM, 2010. p. 285–298. (MobiSys '10).

LOMBARDI, P.; GIORDANO, S.; FAROUH, H.; YOUSEF, W. Modelling the smart city performance. **Innovation: The European Journal of Social Science Research**, v. 25, n. 2, p. 137–149, 2012.

LORIDO-BOTRAN, T.; MIGUEL-ALONSO, J.; LOZANO, J. A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. **Journal of Grid Computing**, Houten, Netherlands, v. 12, n. 4, p. 559–592, 2014.

LUO, L.; WU, W.; TSAI, W.; DI, D.; ZHANG, F. Simulation of power consumption of cloud data centers. **Simulation Modelling Practice and Theory**, v. 39, n. 0, p. 152 – 171, 2013. S.I.Energy efficiency in grids and clouds.

MASTELIC, T.; OLEKSIK, A.; CLAUSSEN, H.; BRANDIC, I.; PIERSON, J.-M.; VASILAKOS, A. V. Cloud Computing: survey on energy efficiency. **ACM Comput. Surv.**, New York, NY, USA, v. 47, n. 2, p. 33:1–33:36, Dec. 2014.

MELL, P. M.; GRANCE, T. **SP 800-145. The NIST Definition of Cloud Computing**. Gaithersburg, MD, United States: National Institute of Standards & Technology, 2011.

MILOJICIC, D.; LLORENTE, I. M.; MONTERO, R. S. OpenNebula: a cloud management tool. **Internet Computing, IEEE**, v. 15, n. 2, p. 11 –14, march-april 2011.

MORVAJ, B.; LUGARIC, L.; KRAJCAR, S. Demonstrating smart buildings and smart grid features in a smart energy city. In: *ENERGETICS (IYCE), PROCEEDINGS OF THE 2011 3RD INTERNATIONAL YOUTH CONFERENCE ON*, 2011. **Anais...**, 2011. p. 1–8.

NATH, S. ACE: exploiting correlation for energy-efficient and continuous context sensing. In: *INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES*, 10., 2012, New York, NY, USA. **Proceedings...** ACM, 2012. p. 29–42. (MobiSys '12).

O'GRADY, M.; O'HARE, G. How Smart Is Your City? **Science**, v. 335, n. 6076, p. 1581–1582, 2012.

ORGERIE, A.-C.; ASSUNÇÃO, M. D. D.; LEFEVRE, L. A survey on techniques for improving the energy efficiency of large-scale distributed systems. **ACM Computing Surveys**, v. 46, n. 4, p. 1–31, Mar. 2014.

PAN, G.; QI, G.; WU, Z.; ZHANG, D.; LI, S. Land-Use Classification Using Taxi GPS Traces. **Intelligent Transportation Systems, IEEE Transactions on**, v. 14, n. 1, p. 113–123, March 2013.

PAN, G.; QI, G.; ZHANG, W.; LI, S.; WU, Z.; YANG, L. Trace analysis and mining for smart cities: issues, methods, and applications. **Communications Magazine, IEEE**, v. 51, n. 6, p. 120–126, June 2013.

PAYA, A.; MARINESCU, D. C. Energy-aware Application Scaling on a Cloud. **CoRR**, v. abs/1307.3306, 2013.

PEJOVIC, V.; MUSOLESI, M. Anticipatory Mobile Computing: a survey of the state of the art and research challenges. **ACM Comput. Surv.**, New York, NY, USA, v. 47, n. 3, p. 47:1–47:29, Apr. 2015.

PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Sensing as a service model for smart cities supported by Internet of Things. **Transactions on Emerging Telecommunications Technologies**, v. 25, n. 1, p. 81–93, 2014.

PETRIDES, P.; NICOLAIDES, G.; TRANCOSO, P. HPC Performance Domains on Multi-core Processors with Virtualization. In: *INTERNATIONAL CONFERENCE ON ARCHITECTURE OF COMPUTING SYSTEMS*, 25., 2012, Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2012. p. 123–134. (ARCS'12).

POWERTUTOR. **PowerTutor, A Power Monitor for Android-Based Mobile Platforms**. 2015.

PUTHAL, D.; SAHOO, B.; MISHRA, S.; SWAIN, S. Cloud Computing Features, Issues, and Challenges: a big picture. In: *COMPUTATIONAL INTELLIGENCE AND NETWORKS (CINE)*, 2015 INTERNATIONAL CONFERENCE ON, 2015. **Anais...** IEEE, 2015. p. 116–123.

QUALCOMM. **Trepp Power Profiler**. 2015.

RIGHI, R.; RODRIGUES, V.; ANDRE DACOSTA, C.; GALANTE, G.; BONA, L.; FERRETO, T. AutoElastic: automatic resource elasticity for high performance applications in the cloud. **Cloud Computing, IEEE Transactions on**, v. PP, n. 99, p. 1–1, 2015.

ROLOFF, E.; BIRCK, F.; DIENER, M.; CARISSIMI, A.; NAVAU, P. Evaluating High Performance Computing on the Windows Azure Platform. In: CLOUD COMPUTING (CLOUD), 2012 IEEE 5TH INTERNATIONAL CONFERENCE ON, 2012. **Anais. . .** , 2012. p. 803 –810.

ROSA, J. H.; BARBOSA, J. L. V.; KICH, M.; BRITO, L. A Multi-Temporal Context-aware System for Competences Management. **International Journal of Artificial Intelligence in Education**, v. 25, n. 4, p. 455–492, 2015.

SATYANARAYANAN, M. Fundamental Challenges in Mobile Computing. In: FIFTEENTH ANNUAL ACM SYMPOSIUM ON PRINCIPLES OF DISTRIBUTED COMPUTING, 1996, New York, NY, USA. **Proceedings. . .** ACM, 1996. p. 1–7. (PODC '96).

SATYANARAYANAN, M. Pervasive computing: vision and challenges. **Personal Communications, IEEE**, v. 8, n. 4, p. 10–17, Aug 2001.

SCHILIT, B.; ADAMS, N.; WANT, R. Context-Aware Computing Applications. In: MOBILE COMPUTING SYSTEMS AND APPLICATIONS, 1994. WMCSA 1994. FIRST WORKSHOP ON, 1994. **Anais. . .** , 1994. p. 85–90.

SCHLEICHER, J. M.; VÖGLER, M.; DUSTDAR, S.; INZINGER, C. Enabling a Smart City Application Ecosystem: requirements and architectural aspects. **IEEE Internet Computing**, v. 20, n. 2, p. 58–65, Mar 2016.

SONG, Z.; ZHANG, X.; ERIKSSON, C. Data Center Energy and Cost Saving Evaluation. **Energy Procedia**, v. 75, p. 1255 – 1260, 2015. Clean, Efficient and Affordable Energy for a Sustainable Future: The 7th International Conference on Applied Energy (ICAE2015).

SU, K.; LI, J.; FU, H. Smart city and the applications. In: ELECTRONICS, COMMUNICATIONS AND CONTROL (ICECC), 2011 INTERNATIONAL CONFERENCE ON, 2011. **Anais. . .** , 2011. p. 1028–1031.

TIAN, Y.; LIN, C.; LI, K. Managing performance and power consumption tradeoff for multiple heterogeneous servers in cloud computing. **Cluster Computing**, v. 17, n. 3, p. 943–955, 2014.

VERGARA, E. J.; NADJM-TEHRANI, S.; PRIHODKO, M. EnergyBox: disclosing the wireless transmission energy cost for mobile devices. **Sustainable Computing: Informatics and Systems**, v. 4, n. 2, p. 118 – 135, 2014. Special Issue on Selected papers from EE-LSDS2013 Conference.

VÖGLER, M.; SCHLEICHER, J. M.; INZINGER, C.; DUSTDAR, S. A Scalable Framework for Provisioning Large-Scale IoT Deployments. **ACM Trans. Internet Technol.**, New York, NY, USA, v. 16, n. 2, p. 11:1–11:20, Mar. 2016.

VÖGLER, M.; SCHLEICHER, J. M.; INZINGER, C.; DUSTDAR, S.; RANJAN, R. Migrating Smart City Applications to the Cloud. **IEEE Cloud Computing**, v. 3, n. 2, p. 72–79, Mar 2016.

WANG, Y.; LIN, J.; ANNAVARAM, M.; JACOBSON, Q. A.; HONG, J.; KRISHNAMACHARI, B.; SADEH, N. A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition. In: INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 7., 2009, New York, NY, USA. **Proceedings...** ACM, 2009. p. 179–192. (MobiSys '09).

WEISER, M. The Computer for the 21st Century. **Scientific American**, v. 265, n. 3, p. 66–75, January 1991.

WEN, X.; GU, G.; LI, Q.; GAO, Y.; ZHANG, X. Comparison of open-source cloud management platforms: openstack and opennebula. In: FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY (FSKD), 2012 9TH INTERNATIONAL CONFERENCE ON, 2012. **Anais...**, 2012. p. 2457–2461.

WILKINSON, B.; ALLEN, C. **Parallel Programming: techniques and applications using networked workstations and parallel computers.** : Pearson/Prentice Hall, 2005. (An Alan R. Apt book).

YAMAMOTO, S.; MATSUMOTO, S.; NAKAMURA, M. Using cloud technologies for large-scale house data in smart city. In: CLOUD COMPUTING TECHNOLOGY AND SCIENCE (CLOUDCOM), 2012 IEEE 4TH INTERNATIONAL CONFERENCE ON, 2012. **Anais...**, 2012. p. 141–148.

YANG, C.-T.; WANG, K.-C.; CHENG, H.-Y.; KUO, C.-T.; CHU, W. C. C. Green Power Management with Dynamic Resource Allocation for Cloud Virtual Machines. In: IEEE INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS, 2011., 2011, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2011. p. 726–733. (HPCC '11).

ZHANG, Q.; METRI, G.; RAGHAVAN, S.; SHI, W. RESCUE: an energy-aware scheduler for cloud environments. **Sustainable Computing: Informatics and Systems**, v. 4, n. 4, p. 215 – 224, 2014. Special Issue on Energy Aware Resource Management and Scheduling (EARMS).

ZHENG, Y.; LIU, Y.; YUAN, J.; XIE, X. Urban Computing with Taxicabs. In: INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING, 13., 2011, New York, NY, USA. **Proceedings...** ACM, 2011. p. 89–98. (UbiComp '11).

ZYGIARIS, S. Smart City Reference Model: assisting planners to conceptualize the building of smart city innovation ecosystems. **Journal of the Knowledge Economy**, v. 4, n. 2, p. 217–231, 2013.