

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA - INE**

Caio César de Melo e Silva

**PROQUALI - PRODUTOR DE CONTEXTO CONSCIENTE DE QUALIDADE: UMA  
ABORDAGEM DE FILTRAGEM PARA AMBIENTES UBÍQUOS MÓVEIS**

Florianópolis- SC

2013



Caio César de Melo e Silva

**PROQUALI - PRODUTOR DE CONTEXTO CONSCIENTE DE QUALIDADE: UMA  
ABORDAGEM DE FILTRAGEM PARA AMBIENTES UBÍQUOS MÓVEIS**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Mário Antônio Ribeiro Dantas

Florianópolis- SC

2013

Catálogo na fonte elaborada pela biblioteca da  
Universidade Federal de Santa Catarina

A ficha catalográfica é confeccionada pela Biblioteca Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:

<http://www.bu.ufsc.br/design/Catalogacao.html>

Caio César de Melo e Silva

**PROQUALI - PRODUTOR DE CONTEXTO CONSCIENTE DE QUALIDADE: UMA  
ABORDAGEM DE FILTRAGEM PARA AMBIENTES UBÍQUOS MÓVEIS**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Ciência da Computação”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis- SC, 21 de julho 2013.

---

Prof Dr. Ronaldo dos Santos Melo  
Coordenador do Curso

---

Prof. Dr. Mário Antônio Ribeiro Dantas  
Orientador

**Banca Examinadora:**

---

Prof. Dr. Mário Antônio Ribeiro Dantas  
Presidente

---

Prof. Dr. Nelson Francisco Favilla Ebecken



---

Prof. Dr. Roberto Willrich

---

Prof. Dr. Frank Siqueira





*Dedico essa dissertação à todos que me suportaram durante o desenvolvimento desse trabalho.*



## AGRADECIMENTOS

Primeiramente, gostaria de agradecer ao meu orientador Prof. Dr. Mario Antônio Ribeiro Dantas por acreditar no projeto dessa dissertação, fornecendo toda a base acadêmica e orientação durante o desenvolvimento desse trabalho. Além disso, agradeço também por sua proatividade em recomendar minha participação em projetos relacionados com o trabalho de pesquisa aqui desenvolvido.

Meus agradecimentos aos professores Reinaldo Haas e Renato Silva, do departamento de física da Universidade Federal de Santa Catarina, por competirem a mim as atribuições de arquiteto e desenvolvedor no projeto de arquiteturas distribuídas para simulações de modelos meteorológicos. Fortaleço minha gratidão pela disponibilidade, fornecida pelos professores, em utilizar os ambientes distribuídos de alta performance para efetuar simulações relevantes para o desenvolvimento dessa dissertação.

Agradeço também a empresa Dígitro, onde participei do desenvolvimento do projeto para monitoramento de veículos em tempo real, tal projeto possibilitou o entendimento de desafios práticos que se relacionavam diretamente com essa dissertação, permitindo verificações mais pertinentes aos desafios encontrados no dia-a-dia.

Aos integrantes do grupo de pesquisa LaPeSD (ressalvo, todos eles!), pelas inúmeras discussões sobre os mais diversos desafios encontrados nos trabalhos de pesquisa desenvolvidos. Também gostaria de agradecer aos integrantes do projeto Petrobras que, apesar da minha pequena participação, deram o ponta pé inicial para o entendimento no que diz respeito ao desenvolvimento de projetos de pesquisa.

Finalmente, agradeço fortemente a minha família e amigos por demonstrarem interesse em entender esse projeto de pesquisa (mesmo que não tivessem conhecimento específico da área) e, conseqüentemente, tornarem o desenvolvimento escrito dessa dissertação um pouco mais didático e instrutivo.



*“Acredite naqueles que estão buscando a verdade. Duvide daqueles que a encontraram.” - André Gide*



## RESUMO

Em ambientes ubíquos móveis a sensibilidade ao contexto exerce um papel fundamental quanto a adaptabilidade às necessidades dos usuários. Os dispositivos ubíquos estão presentes em diversos equipamentos adquiridos e utilizados pelos usuários o que os torna cada vez mais integrados às ações cotidianas. As funcionalidades e interfaces de comunicação possibilitam a esses dispositivos captar e disseminar informações sobre o ambiente de execução. Tais informações, providas dos dispositivos ubíquos móveis, são consideradas os elementos de entrada para a sensibilidade ao contexto.

Dados de contexto fornecem um conjunto de informações as aplicações e serviços para que os mesmos possam se adaptar de maneira coerente com o atual ambiente de execução. Porém, as informações fornecidas pelos dispositivos ubíquos possuem características inerentes de imprecisão devido a forma como são captadas. Dessa maneira, os dados devem possuir algumas garantias de qualidade para que a utilização dos mesmos não prejudique o correto funcionamento das aplicações e serviços sensíveis ao contexto. A qualidade de contexto (QoC), então, é um requisito necessário para que seja possível a construção de aplicações verdadeiramente sensíveis ao contexto.

Nesse sentido, a proposta dessa dissertação se concentra em garantir a qualidade de contexto nos dados providos por dispositivos ubíquos móveis. Para tal, são modelados componentes para provedores ubíquos móveis de dados de contexto (ProQuali). Os componentes da abordagem ProQuali buscam inserir a qualidade nas informações de contexto desde a captura até a disseminação dos dados. As informações que não possuem a qualidade desejada são eliminadas através de um processo de filtragem de dados também previsto na proposta. Para eliminar os dados, o processo de filtragem utiliza políticas de qualidade que se baseiam tanto em parâmetros de QoC quanto em requisitos das aplicações sensíveis ao contexto. Dessa forma, possibilitando ao provedor de contexto o envio de dados de maior relevância para os consumidores de contexto. Para verificar a abordagem ProQuali, foi projetado um ambiente experimental simulado onde os dispositivos ubíquos (ProQuali) enviam dados de contexto para um servidor que processa esses dados e os envia para as aplicações. Os dispositivos enviam informações sobre a posição geográfica do usuário com o intuito de descobrir possíveis padrões nas trajetórias dos mesmos. Para qualificar as informações foram definidas políticas de qualidade baseadas tanto em parâmetros de contexto quanto em requisitos da aplicação. As políticas são utilizadas por alguns métodos que efetuam a identificação e eliminação de dados conflitantes e redundantes, compondo assim todo o processo de filtragem.

Os resultados das experimentações indicaram uma redução significativa na quantidade de dados enviados pelos dispositivos, além de uma melhor garantia de qualidade de contexto. Além disso, a redução no envio dos dados proporcionou aos dispositivos uma melhor performance energética, e um aumento na escalabilidade do servidor. Corroborantemente, as implementações da abordagem exigiram uma maior demanda de processamento dos dispositivos, obtendo uma melhor distribuição do processamento dentro da infraestrutura computacional. Assim, a proposta atingiu o objetivo de garantir a qualidade de contexto em todas as fases do processo de provimento de dados por dispositivos ubíquos.

**Palavras-chave:** Sensibilidade ao contexto, qualidade de contexto, computação ubíqua, computação móvel





## LISTA DE FIGURAS

Figura 1	Estados de conexão dos dispositivos em ambientes móveis .....	29
Figura 2	Taxonomia dos problemas de pesquisa na computação ubíqua .....	34
Figura 3	Arquitetura lógica do mediador de dados de contexto .....	40
Figura 4	Classificação das abordagens de gerenciamento de dados de contexto .....	50
Figura 5	Classificação das abordagens de distribuição de dados de contexto .....	58
Figura 6	Processo de filtragem de informações de contexto .....	68
Figura 7	Modelo lógico do provedor de contexto consciente de qualidade .....	71
Figura 8	Arquitetura lógica de sistemas sensíveis ao contexto .....	75
Figura 9	Figura ilustrativa do ambiente experimental .....	81
Figura 10	Arquitetura implementada para os experimentos .....	83
Figura 11	Quantidade de dados de contexto filtrados .....	86
Figura 12	Trajéorias não filtradas .....	87
Figura 13	Trajéorias filtradas .....	88
Figura 14	Sobrecarga pela utilização do ProQuali .....	89
Figura 15	Consumo de bateria do dispositivo .....	90
Figura 16	Exemplo de arquitetura descentralizada para aplicações sensíveis ao contexto ..	95
Figura 17	Exemplo de arquitetura para aplicações sensíveis ao contexto .....	96



## LISTA DE TABELAS

Tabela 1	Tipos de rede sem fio .....	28
Tabela 2	Estratégias para quantificação de parâmetros de QoC .....	48
Tabela 3	Informações utilizadas para medir parâmetros de QoC .....	49
Tabela 4	Comparação entre os provedores de contexto propostos .....	80
Tabela 5	Ambiente experimental .....	82
Tabela 6	Porcentagem de dados filtrados .....	87



## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	21
1.1 CONTEXTUALIZAÇÃO .....	21
1.2 PERGUNTA DE PESQUISA .....	22
1.3 MOTIVAÇÃO .....	23
1.4 OBJETIVOS .....	24
<b>1.4.1 Objetivo Geral</b> .....	24
<b>1.4.2 Objetivos Específicos</b> .....	24
1.5 MÉTODO DE PESQUISA .....	25
1.6 ORGANIZAÇÃO DO TRABALHO .....	25
<b>2 COMPUTAÇÃO MÓVEL E UBÍQUA</b> .....	27
2.1 COMUNICAÇÃO SEM FIO .....	27
<b>2.1.1 Infraestruturas de rede sem Fio</b> .....	27
<b>2.1.2 Redes de Sensores sem Fio</b> .....	30
2.2 DESAFIOS DA COMPUTAÇÃO UBÍQUA E MÓVEL .....	33
<b>2.2.1 Adaptabilidade</b> .....	34
<b>2.2.2 Gerenciamento de Energia</b> .....	35
<b>2.2.3 Transparência</b> .....	35
<b>2.2.4 Capacidade Computacional dos Dispositivos</b> .....	36
2.3 DISTRIBUIÇÃO DE DADOS EM AMBIENTES UBÍQUOS MÓVEIS .....	37
2.4 CONSIDERAÇÕES SOBRE O CAPÍTULO .....	40
<b>3 SENSIBILIDADE AO CONTEXTO</b> .....	41
3.1 DEFINIÇÃO DE CONTEXTO .....	41
3.2 QUALIDADE DE CONTEXTO .....	44
<b>3.2.1 Parâmetros de qualidade de contexto</b> .....	45
3.3 GERENCIAMENTO DE DADOS DE CONTEXTO .....	49
<b>3.3.1 Representação de contexto</b> .....	50
<b>3.3.2 Processamento de dados de contexto</b> .....	55
3.4 DISTRIBUIÇÃO DE DADOS DE CONTEXTO .....	57
<b>3.4.1 Disseminação</b> .....	58
<b>3.4.2 Sobreposição de Roteamento</b> .....	62
3.5 CONSIDERAÇÕES SOBRE O CAPÍTULO .....	63
<b>4 PROQUALI: PRODUTOR DE CONTEXTO CONSCIENTE DE QUALIDADE</b> ..	65
4.1 POLÍTICAS DE QUALIDADE DE CONTEXTO .....	65
<b>4.1.1 Políticas orientadas à parâmetros de qualidade</b> .....	65

<b>4.1.2 Políticas orientadas à aplicação</b> .....	66
4.2 PROCESSO DE FILTRAGEM .....	67
4.3 MODELO DO PROVEDOR DE CONTEXTO .....	70
<b>4.3.1 Criador de Dados de Contexto</b> .....	70
<b>4.3.2 Injetor de Qualidade de Contexto</b> .....	71
<b>4.3.3 Validador de Contexto</b> .....	72
<b>4.3.4 Cache de Contexto</b> .....	73
<b>4.3.5 Distribuidor de Dados de Contexto</b> .....	73
4.4 CONSIDERAÇÕES SOBRE A PROPOSTA .....	74
4.5 TRABALHOS CORRELATOS .....	77
4.6 CONSIDERAÇÕES SOBRE O CAPÍTULO .....	80
<b>5 AMBIENTE E RESULTADOS EXPERIMENTAIS</b> .....	81
5.1 DESCRIÇÃO DO AMBIENTE EXPERIMENTAL .....	81
5.2 ANÁLISE DOS RESULTADOS .....	85
5.3 CONSIDERAÇÕES SOBRE O CAPÍTULO .....	91
<b>6 CONCLUSÕES, LIMITAÇÕES E TRABALHOS FUTUROS</b> .....	93
6.1 CONCLUSÕES .....	93
6.2 LIMITAÇÕES .....	94
6.3 TRABALHOS FUTUROS .....	96
<b>Referências Bibliográficas</b> .....	99
<b>APÊNDICE A – Trabalhos Publicados</b> .....	109

# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

A computação ubíqua é um paradigma caracterizado pela presença de dispositivos portáteis que estão crescentemente fazendo parte do cotidiano. Tais dispositivos possuem um considerável poder de processamento, espaço de armazenamento e capacidades de comunicação sem fio. Além disso, esses equipamentos possuem diversas funcionalidades e interfaces como GPS, Bluetooth, acelerômetro, reprodutores de áudio, e câmeras digitais. Essas funcionalidades possuem uma forte ligação com o mundo físico, e as informações providas por elas podem ser utilizadas para implantar serviços com a completa consciência do ambiente de execução. A informação capturada do ambiente do usuário é usualmente chamada de contexto e representa o elemento de entrada para computação sensível ao contexto (NAZARIO; DANTAS; TODESCO, 2012).

A sensibilidade ao contexto permite que serviços móveis se adaptem dinamicamente e de forma eficiente tanto à situação atual, quanto ao atual espaço físico e/ou atividade social, e para as desafiadoras e altamente variáveis condições, típicas de ambientes móveis (ex.: limitação de recursos computacionais, conectividade sem fio não confiável e intermitente) (BELLAVISTA et al., 2013). Uma abordagem sensível ao contexto pode, também, minimizar o consumo de energia, tempo de processamento e a quantidade de dados propagados na rede, otimizando, assim, o uso de recursos computacionais.

Entretanto, informações de contexto possuem uma característica intrínseca de imperfeição e sua qualidade é altamente influenciada pelo forma que é captada. De fato, a informação pode estar até incorreta. Grande parte dos sensores possuem imprecisões inerentes (ex.: alguns metros na posição do GPS), e com o passar do tempo o valor captado envelhece em relação as mudanças no mundo físico, aumentando a imprecisão das informações captadas. Os dados captados por sensores podem ser afetados por várias fontes de erros, tais como indisponibilidade, inaplicabilidade, influências externas (ex. temperatura, interferências). Além disso, as operações de refinamento sobre os dados de contexto (ex.: fusão, inferência) também são responsáveis pela redução na qualidade dos dados (HENRICKSEN; INDULSKA, 2004).

Dados de contexto imperfeitos e conflitantes podem impedir o bom funcionamento de aplicações sensíveis ao contexto. Dessa forma, os princípios de qualidade de contexto (QoC) podem ser usados para enfrentar situações de conflito. Algumas pesquisas, como apresentado em (BUCHHOLZ; KÜPPER; SCHIFFERS, 2003) (SHEIKH; WEGDAM; SINDEREN, 2007) (KLEIN; DAVID, 2010) (NEISSE; WEGDAM; SINDEREN, 2008) (MANZOR; TRUONG; DUSTDAR, 2008) (TANG; YANG; WU, 2007), propõe soluções para coletar, representar, e

quantizar parâmetros de QoC. Porém, pouca atenção tem sido dada para a aplicação da qualidade de contexto em ambientes ubíquos e móveis para amenizar o impacto causado pelas limitações computacionais encontradas em dispositivos ubíquos.

Algumas soluções encontradas na literatura apresentam políticas para resolução de conflitos que são definidas em termos de parâmetros de qualidade de contexto. Esses trabalhos consideram que as políticas de qualidade devam ser utilizadas para assegurar uma melhor qualidade das informações de contexto. Porém, tais trabalhos sugerem apenas alguns parâmetros de qualidade (ex.: probabilidade de corretude e confiança). No entanto, uma abordagem baseada em QoC deve ser capaz de utilizar todos os indicadores de qualidade disponíveis para aumentar as chances de alcançar a qualidade desejada. Além disso, as políticas de qualidade encontradas na literatura não levam em consideração os parâmetros de contexto presentes nos modelos, restringindo-se apenas em incluir parâmetros de qualidade.

Outras soluções, fazem o pré-processamento dos dados no produtor de contexto realizando operações de filtragem, agregação e distribuição. Porém, essas operações não garantem a qualidade de contexto no dado de forma direta, ou seja, não são utilizados parâmetros e operações de qualidade de contexto. Contudo, mesmo que o foco desses trabalhos esteja voltado para o gerenciamento dos dados de contexto, algumas garantias de qualidade podem ser alcançadas ao se utilizar tais operações.

Assim, mesmo com as abordagens acima citadas, os provedores de contexto enviam uma grande quantidade de dados de contexto que são redundantes, conflitantes, e não confiáveis, ou não possuem a informação desejada por um consumidor de contexto. Tal informação não deve ser enviada através da rede para os consumidores de contexto, o próprio dispositivo pervasivo pode assumir alguma responsabilidade em validar o dado captado.

Dessa forma, a proposta apresentada nessa dissertação busca utilizar de uma melhor maneira o provedor de contexto (PC), enviando apenas informações úteis para os consumidores de contexto, assegurando uma melhor qualidade de contexto e otimizando o uso dos recursos do dispositivo. Assim, é aqui considerado, que a detecção e resolução de conflitos deve ser também realizada pelo provedor de contexto, com o intuito de reduzir a probabilidade de propagação de erros para as camadas de gerenciamento superiores.

## 1.2 PERGUNTA DE PESQUISA

Com a finalidade de indicar o foco da pesquisa conduzida nessa dissertação é apresentada nessa seção a pergunta de pesquisa: **Como melhorar a qualidade dos dados providos por dispositivos ubíquos móveis para as aplicações sensíveis ao contexto?**

Para responder essa pergunta, é sugerido uma abordagem de qualidade de contexto para ambientes ubíquos móveis. A proposta se baseia em um conjunto de conceitos relativos a qua-



lidade de contexto discutidos na literatura, objetivando encapsular em diferentes componentes as principais funcionalidades que asseguram a qualidade de contexto, respeitando as limitações encontradas nos dispositivos pervasivos.

A identificação dos fatores que influenciam na qualidade dos dados são descritos através de políticas de qualidade que buscam não só qualificar os dados, mas também, caracterizar o interesse das aplicações. Então, o provedor de contexto tem responsabilidade de remover redundâncias e conflitos, além de impor garantias de qualidade de contexto. Para isso, os provedores de contexto devem utilizar políticas de qualidade para realizar um processo de filtragem nos dados recém capturados, com o intuito de descartar informações que não são úteis para as aplicações. Dessa forma, a proposta de pesquisa considera que o processo de filtragem dos dados de contexto deva ser realizados antes que esses sejam enviados aos consumidores de contexto.

### 1.3 MOTIVAÇÃO

A motivação desse trabalho se deve, primeiramente, ao constante crescimento no uso dispositivo ubíquos e móveis em relação à outros equipamentos computacionais. Além disso, dispositivos ubíquos móveis são os principais habilitadores da sensibilidade ao contexto, dado a capacidade dos mesmos de captar informações do ambiente enquanto estão em movimento.

Mais especificamente, a sensibilidade ao contexto permite que serviços móveis se adaptem dinamicamente e de maneira eficiente aos objetivos de uma determinada aplicação/usuário. A utilização de uma abordagem sensível ao contexto possibilita uma melhor utilização dos recursos presentes nos dispositivos móveis, através de operações de adaptação contextual. Esse tipo de abordagem é potencialmente interessante para ambientes ubíquos e móveis, percebida a limitação encontrada nos recursos computacionais presentes nos dispositivos atuais.

Porém, de fato, a sensibilidade ao contexto só pode ser totalmente alcançada se os dados captados possuírem garantias de qualidade. A qualidade de contexto, entra como fator fundamental para o correto funcionamento das aplicações sensíveis ao contexto. Sem garantias de qualidade no dado capturado, as operações de adaptação e inferências podem divergir bastante dos objetivos reais das aplicações, inclusive realizando operações indesejadas.

Assim, considerando que a qualidade dos dados de contexto deva ser assegurada desde a captura, e percebido que poucos trabalhos focam na exploração de dispositivos móveis para assegurar a qualidade de contexto, a motivação maior desse trabalho se da pela exploração dos dispositivos ubíquos e móveis no processo de garantia de qualidade de contexto. De maneira resumida, as motivações desse trabalho podem ser descritas como:

- Dispositivos ubíquos móveis estão massivamente presentes no cotidiano, e em constante crescimento de utilização;

- A sensibilidade ao contexto possibilita a utilização otimizada de dispositivos ubíquos móveis, e permite que aplicações se adaptem dinamicamente para mudanças ocorridas no ambiente de execução;
- Sem a qualidade de contexto, os dados fornecidos pelos dispositivos e sensores podem divergir da realidade, impactando diretamente na qualidade dos serviços providos;
- Poucos trabalhos exploram a qualidade de contexto no dispositivo ubíquo móvel que é, de fato, um dos principais elementos que influenciam na qualidade das aplicações que utilizam uma abordagem sensível ao contexto.

## 1.4 OBJETIVOS

Dado o problema de pesquisa e as motivações apresentadas na seção 1.2 e 1.3, respectivamente, apresentamos nessa seção os objetivos desse trabalho para estabelecer a qualidade de contexto em dados produzidos por dispositivos ubíquos móveis e fornecidos às aplicações sensíveis ao contexto.

### 1.4.1 Objetivo Geral

O objetivo geral desse trabalho é definir uma abordagem de qualidade de contexto que filtre as informações capturadas pelos dispositivos móveis e possibilite um melhor aproveitamento dos recursos computacionais dos dispositivos pervasivos em uma arquitetura distribuída.

### 1.4.2 Objetivos Específicos

- Definir políticas de qualidade de contexto baseadas em parâmetros de qualidade e em requisitos definidos nas aplicações sensíveis ao contexto;
- Elaborar um processo de filtragem de dados de contexto baseado em políticas de QoC para dispositivos ubíquos móveis;
- Otimizar a distribuição de dados de contexto em ambientes ubíquos móveis;
- Otimizar a utilização dos recursos computacionais presentes em ambientes ubíquos móveis, dedicados a aplicações sensíveis ao contexto;
- Refinar a qualidade de contexto dos dados de contexto captados e transmitidos por dispositivos ubíquos móveis.

## 1.5 MÉTODO DE PESQUISA

Esta dissertação propõe uma abordagem para melhorar a qualidade dos dados enviados por dispositivos ubíquos móveis, baseado nos conceitos que envolvem a sensibilidade ao contexto e nas limitações impostas por dispositivos pervasivos. Assim, esta dissertação não objetiva o desenvolvimento de um produto para produção e envio dos dados, mas sim desenvolvimento de protótipos para validação dos conceitos e métodos utilizados na proposta desse trabalho.

Com relação aos objetivos, a pesquisa conduzida nessa dissertação tem caráter **exploratório**, pois não prova nenhuma teoria e nem apresenta uma gama de resultados estatísticos capazes de validar a modelagem. Porém, para áreas emergentes e muito novas a validação alcança seu maior potencial através da utilização de uma metodologia exploratória que, usufruindo de técnicas de pesquisa, estudos de caso, e argumentações, resultam na criação de novas abordagens e conhecimentos. De forma mais específica, as etapas realizadas nesse trabalho foram:

- Pesquisa do estado-da-arte nas áreas de sensibilidade ao contexto e computação ubíqua;
- Implementação de um protótipo baseado na modelagem de componentes proposta;
- Aplicação, em ambiente simulado, de um estudo de caso no protótipo implementado;
- Análise dos resultados gerados a partir do estudo de caso implementado.

## 1.6 ORGANIZAÇÃO DO TRABALHO

A dissertação é composta por seis capítulos. O *capítulo 1 - Introdução* contextualiza e descreve o problema de pesquisa, bem como apresenta a motivação e os objetivos geral e específicos.

O *capítulo 2 - Computação Móvel e Ubíqua* apresenta os conceitos relativos a computação móvel e ubíqua, que possuem influência direta na sensibilidade ao contexto.

O *capítulo 3 - Sensibilidade ao Contexto* fornece a fundamentação dos conceitos relativos a sensibilidade ao contexto, bem como exemplos de trabalhos que utilizam diferentes abordagens para representar, processar e distribuir dados de contexto.

No *capítulo 4 - ProQuali: Produtor de Contexto Consciente de Qualidade* é apresentada a proposta desse trabalho de pesquisa, que foi dividida em três seções denominadas: **políticas de qualidade de contexto, processo de filtragem e modelo do provedor de contexto**. Além disso, nessa seção são apresentados os trabalhos correlatos.

No *capítulo 5 - Ambiente e Resultados Experimentais* é apresentada infraestrutura computacional utilizada, bem como a arquitetura da implementação. Além disso, são apresentados os

pseudo-códigos utilizados. E por fim, são analisados os resultado experimentais obtidos através da simulação de uma aplicação sensível ao contexto.

Finalmente, no *capítulo 6 - Conclusões e Trabalhos Futuros* apresentamos as conclusões do trabalho e os próximos passos da pesquisa.

## 2 COMPUTAÇÃO MÓVEL E UBÍQUA

Nessa seção são contempladas a computação móvel e ubíqua sob a ótica da sensibilidade ao contexto. Apesar de soluções sensíveis ao contexto terem aparecido em diversas áreas, a sensibilidade ao contexto alcança o seu máximo potencial quando aplicada em ambientes ubíquos móveis (BELLAVISTA et al., 2013). Nesse sentido, são apresentados nesse capítulo alguns aspectos relevantes que envolvem a computação móvel e ubíqua, com o intuito de caracterizar esses grupos computacionais no escopo da sensibilidade ao contexto. Porém, primeiramente, são descritos os conceitos relativos à comunicação sem fio, dado que sem a mesma não há de ocorrer uma plena habilitação da sensibilidade ao contexto em ambientes ubíquos e móveis.

### 2.1 COMUNICAÇÃO SEM FIO

A habilidade de se comunicar em movimento, sem a utilização de fios e cabos, está em constante evolução desde a descoberta do rádio. Mais recentemente, dispositivos móveis e ubíquos possuem diversas tecnologias que permitem a comunicação sem fio (ex.: Wi-Fi, Bluetooth, 3G, 4G, ...). Com o constante crescimento desses dispositivos, novas abordagens computacionais se fazem necessárias para uma melhor utilização dos recursos computacionais disponíveis. De fato, a comunicação sem fio é um elemento chave para a exploração da computação móvel e ubíqua em sua completude. A transferência de informação entre dois ou mais pontos que não estão ligados por um condutor elétrico, possibilita a comunicação a todo tempo e em qualquer lugar. Em outras palavras, a distância geográfica deixa de ser um limitante para a criação de redes entre dispositivos ubíquos, e aspectos como mobilidade e onipresença passam a direcionar a projeção de sistemas e serviços computacionais.

Dispositivos móveis e sensores podem interagir através de diferentes interfaces para trocar informações, porém isso depende diretamente da infraestrutura de rede em que os dispositivos se encontram. Na subseção 2.1.1, apresentamos as diferentes configurações de rede que podem ser exploradas pelos dispositivos para realizar a comunicação sem fio. Na subseção 2.1.2 apresentamos de maneira abrangente as redes de sensores sem fio, que são amplamente utilizadas para a implantação de serviços sensíveis ao contexto.

#### 2.1.1 Infraestruturas de rede sem Fio

A comunicação sem fio torna viável a concepção de diferentes infraestruturas de rede, que variam de acordo com algumas características como área de alcance, tipos aplicação, e tecnologias utilizadas. Infraestruturas sem fio podem ser divididas em quatro grandes grupos: wPAN,

Rede	Exemplo	Alcance	Largura de Banda (Mbps)	Latência (ms)
WPAN	Bluetooth (IEEE 802.15.1)	10-30m	0.5-2	5-20
WLAN	Wifi (IEEE 802.11)	0.15-1.4km	2-54	5-20
WMAN	WiMax (IEEE 802.16)	5-50km	1.5-20	5-20
WWAN	Redes telefônicas GSM, 3G	Mundial	0.010-2	100-500

Tabela 1: Tipos de rede sem fio

wLAN, wMAN e wWAN, que são caracterizados na tabela 1 (COULOURIS; DOLLIMORE; KINDBERG, 2007).

- *WPAN - (Wireless Personal Area Network)* - rede pessoal sem fio. Fornece conectividade com uma pequena abrangência para dispositivos móveis, ou seja, que estejam fisicamente próximos. É interessante para interligar teclados, impressoras, telefones móveis, agendas eletrônicas, computadores de mão, câmeras fotográficas digitais, mouses, e até sensores de ambiente. Os padrões mais utilizados para estabelecer a comunicação são ondas de raio infravermelho.
- *WLAN - (Wireless Local Area Networks)* - redes locais sem fio - rede para fornecer conectividade para dispositivos móveis dentro de uma área (casa, prédio) e prover acesso à Internet ou outra rede fixa (LAN, WAN). Possui variantes de padrões, oferecendo largura de banda de 10 e 100 Mbps com abrangência de até 1,5 quilômetros.
- *WMAN - (Wireless Metropolitan Area Network)*- rede metropolitana sem fio - possuem, geralmente, maior abrangência que as WLAN, por exemplo, uma cidade ou região metropolitana.
- *WWAN - (Wireless Wide Area Networks)* - rede de longa distância sem fio - é por meio deste tipo de rede que os usuários de telefonia móvel têm acesso com ampla cobertura, seja por antenas ou sistemas de satélite. É importante salientar que estas redes oferecem taxas de transmissão de dados relativamente baixas quando comparadas com as demais redes.

Diferentemente das infraestruturas de rede cabeadas, as redes sem fio são altamente vulneráveis no que diz respeito a conectividade. A desconexão de dispositivos pervasivos, em especial os que possuem a característica de mobilidade, podem ocorrer por diferentes razões, como por exemplo:

- alta variabilidade na qualidade da conexão.

- necessidade de economia de recursos (ex.: bateria) ou descarga completa da fonte de energia.
- *hand-offs*, podem ocorrer saltos na força do sinal quando se desconecta do ponto de acesso do sinal mais fraco e conecta-se em outro de sinal forte)
- interferências, como por exemplo, a interferência eletromagnética que pode ocorrer com ondas de microondas, sinais de motores ou outros equipamentos elétricos.
- sombreamento, isto é, variação de sinal causada pela obstrução de objetos tais como montanhas, prédios, outdoors, móveis ou paredes; pode ocorrer também quando entre duas estações existe uma terceira que impede a comunicação entre as duas primeiras.

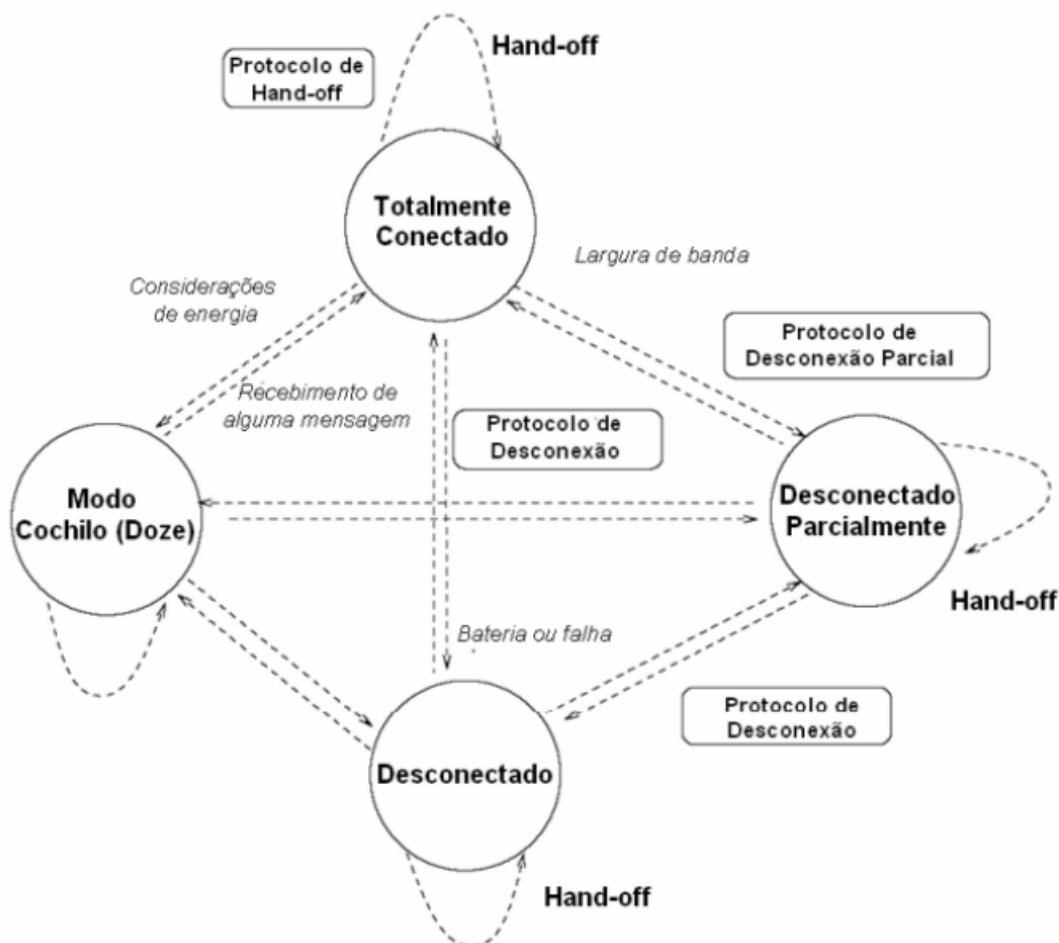


Figura 1: Estados de conexão dos dispositivos em ambientes móveis (PITOURA; SAMARAS; SAMARAS, 1998)

Conforme explicitado em (FORMAN; ZAHORJAN, 1994), falhas nas redes são uma grande preocupação para computação móvel e ubíqua. Nos projetos de aplicações para computação pervasiva, é preciso analisar quais serão os investimentos feitos para amenizar as limitações do ambiente: disponibilizar mais recursos na rede, tentar prevenir desconexões, ou possibilitar ao

sistema lidar com desconexões de maneira refinada. Porém, nem tudo pode ser transparente para o usuário; nestes casos, boas interfaces (ex.: naturais) podem ajudar fornecendo *feedback* sobre quais operações estão indisponíveis em função da desconexão.

Segundo (PITOURA; SAMARAS; SAMARAS, 1998), para ambientes ubíquos móveis, os dispositivos podem se encontrar em diferentes estados de conexão: totalmente conectado, parcialmente conectado, desconectado e modo “cochilo”, a figura 1 apresenta a esquematização desses estados. Dado que desconexões parciais e totais são frequentes, (PITOURA; SAMARAS; SAMARAS, 1998) sugere que essas não deveriam ser tratadas como falhas. Pelo contrário, um nó móvel deveria ser capaz de operar sob um baixo grau de conexão.

A comunicação sem fio, então, é de vital importância para o estabelecimento da computação móvel e ubíqua, pois sem ela não seria possível realizar a troca de informações entre dispositivos de maneira escalável e sem limitações geográficas. As diferentes infraestruturas sem fio permitem que os dispositivos sejam explorados, respeitando as limitações dos mesmos, por uma gama de aplicações em diversos cenários. Dessa forma, podem ser criadas redes sem fio para direcionar o projeto de serviços móveis e ubíquos e minimizar as falhas (ou características) inerentes a comunicação sem fio.

### 2.1.2 Redes de Sensores sem Fio

Redes de sensores sem fio são redes densas compostas por pequenos sensores, de baixo custo, que coletam e divulgam dados sobre o ambiente atual. Tais redes facilitam o monitoramento e controle de ambientes físicos a partir de locais remotos com uma melhor precisão. Cada sensor tem a capacidade de realizar uma quantidade limitada de processamento. Porém, quando coordenados corretamente, possuem a capacidade de caracterizar um determinado ambiente físico de maneira detalhada. Assim, uma rede de sensores sem fio pode ser descrita como um conjunto de sensores coordenados para executar alguma ação específica. Diferentemente das redes tradicionais, redes de sensores dependem de uma implementação densa e coordenada para realizar um conjunto de tarefas (BHARATHIDASAN; PONDURU, 2002).

Nesse sentido, são necessárias novas abordagens para se trabalhar com sensores. Como explicitado em (AKYILDIZ et al., 2002), apesar dos diversos algoritmos e protocolos existentes para as tradicionais redes ad-hoc sem fio, as características das redes de sensores sem fio demandam outras implementações. As principais diferenças entre essas duas configurações de rede são:

- O número de nós existentes em uma rede de sensores sem fio pode ultrapassar em várias ordens de magnitude a quantidade de nós existentes em uma rede ad-hoc.
- Sensores são usualmente densos, i.e., dedicados a computar o máximo que podem.



- Sensores são propensos a falhas de variados tipos.
- A topologia de uma rede de sensores muda frequentemente.
- Sensores possuem limitações energéticas, de processamento, e de armazenamento.
- Sensores utilizam, principalmente, o paradigma de comunicação *broadcast*, enquanto redes ad-hoc exploram, usualmente, comunicação ponto-a-ponto.

Outra diferença bem interessante consiste no fato de que redes de sensores podem ser aplicadas em uma grande variedade de cenários. Por exemplo, monitoramento de ambientes que envolve: monitoramento do ar, água e solo, manutenção baseada em uma determinada condição, monitoramento de um habitat (determinação da população de espécies de plantas e animais e seus respectivos comportamentos), detecção sísmica, vigilância militar, controle de estoque, espaços inteligentes, dentre outros. De fato, devido a natureza pervasiva dos micros sensores, redes de sensores têm o potencial de revolucionar a própria maneira de entender e construir sistemas físicos complexos (ESTRIN et al., 1999). Apesar das diversas aplicações, a implantação de redes de sensores apresenta uma série de desafios técnicos devido, principalmente, aos seguintes fatores:

- *Implementações Ad-Hoc:* A rede de sensores pode ser implementada em lugares sem qualquer infraestrutura. Uma maneira usual de conceber essas redes em florestas, por exemplo, é “jogando” os sensores de aviões. Nessas situações, fica a cargo dos sensores identificar a conectividade e sua distribuição.
- *Manutenibilidade:* Em grande parte dos casos, uma vez implementadas, as redes de sensores não possuem intervenção humana. Assim, os próprios sensores são responsáveis pela reconfiguração em caso de quaisquer mudanças significativas.
- *Fonte de energética:* Os sensores não estão ligados a qualquer fonte de energia. Existe apenas uma fonte finita de energia, que deve ser utilizada de forma ótima para o processamento e a comunicação. Um fato interessante é que a comunicação domina o consumo de energia. Assim, para que se tenha uma melhor performance energética, a comunicação deve ser otimizada o tanto quanto possível.
- *Dinamicidade:* Um sistema que utiliza uma rede de sensores deve ser adaptável as alterações da conectividade (por exemplo, devido à adição de mais sensores ou falhas nos mesmos), bem como alterações devido a estímulos do ambiente.

Para fornecer uma ideia geral de que tipos de arquiteturas e sistemas operacionais são apropriados para redes de sensores sem fios, são apresentados exemplos. (SRISATHAPORNPHAT;

JAIKAE0; SHEN, 2000) propõe uma arquitetura de *middleware* chamada SINA (*Sensor Information Networking Architecture*), a camada de mais baixo nível da arquitetura possui dois mecanismos. No primeiro, chamado de *Clusterização Hierárquica*, os sensores são organizados em uma hierarquia com base em seus níveis de energia e de proximidade, constituindo um *cluster* de sensores. Então, um sensor é eleito o mestre responsável por executar várias funções, inclusive a capacidade de re-inicialização do sistema em caso de falha.

Já o segundo foi denominado *Nomeação baseada em atributos*. Devido à grande quantidade de dispositivos em uma rede de sensores, é muito difícil captar informações de um nó individualmente. Portanto, é improvável que requisições como: qual é a temperatura do sensor de ID 101, sejam solicitadas. Em vez disso, os usuários estão mais interessados em consultar, por exemplo, qual a área tem temperatura superior a 30 graus Celsius, ou qual área tem a temperatura mais alta, ou a temperatura média no quadrante Sudeste. Assim, a tupla [tipo=temperatura, localização=N, temperatura=40] refere-se a todos os sensores localizados no quadrante norte e que possuem uma leitura de temperatura de 40 graus Celsius. Dessa forma as aplicações podem acessar a informação diretamente pelo nome. Essa abordagem também é interessante pois elimina a necessidade de manter serviços de mapeamento dos sensores, o que causaria uma sobrecarga.

Como exemplo de sistema operacional, TinyOS (LEVIS et al., 2005) é um sistema operacional baseado em componentes projetados especialmente para rede de sensores. (CULLER et al., 2001) descreve um modelo de comunicação de mensagens utilizando o TinyOS, que pode ser usado como base para o desenvolvimento de funcionalidades de rede de mais alto nível. Com o intuito de esclarecer o que deve ser levado em consideração para conceber um sistema operacional para sensores, é preciso discretizar os subsistemas que compõem um sensor único. (RAGHUNATHAN et al., 2002) divide sensores quatro subsistemas:

- subsistema **computacional**: é composto por um microprocessador (unidade microcontroladora, MCU) que é responsável pelo controle dos sensores e a execução dos protocolos de comunicação. MCU's, usualmente, funcionam em diferentes modos de operação para fins de gerenciamento de energia. Porém, a alternância entre os modos de operação envolve, também, consumo de energia, demandando uma análise do tempo de vida da bateria antes da troca do modo de operação.
- subsistema de **comunicação**: é composto pelas interfaces de comunicação disponíveis no sensor, que podem ser desde uma comunicação de rádio de curto alcance até interfaces mais robustas para comunicação a longo alcance.
- subsistema **sensorial**: é composto por um conjunto de sensores e atuadores que conectam o dispositivo com o mundo real. Captura informações como: temperatura, umidade movimento de veículos, luminosidade, pressão, composição do solo, nível de som, presença

ou falta de objetos, níveis de desgaste de equipamentos, e as características atuais como velocidade, direção, e tamanho do objeto.

- subsistema de **energia**: é composto pela bateria presente no dispositivo. O tempo de vida da bateria é um fator crítico para a concepção de redes de sensores sem fio, dado que essa é a única fonte de energia presente. Por essa razão, a utilização dos recursos nos subsistemas deve ser otimizada. De fato, a vida útil de uma rede de sensores pode ser significativamente aumentada se o sistema operacional, a camada de aplicação e os protocolos de rede forem projetados para ter ciência do consumo de energia.

Redes de sensores sem fio são infraestruturas que tem a capacidade de sentir o ambiente e divulgar as informações relativas a ele. Porém, as limitações dos dispositivos devem ser levadas em consideração para que possam ser atingidos os objetivos das aplicações. Essa modalidade de rede torna possível ter consciência do ambiente de execução, e tomar decisões baseadas nessas informações, ou seja, redes de sensores sem fio são altamente desejáveis para aplicações sensíveis ao contexto. Nas próximas seções são descritas as características de distribuição de dados e limitações nos dispositivos ubíquos e móveis, e que impactam diretamente na implantação de sistemas sensíveis ao contexto.

## 2.2 DESAFIOS DA COMPUTAÇÃO UBÍQUA E MÓVEL

“As tecnologias mais profundas são aquelas que desaparecem. Entram no tecido da vida cotidiana até que passam a ser imperceptíveis” (WEISER, 1991). Com essas palavras que Weiser inicia seu artigo que introduz as ideias sobre a computação ubíqua. Dessa forma, podemos pensar que ambientes ubíquos móveis consistem de um grande número de agentes autônomos que trabalham juntos com o intuito de transformar espaços físicos em ambientes interativos e inteligentes. Para tal, é preciso que os agentes ubíquos capturem informações do ambiente, processem essas informações e divulgue para outros agentes, de maneira transparente ao usuário. Ou seja, a computação ubíqua se refere a onipresença dos dispositivos computacionais, que interagem com usuários através de interfaces naturais (ex.: voz, gestos, movimentação dos olhos,...) no cotidiano.

A computação ubíqua também pode ser caracterizada como um advento da computação distribuída e da computação móvel. (SATYANARAYANAN, 2001) apresenta os principais problemas de pesquisa relativos a computação ubíqua e que são derivados desses outros paradigmas computacionais, veja figura 2.

Assim, para que seja possível criar de sistemas ubíquos transparentes, invisíveis e com interfaces naturais, é preciso primeiro propor soluções para alguns problemas de implementação. Nas próximas seções são apresentados alguns desafios que devem ser superados para que seja

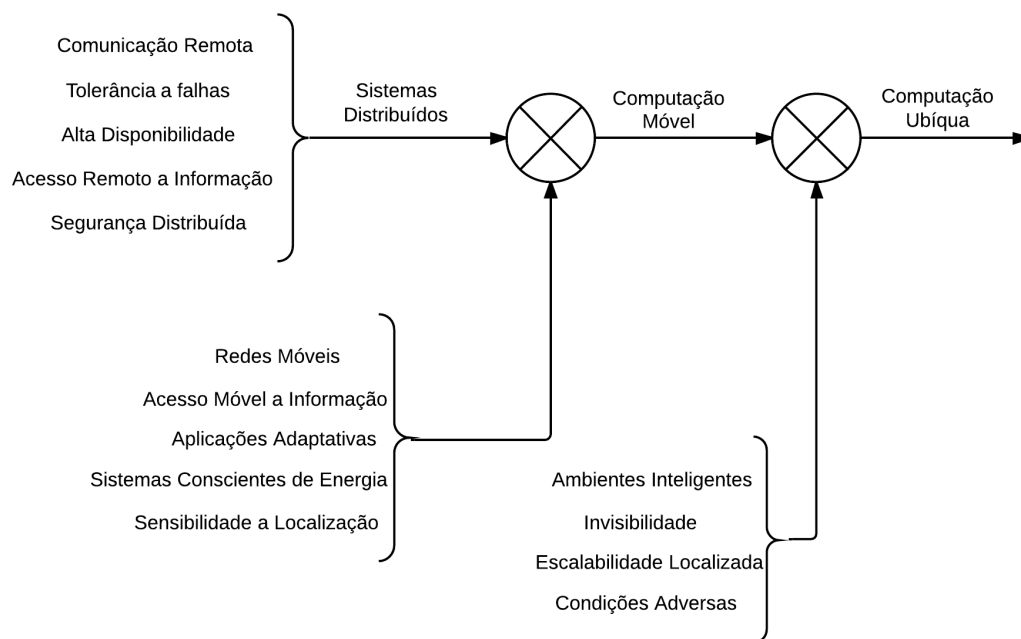


Figura 2: Taxonomia dos problemas de pesquisa na computação ubíqua (SATYANARAYANAN, 2001)

possível o desenvolvimento de sistemas verdadeiramente ubíquos, e como a sensibilidade ao contexto poderia ser aplicada para minimizar esses problemas.

### 2.2.1 Adaptabilidade

A capacidade de se adaptar dinamicamente a situação atual do ambiente é uma característica altamente desejada em ambientes ubíquos móveis, principalmente, quando existe uma discrepância entre os recursos disponíveis e a demanda por esses. Os recursos em questão podem ser largura de banda, fonte de energia, capacidade de processamento, memória, e etc. Além disso, as aplicações implementadas em ambientes ubíquos móveis também devem possuir a capacidade de se adaptar as necessidades do usuário, permitindo, por exemplo, a implantação de interfaces mais transparentes para o usuário. Nesse sentido, podemos utilizar a sensibilidade ao contexto para explorar aspectos declarados pelos os usuários, além de técnicas como aprendizado de máquina, para descobrir como o sistema melhor se adapta para aquela determinada situação e usuário. (KAKOUSIS; PASPALLIS; PAPADOPOULOS, 2010)

Uma abordagem sensível ao contexto torna possível a adaptabilidade do sistema tanto para as limitações de recursos computacionais, quanto para as necessidades dos usuários. Isso se justifica pelo fato de que uma abordagem verdadeiramente sensível ao contexto tem consciência do ambiente como um todo, não só dos aspectos físicos, mas também, dos aspectos computacionais e do usuário. Dessa forma, os serviços e aplicações podem utilizar as informações de contexto para se adaptar dinamicamente a situação atual do sistema, permitindo a alteração do estado das diferentes camadas que compõem o sistema, em acórdância com demandas atuais

dos recursos computacionais (ex.: o dispositivo não está enviando dados, pode-se limitar a sua conexão para economizar bateria) e aos requisitos dos usuários (ex.: o sistema pode inferir quais são as funções mais utilizadas pelo o usuário e dar prioridade de recursos para essas).

### 2.2.2 Gerenciamento de Energia

O consumo de energia por parte dos dispositivos ubíquos móveis vem sendo fonte de pesquisa de diversos trabalhos como (BALASUBRAMANIAN; BALASUBRAMANIAN; VENKATARAMANI, 2009) (BERNAL et al., 2010) (MURMURIA et al., 2012) e (YOON et al., 2012). A eficiência energética é, de fato, um fator limitante na implantação de sistemas ubíquos móveis, sem um correto gerenciamento dos recursos energéticos disponíveis, as operações realizadas pelos dispositivos podem não funcionar corretamente, ou até mesmo não serem acessíveis, comprometendo o funcionamento do sistema. Por essa razão, as aplicações e serviços implantados em ambientes ubíquos devem estar cientes do consumo de energia em tempo de execução.

(BALASUBRAMANIAN; BALASUBRAMANIAN; VENKATARAMANI, 2009) apresenta como algumas interfaces de comunicação (3G, WiFi e GSM) impactam no consumo de bateria. A comunicação costuma ser a principal variável no consumo de recursos energéticos de dispositivos ubíquos móveis. Nesse sentido, a sensibilidade ao contexto pode otimizar a utilização dos recursos energéticos através da redução no envio de dados de contexto conflitantes, redundantes e não desejados pelas aplicações. Além disso, com a redução dos dados enviados, menos dados serão processados pelos dispositivos otimizando também o consumo de energia.

### 2.2.3 Transparência

A Transparência, sob a ótica da computação distribuída, pode ser definida como a capacidade de esconder dos usuários o fato de que os processos e recursos computacionais não estão concentrados em um único dispositivo (TANENBAUM; STEEN, 2002). (10746.1-4, 1995-1998) apresenta diversos tipos de transparências que contemplam os sistemas distribuídos, a seguir são listados os tipos que impactam diretamente na sensibilidade ao contexto.

- *Acesso*: Esconde as diferentes representações de dados e como o recurso é acessado;
- *Localização*: Esconde onde um determinado recurso está localizado;
- *Migração*: Esconde que o recurso pode se locomover para outra localização;
- *Realocação*: Esconde que o recurso pode se locomover enquanto estiver em uso;

- *Falhas*: Esconde a falha e a recuperação de um determinado recurso;

Assim, é considerado nesse trabalho que a garantia de transparência deve ser de responsabilidade do *middleware* de contexto. Pois no âmbito da sensibilidade ao contexto, o *middleware* é responsável pelo provimento dos dados de contexto as aplicações, sem que as mesmas saibam diretamente quais recursos fornecem os dados. De maneira mais direta, aqui é defendida a ideia que parte do *middleware* de contexto esteja presente nos dispositivos pervasivos que compõem a infraestrutura computacional, para que, as questões relativas a transparência sejam de fato contempladas. Para um melhor entendimento, apresentamos a seguir diretrizes que contemplam a utilização dos recursos ubíquos para o provimento de serviços sensíveis ao contexto mais transparentes.

Para a obtenção de uma melhor transparência de acesso, os dispositivos pervasivos devem ser capazes de fornecer dados padronizados para o sistema, ou seja, antes do envio, o dado deve ser padronizado segundo as especificações do *middleware*. As transparências de localização, migração e realocação, são na verdade não desejáveis em ambientes totalmente sensíveis ao contexto, dado que saber aonde um dispositivo se encontra em um determinado momento é uma informação relevante. Porém, a ideia pode ser projetada para “transparência de comunicação”, onde independentemente da localização do dispositivo, o dado deve ser enviado sem que os usuários saibam das alterações ocorridas para realizar a comunicação. Assim, é de vital importância que os dispositivos pervasivos possuam mecanismos para lidar com situações como, por exemplo, desconexões temporárias, que impactam diretamente no correto funcionamento dos serviços sensíveis ao contexto.

Por fim, dispositivos pervasivos em configurações sensíveis ao contexto, possuem basicamente três funções: captar, processar e enviar dados de contexto. Esconder e recuperar falhas dessas operações, sem a percepção do usuário, é uma tarefa quase que impossível em sistemas reais. Apesar disso, algumas medidas podem ser tomadas para que as falhas sejam menos impactantes.

## **2.2.4 Capacidade Computacional dos Dispositivos**

Apesar da capacidade computacional dos dispositivos ser diretamente dependente das tecnologias utilizadas na implementação dos mesmos, deve-se pontuar que atualmente esses dispositivos possuem uma capacidade reduzida de processamento, memorização, armazenamento e comunicação de dados, em relação a outros equipamentos computacionais (PC's, mainframes ...). Dessa forma, sistemas que são suportados por infraestruturas compostas por dispositivos ubíquos móveis devem estar cientes das limitação desses equipamentos, buscando uma utilização mais otimizada dos recursos computacionais disponíveis.

Assim sendo, aplicações e serviços sensíveis ao contexto podem ponderar a utilização

dos recursos disponíveis em um dispositivo, através das informações de contexto captadas pelos próprios dispositivos. Por exemplo, se um determinado serviço sensível ao contexto perceber que uma quantidade significativa dos dados recentemente enviados possuem uma carga semântica parecida, ou seja, são redundantes, esse serviço pode interromper o envio até que uma nova informação apareça. Além disso, caso a aplicação realize um processamento antes de enviar os dados, para os dados redundantes não há a necessidade de um re-processamento, o dispositivo pode armazenar o dado e realizar o re-envio, se necessário.

Por outro lado, apesar das limitações que envolvem dispositivos ubíquos móveis, algumas operações podem ser realizadas nesses dispositivos para que não ocorra uma subutilização dos recursos computacionais explorados pelos sistemas. Operações de filtragem de dados, operações de armazenamento, e operações de gerenciamento de dados são exemplos de funcionalidades que podem ser exploradas nos dispositivos atuais se repetida a limitação dos mesmos. Além disso, a medida que as tecnologias evoluem os dispositivos ubíquos móveis também crescem em capacidade computacional, favorecendo a concepção de sistemas que explorem cada vez mais a capacidade desses dispositivos.

### 2.3 DISTRIBUIÇÃO DE DADOS EM AMBIENTES UBÍQUOS MÓVEIS

A concepção de serviços sensíveis ao contexto em ambientes ubíquos móveis é uma tarefa complexa que requer um entendimento profundo de muitos detalhes tecnológicos e de operações não triviais abrangendo diferentes camadas e dependentes da plataforma de execução. Os diversos esforços necessários para gerenciar todos esses aspectos tecnológicos podem ser responsabilizados pela lentidão na implantação de serviços sensíveis ao contexto em sistemas ubíquos móveis (CHEN; KOTZ et al., 2000). Conseqüentemente, para enfrentar todas essas questões e facilitar a difusão de serviços sensíveis ao contexto, existe uma necessidade de soluções de *middlewares* sensíveis ao contexto destinadas a abordar de forma transparente todas as principais fases de gestão envolvidos na provisão de contexto para as camadas de serviços, dentre elas a distribuição dos dados. De maneira mais específica, serviços sensíveis ao contexto devem, somente, produzir e enviar dados de contexto e declarar especificamente quais dados deseja receber para *middleware*. Enquanto, por outro lado um *middleware* de contexto tem a função de transparentemente executar operações de gerenciamento para distribuir dados de contexto.

Mais explicitamente, a distribuição de dados de contexto é definida como uma função do *middleware* que torna possível a injeção de dados de contexto no sistema e a entrega automática para todas as entidades que expressem interesse nesses dados. Como destacado em (BELLAVISTA et al., 2013), pode-se classificar a distribuição de dados em dois grupos principais:

- *Distribuição de dados não-informada*: essa abordagem simplesmente encaminha dados de contexto, de acordo com as necessidades de contexto expressas pelos elementos

móveis, onde os dados são cegamente entregues sem uma inspeção prévia do seu conteúdo.

- *Distribuição de dados informada*: agrupa outras abordagens que se aproveitam das informações disponíveis nos dados de contexto para se adaptar dinamicamente e autogerir o próprio processo de distribuição.

As abordagens de distribuição informadas são mais recentes na literatura e, ainda, muito especializadas. Essas abordagens se adequam a cenários de implantação que não utilizam infraestruturas de rede sem fio fixas, e nesse sentido, a exploração da sensibilidade ao contexto é de crucial importância para a efetividade e eficiência a distribuição de dados de contexto nos ambientes ubíquos móveis.

Os primeiros trabalhos que surgiram explorando a sensibilidade ao contexto, costumeiramente, eram implementados em redes de pequena escala, como *smart homes* e *smart spaces*. Nos últimos anos, um número crescente de sistemas buscam exigindo fornecimento de dados de contexto para redes sem fio de longa distância. Nesse sentido, a distribuição de dados de contexto passa a ser um dos fatores que necessitam de uma atenção especial devido ao tamanho dos sistemas emergentes. Em especial, definir explicitamente como realizar uma distribuição eficaz e eficiente, com respeito aos aspectos da qualidade de contexto. Para tal (BELLAVISTA et al., 2013) discretiza alguns requisitos, são eles:

- **Desacoplamento entre a produção e consumo dos dados**: a distribuição de dados de contexto deve rotear transparentemente os dados para todos os consumidores interessados naquela informação e conectados ao sistema móvel. Com a finalidade de fortalecer a escalabilidade sistema e a disponibilidade dos dados, a produção e o consumo dos dados pode ser realizado em diferentes espaços de tempo (desacoplamento temporal). Além disso, consumidores e fontes não precisam necessariamente conhecer uns aos outros (desacoplamento espacial). Resumidamente, a comunicação deve ser assíncrona e anônima entre produtores e consumidores de dados de contexto.
- **Adaptabilidade em ambientes heterogêneos e móveis**: a distribuição de dados de contexto deve suportar a heterogeneidade existente em cenários móveis de comunicação sem fio. Dispositivos móveis requerem serviços sensíveis ao contexto enquanto se movem entre diferentes configurações de rede, introduzindo assim variações nas necessidades contextuais. Assim, a distribuição de dados deve prontamente adaptar-se a essa mobilidade, distribuindo apenas dados de contexto requeridos naquele exato momento. Da mesma forma, os serviços sensíveis ao contexto devem estar em conformidade com os sistemas heterogêneos, incluindo os dispositivos com diferentes capacidades computacionais e os padrões e modalidades sem fio. Portanto, a adaptação aos recursos atualmente disponíveis é fundamental para evitar a saturação do sistema.



- **Visibilidade dos dados de contexto em escopos definidos:** a distribuição de dados de contexto deve introduzir, preservar e aplicar diferentes escopos de visibilidade para dados de contexto. De fato, dados de contexto possuem tipicamente um escopo limitado quanto a visibilidade, que depende dos princípios de localidade física e lógica. Por exemplo, os recursos computacionais (ex.: redes disponíveis, outros dispositivos no ambiente, ...) que estão em um determinado local, são visíveis somente aos nós que se encontram nesse mesmo local (localidade física). Similarmente, dados de contexto associados a um usuário em um determinado evento devem ser visíveis somente aos participantes desse evento (localidade lógica). Em outras palavras, dados de contexto possuem intrinsecamente escopos visibilidade, cujo os quais a distribuição de dados de contexto deve explorar para evitar sobrecargas desnecessários no gerenciamento dos dados.
- **Distribuição de dados baseada em princípios da qualidade de contexto:** a distribuição de dados de contexto deve impor restrições de qualidade para realizar uma correta gestão do sistema. Restrições de QoC especificam a qualidade do dado a ser recebido. Além disso, considerando que sistemas sem fio reais lidam com frequentes mudanças topológicas, garantias limitadas na entrega dos dados, e desconexões temporárias, restrições de QoC permitem a entrega dos dados com garantias de pontualidade e confiança. Adicionalmente, a distribuição de dados de contexto pode ser implementada em arquiteturas distribuídas com vários servidores, onde cada um possui o seu repositório local para armazenar, processar e rotear dados de contexto. Sem protocolos de coordenação adequados, diversos dados de contexto podem estar redundantes e conflitantes dentro do sistema. Dessa forma, onde a consistência nos dados de contexto pode ser custosa para gerenciar, é preferível utilizar abordagens de melhor-esforço baseadas em restrições de qualidade de contexto.
- **Gerenciamento do tempo de vida de um dado de contexto:** a distribuição de dados de contexto deve lidar com o tempo de vida do dado, da sua construção até sua destruição (CHANG; SHIN; CHUNG, 2007). Para tal finalidade, técnicas de agregação e filtragem devem ser implementadas para reduzir a sobrecarga no gerenciamento dos dados de contexto. Fora isso, a distribuição de dados de contexto deve definir alguns níveis de disponibilidade (com um certa probabilidade), influenciando o grau de distribuição e replicação dos dados no sistema. Dessa forma, o sistema de distribuição deve ser capaz de auto-gerenciar os processos de distribuição.

Dado essas características desejáveis, (BELLAVISTA et al., 2013) sugere uma arquitetura para distribuição de dados de contexto, que é apresentada na figura 3. Essa arquitetura visa definir quais são os principais processos que ocorrem entre a produção e o consumo de dados de contexto. Dessa maneira podemos descrever a arquitetura através de três componentes

principais: fontes de contexto, consumidores de contexto e mediador de dados de contexto.

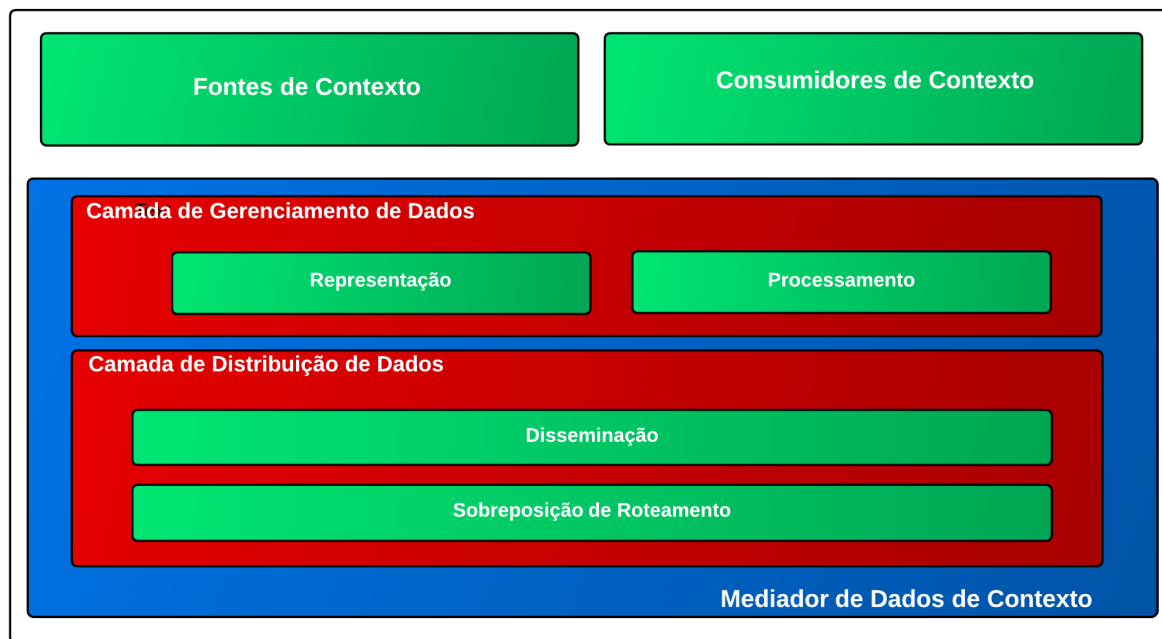


Figura 3: Arquitetura lógica do mediador de dados de contexto (BELLAVISTA et al., 2013)

Mais especificamente, o componente fontes de contexto engloba todas as operações de acesso a sensores e que habilitam a distribuição dos dados de contexto. Já, o componente consumidores de contexto trata das requisições que expressam os dados de contexto desejados pelas aplicações e serviços. E por fim, o componente denominado mediador de dados de contexto faz a interação entre os fontes e consumidores de dados de contexto, estabelecendo a forma de representação, processamento e distribuição dos dados. A descrição dos componentes da arquitetura que realizam as operações em dados de contexto é feita detalhadamente no capítulo 3, em especial nas seções 3.2 e 3.3.

## 2.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo foram apresentados os conceitos da computação móvel e ubíqua que possuem um impacto direto na implantação de sistemas sensíveis ao contexto. A comunicação sem fio é determinante para o estabelecimento de sistemas sensíveis ao contexto, pois permite que dispositivos se comuniquem “a qualquer hora em qualquer lugar”. Porém, alguns desafios surgem pela utilização desse tipo de comunicação como, por exemplo, a autonomia energética do dispositivo e situações de desconexão. Assim, na seção 2.3 desse capítulo é apresentado um modelo para provedores de dados de contexto cujo o principal objetivo é minimizar o impacto dos desafios existentes para captar e disseminar dados de contexto, levando em consideração os desafios da computação móvel e ubíqua.

### 3 SENSIBILIDADE AO CONTEXTO

Nesse capítulo são apresentados os conceitos relativos a sensibilidade ao contexto, bem como os aspectos relacionados ao provimento de dados de contexto. Esse capítulo não apenas apresenta conceitos e aspectos, mas também explicita de forma organizada as diferentes abordagens utilizadas pelos trabalhos encontrados na literatura para prover dados de contexto. Assim sendo, esse capítulo proporciona um estudo detalhado das soluções que abordam a sensibilidade ao contexto, desde de conceituação de contexto até soluções que propõem *middlewares* de contexto.

#### 3.1 DEFINIÇÃO DE CONTEXTO

Para se entender a sensibilidade ao contexto é necessário, primeiramente, definir o conceito de contexto. Contexto ainda é um conceito vago que identifica os aspectos que um projetista considera útil para modelar e descrever o ambiente em que um determinado serviço será projetado (BELLAVISTA et al., 2013). Entretanto, a composição das diferentes definições encontradas na literatura possibilitam o entendimento desse conceito e da sua aplicabilidade em diferentes esferas da computação.

O primeiro trabalho que introduz o termo “sensibilidade ao contexto”, Schilit e Theimer (SCHILIT; ADAMS; WANT, 1994) referênciam contexto como localização, objetos e pessoas ao redor, e mudanças nesses objetos. No mesmo caminho, Brown (BROWN; BOVEY; CHEN, 1997) define contexto como localização, pessoas em volta do usuário, hora do dia, estação do ano, temperatura, e *etc.* Outras definições que apontam ideias parecidas são as encontradas em (RYAN; PASCOE; MORSE, 1998) e (DEY, 1998). Essas definições, apesar de fornecerem elementos que sugerem o que efetivamente é contexto, são baseadas em exemplos e por conseguinte são difíceis de serem aplicadas em diversas aplicações.

A ideia de contexto está concentrada em determinar basicamente quatro aspectos sobre as entidades: *quem?*, *aonde?*, *quando?* e *o que?*, e usar essas informações para determinar *porque?* uma situação está ocorrendo. Nesse perspectiva algumas definições mais abrangentes surgiram, como em (DEY, 2000) que define contexto como sendo qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade. Outra definição que vai nesse sentido é a de (BOLCHINI et al., 2009), onde o contexto é um conjunto de variáveis que pode ser do interesse de um agente e influencie suas ações. Essas definições caracterizam de uma maneira mais ampla o contexto, porém não de forma operacional, isto é, não indicam categoricamente onde as informações de contexto se enquadram.

Já o trabalho de (ZIMMERMANN; LORENZ; OPPERMANN, 2007), cita que os ele-

mentos para a descrição de uma informação de contexto encontram-se em cinco categorias: individualidade, atividade, localização, tempo e relações. Essa definição apresenta categorias onde as informações de contexto podem ser enquadradas, e conseqüentemente, quais aspectos da aplicação são sujeitos a variação de contexto. Porém, esse não foi o primeiro trabalho que trouxe uma perspectiva operacional para a definição de contexto. (CHEN; KOTZ et al., 2000) definiram contexto como um espaço quadridimensional composto por: *contexto computacional*, *contexto físico*, *contexto temporal* e *contexto do usuário*, onde:

- Contexto computacional: lida com todos os aspectos técnicos relacionados às capacidades e recursos computacionais;
- Contexto físico: agrupa todos os aspectos que representam o mundo real e que são acessíveis por meio de sensores e/ou recursos implantados no ambiente;
- Contexto temporal: captura a dimensão de tempo, como hora do dia, semana, mês, ano, estação do ano, de qualquer atividade realizada no sistema;
- Contexto do usuário: contém aspectos de contexto de alto nível relacionados com a dimensão social dos usuários, tais como o perfil do utilizador, as pessoas próximas, e atual situação social.

O contexto computacional possui dois objetivos principais. Primeiro, expressar toda heterogeneidade que normalmente está presente em ambientes móveis, como diferentes capacidades computacionais dos dispositivos e aspectos de conectividade (BARTOLINI; RUGGIERO; BENINI, 2009) e (CERI et al., 2007). E em segundo lugar, ele também considera os diferentes recursos que um dispositivo móvel encontra enquanto está em *roaming* (SCHILIT; ADAMS; WANT, 1994).

A localização geográfica do usuário/dispositivo é um exemplo básico de contexto físico. Outros exemplos são: nível de barulho, temperatura, luminosidade, e condição do trânsito (KIM; LEE, 2006a). Devido a sua natureza, o contexto físico é intrinsecamente propenso a erros de medição, dado as várias fontes de incorreções e imprecisões, e a natureza estocástica dos processos físicos. Algumas aplicações já exploram as informações de contexto físico: indicar rotas alternativas quando a rota principal está congestionada, ou alterar a temperatura de ar-condicionados caso a temperatura se eleve ou abaixe abruptamente.

As informações de contexto temporal não indicam apenas quando uma determinada atividade aconteceu, são utilizadas, também, para agendar atividades. Dessa maneira, podemos enquadrar os itens dessa dimensão de contexto em dois grupos principais: esporádicos e periódicos. Eventos esporádicos são acionados ocasionalmente, podendo ocorrer apenas uma vez. Já eventos periódicos descrevem acontecimentos esperados que se apresentam de uma maneira repetida e previsível. Além disso, essas duas categorias podem ser combinadas para

construir eventos de contexto complexos baseando-se tanto na sequência de eventos, como no número de eventos em um período de tempo específico (ABOWD et al., 1999).

No contexto do usuário, como mencionado em (EUGSTER; GARBINATO; HOLZER, 2009), cada nó (usuário/dispositivo) possui a sua dimensão individual, provinda da sua própria visão egocêntrica (como o perfil do usuário e suas preferências), e a sua dimensão social, provinda da consciência de ser parte de todo um sistema (como outros usuários ao redor, ...). Alguns sistemas usam padrões de co-localização para inferir interesses comuns e recomendar possíveis amigos, enquanto outros inferem a situação atual do usuário, por exemplo, desligam o tom de toque de telefone celular durante um jantar de negócios. Nesse trabalho, então, adotamos a definição proposta por (CHEN; KOTZ et al., 2000), pois ela cobre os principais aspectos de contexto com uma classificação bastante direta e simples e, além disso, possui semelhanças significativas com a definição apresentada em (SCHILIT; ADAMS; WANT, 1994), que é bem aceita pela comunidade.

Assim, considerando todas as dimensões do contexto mencionadas, diferentes comportamentos sensíveis ao contexto podem ser modelados para adaptar os serviços e torná-los mais adequados aos usuários, colocando os mesmos em conformidade com as características atuais do ambiente de execução (BELLAVISTA et al., 2013). Ou seja, serviços e aplicações sensíveis ao contexto possuem a capacidade de aptar-se através de informações que caracterizam um contexto. Nesse sentido, as aplicações sensíveis ao contexto, projetadas em infraestruturas móveis e ubíquas, devem ser capazes de executar ações adaptadas a situação atual do usuário, a medida que esse altera sua condição contextual.

Entretanto, a sensibilidade ao contexto não trata apenas da adaptabilidade das aplicações, de uma maneira mais geral, um sistema é sensível ao contexto se ele utiliza o contexto para fornecer informações e/ou serviços relevantes para o usuário, onde a relevância depende da tarefa do usuário (DEY, 2000). Então, entendemos a sensibilidade ao contexto como a capacidade de reconhecer o ambiente de execução atual e tomar decisões, relevantes para o usuário, através das informações que caracterizam o ambiente. As definições mais específicas de sensibilidade ao contexto exigem que o comportamento de uma aplicação seja modificado para que essa possa ser considerada sensível ao contexto. Porém, quando tentamos aplicar essas definições para aplicações sensíveis ao contexto, descobrimos que eles não se encaixam. Por exemplo, um aplicativo que simplesmente exhibe o contexto atual do ambiente do usuário não está modificando o seu comportamento, mas é sensível ao contexto. Se usarmos as definições menos gerais, esse tipo de aplicação não seria classificada como sensível ao contexto. E por essa razão, optamos por uma definição mais abrangente de sensibilidade ao contexto em detrimento a definição restrita à “adaptação ao contexto”.

A característica de reconhecer o ambiente a medida que esse se altera e tomar decisões a fim de prover uma condição mais adequada ao propósito do usuário final vai de encontro as

ideias de personificação da computação, isto é, tornar a computação cada vez mais próxima e natural para os usuários finais. Porém, o processo de reconhecimento do ambiente depende diretamente da qualidade da informação provida às aplicações. Por esse motivo, a qualidade dos dados de contexto é uma questão chave, dado que sem a mesma as operações sob esses dados, não refletirão o comportamento desejado. Sendo assim, na próxima seção apresentamos os conceitos que envolvem a qualidade de contexto (QoC), bem como a sua importância para uma habilitação efetiva da sensibilidade ao contexto.

### 3.2 QUALIDADE DE CONTEXTO

Similar a sensibilidade ao contexto, não há uma definição bem aceita do que efetivamente seja qualidade de contexto. (BUCHHOLZ; KÜPPER; SCHIFFERS, 2003) define qualidade de contexto (QoC) como: *“qualquer informação que descreve a qualidade da informação a ser utilizada como informação de contexto. Assim, QoC refere-se à qualidade da informação e não ao processo e nem ao componente de hardware que, possivelmente, fornece a informação”*. Já a definição de (KRAUSE; HOCHSTATTER, 2005) diz que qualidade de contexto é qualquer informação inerente que descreve informação de contexto e pode ser usada para determinar o valor da informação para uma aplicação específica. Isso inclui informações sobre o processo de provisionamento que a informação foi submetida (“histórico”, “idade”), mas não tratam de estimativas sobre os passos de provisionamentos futuros.

Nesse sentido, a qualidade de contexto (QoC) é usualmente concebida como um conjunto de parâmetros que expressam os requisitos de qualidade e propriedades para dados de contexto (por exemplo, a precisão, atualidade, confiabilidade). Várias pesquisas estão sendo realizadas para identificar e avaliar as informações relevantes para garantir a qualidade de contexto (KLEIN; DAVID, 2010) (MANZOOR; TRUONG; DUSTDAR, 2008) (NEISSE; WEGDAM; SINDEREN, 2008) (BRGULJA et al., 2009) (KIM; LEE, 2006b) (MIRON et al., 2010). Entretanto, essa noção de qualidade de contexto não engloba todos os componentes que interferem para a qualidade de uma informação de contexto. A qualidade de contexto refere-se não só à qualidade dos dados de contexto, mas também à qualidade de distribuição desses dados (BUCHHOLZ; KÜPPER; SCHIFFERS, 2003). Sem uma distribuição eficiente de dados, não é viável a implantação de serviços verdadeiramente sensíveis ao contexto. Assim, o provedor de contexto deve ter algumas garantias sobre a distribuição de dados de contexto, antecipando a ocorrência de algumas situações adversas (por exemplo, a desconexão, falta de sinal, e *roaming*).

Para que seja entendida a necessidade de se garantir a qualidade de contexto, precisamos discretizar os fatores que interferem na qualidade de uma informação de contexto. (KRAUSE; HOCHSTATTER, 2005) lista algumas fontes de erros em dados de contexto:

- Uma informação de contexto necessária (ou fonte da informação) pode não estar disponível;
- A informação de contexto pode estar “vencida” e não ser mais aplicável para a situação atual;
- Restrições físicas podem limitar a precisão dos sensores;
- Informações de perfil padrão (*default*) podem não ser aplicáveis à situação atual;
- Regras de raciocínio, muitas vezes baseadas em probabilidades, não se aplicam em determinadas situações e que poderiam, portanto, obter informações de contexto equivocadas. (Nem todo ruído detectado significa que uma pessoa está na sala).
- Informações de fontes mal-intencionadas poderiam simular um contexto que não é real.

Existem outras fontes de erros para dados de contexto, (DEY et al., 2002) levanta que tanto informações de contexto sensoriadas quanto interpretadas são frequentemente ambíguas, então um dos principais desafios na criação de serviços sensíveis ao contexto realísticos, é saber como gerenciar contextos ambíguos. (HENRICKSEN; INDULSKA, 2004) sugere que a imperfeição é uma característica das informações de contexto: A informação pode estar incorreta se não refletir a verdadeira situação real, pode estar inconsistente se contiver informações contraditórias, pode estar incompleta se alguns atributos de um determinado contexto forem desconhecidos. Essa imperfeição ocorre sob o ponto de vista da distribuição dos dados, por questões como, grandes *delays* entre a produção e o uso de informações de contexto, e desconexões entre o produtor e o consumidor de contexto.

Os modelos de contexto então devem, também, especificar um conjunto de características que qualifiquem as informações de contexto, para que assim ocorra um uso adequado das informações de contexto. A subseção a seguir apresenta alguns fatores que podem ser utilizados para auxiliar na garantia de qualidade de informação de contexto.

### 3.2.1 Parâmetros de qualidade de contexto

Apesar das indefinições sobre o que permeia a qualidade de contexto (QoC), indicadores (ou parâmetros) de QoC são sugeridos na literatura como ferramentas essenciais para se garantir qualidade de contexto. Indicadores de QoC tem como principal objetivo delimitar o que é de interesse para as aplicações que os utilizam. Além de, controlar e gerenciar as possíveis imprecisões nas informações de contexto. Por esse motivo, a tentativa de encontrar um conjunto finito definitivo de parâmetros de QoC que qualifiquem informações de contexto em sua totalidade, é um esforço desnecessário. De fato, para que se determine um conjunto de indicadores de QoC

é necessário entender as especificidades da aplicação em questão, bem como a infraestrutura computacional onde o serviço será projetado.

Entretanto, alguns parâmetros de qualidade podem ser utilizados em uma grande gama de aplicações, devido principalmente, a natureza da captação dos dados de contexto e da similaridade encontrada em serviços computacionais sensíveis ao contexto, mesmo que esses possuam propósitos distintos. A definição desses parâmetros tem sido o foco de algumas pesquisas como (BUCHHOLZ; KÜPPER; SCHIFFERS, 2003) (MANZOR; TRUONG; DUSTDAR, 2008) (SHEIKH; WEGDAM; SINDEREN, 2007) (NEISSE; WEGDAM; SINDEREN, 2008) (KLEIN; DAVID, 2010) (CHEN; KOTZ et al., 2000) (DEY, 2000), apresentamos abaixo os parâmetros que foram mais citados na literatura:

- **Precisão:** descreve a proximidade da informação com a realidade;
- **Prazo de Validade:** lida com a idade do dado;
- **Atualidade:** o tempo transcorrido entre a geração da informação de contexto e a entrega da informação ao consumidor;
- **Probabilidade de Corretude:** denota a probabilidade daquela informação estar correta;
- **Resolução:** apresenta a granularidade da informação;
- **Confiança:** descreve a confiança naquela informação;
- **Completude:** indica o grau de informações que estão disponíveis, em relação as informações desejadas;
- **Significância:** expressa diferentes prioridades, específicas para cada consumidor de dados de contexto;
- **Segurança de Acesso:** representa a probabilidade da informação ser entregue com segurança para os consumidores de contexto.

Não é difícil imaginar situações em que esses parâmetros possam ser aplicados, a exemplo do parâmetro **precisão**, que pode ser utilizado para assegurar a qualidade da informação provida por um GPS, onde dados que possuem uma precisão maior que um determinado valor limítrofe são eliminados. Ou ainda, a utilização do parâmetro **atualidade** para eliminar dados que possuam um determinado *delay*, e sejam entrada para uma aplicação em tempo real. Parâmetros mais específicos, também foram sugeridos na literatura e recebem destaque nesse trabalho de pesquisa. É importante ressaltar que, apesar dos parâmetros serem específicos para algumas aplicações, a ideia que envolve a concepção desses parâmetros pode ser aplicadas no projeto de diferentes serviços sensíveis ao contexto.



Em (SHEIKH; WEGDAM; SINDEREN, 2007), outros dois parâmetros são discutidos: **resolução temporal** e **resolução espacial**. A resolução temporal é definida como o período de tempo em que uma determinada informação de contexto é aplicável. Em outras palavras, é o intervalo de tempo máximo para se enviar uma determinada informação. Já a resolução espacial seria a precisão com que uma área física é expressada. A ideia da resolução espacial é delimitar onde (localização geográfica) as informações de contexto podem ser captadas e enviadas. Esse dois parâmetros podem ser utilizados para auxiliar na segurança do provimento de informações de contexto, limitando o espaço e tempo em que os dados devem ser comunicados. Este é um dos primeiros trabalhos que utiliza parâmetros de QoC na perspectiva de segurança do usuário.

A comparação entre informações providas por diferentes dispositivos também podem ser utilizadas para avaliar a qualidade de contexto. No trabalho de (KLEIN; DAVID, 2010), é sugerido um parâmetro denominado: **localidade temporal**. Esse indicador expressa a diferença de tempo entre amostras de diferentes sensores, que são agregados para formar um contexto. A grande contribuição desse trabalho consiste na ideia de se trabalhar com amostras de diferentes sensores para garantir qualidade de contexto. Em aplicações sensíveis ao contexto, projetadas em ambientes ubíquos móveis heterogêneos, a comparação entre diferentes amostras tem um impacto significativo na garantia de qualidade, dado que as informações fornecidas pelos sensores podem diferir da realidade devido a imprecisão intrínseca encontrada em muitos destes dispositivos.

Através de indicadores de qualidade é possível distinguir quais informações são relevantes para uma determinada aplicação. Porém para um uso efetivo desses indicadores é necessário quantificá-los, ou seja, atribuir um valor (geralmente numérico) ao parâmetro de QoC. Na tabela 2 apresentamos um resumo das estratégias utilizadas para quantizar parâmetros de QoC (NAZARIO; DANTAS; TODESCO, 2012). Diversas estratégias são sugeridas para atribuir valores aos parâmetros de qualidade de contexto. Em sua maioria, as estratégias apresentadas tentam atribuir valores numéricos aos indicadores de QoC para que possam ser conduzidas operações matemáticas, com a finalidade de qualificar uma informação de contexto.

Entretanto, independentemente da estratégia utilizada, é preciso discretizar os elementos que auxiliam a descoberta do valor de um determinado indicador de QoC. (MANZOOR; TRUONG; DUSTDAR, 2008) e (MIRON et al., 2010) sugerem alguns fatores para realizar a quantificação dos parâmetros de QoC, a tabela 3 apresenta tais fatores.

A discretização dos fatores que influenciam nos valores dos parâmetros a serem utilizados em uma determinada aplicação é de suma importância no projeto de serviços sensíveis ao contexto. Mas, por outro lado, existem trabalhos que sugerem que o valor atribuído a certos parâmetros não deve ser numérico, um exemplo desta abordagem está em (NEISSE; WEGDAM; SINDEREN, 2008) que tem o enfoque no parâmetro **confiança**. Nesse trabalho, o conceito de confiança foi criado a partir das definições encontradas nas áreas sociais.

<b>Estratégia de Quantificação</b>	<b>Autores</b>
Estimativa estatística	(KIM; LEE, 2006b)
Álgebra min-plus, max-plus e min-max-plus	(WIDYA; BEIJNUM; SALDEN, 2006)
Algoritmos genéticos	(ZIMMER et al., 2006a) e (ZIMMER, 2006b)
Lógica <i>fuzzy</i>	(GIAFFREDA; BARRIA, 2007) e (MANZOOR; TRUONG; DUSTDAR, 2011)
Descrição de alternativas para quantificação de parâmetros de forma textual	(SHEIKH; WEGDAM; SINDEREN, 2007)
QoC (fontes e parâmetros): os valores de fontes são usados para determinar os parâmetros de QoC	(MANZOOR; TRUONG; DUSTDAR, 2008) e (MANZOOR; TRUONG; DUSTDAR, 2011)
Lógica subjetiva (crença, descrença e incerteza)	(NEISSE; WEGDAM; SINDEREN, 2008)
Indicadores quantificados através da composição: $M_{QoC} = \langle CI, QoCI, Alg \rangle$	(MARTIN et al., 2008)
QoC (indicadores e parâmetros): os parâmetros são usados para medição dos indicadores de QoC	(MIRON et al., 2010)
Definições matemáticas (Projeto Nexus)	(GROSSMANN et al., 2009)
Método baseado na Teoria de Probabilidade Bayesiana	(BRGULJA et al., 2009)
Diferença de tempo entre sensores	(KLEIN; DAVID, 2010)
Fórmulas matemáticas para cálculo de parâmetros	(MANZOOR; TRUONG; DUSTDAR, 2008) e (VANROMPAY; MEHLHASE; BERBERS, 2010)
Modelos matemáticos baseados em: distribuições de probabilidade, regiões de tolerância, medidas estatísticas, medidas probabilísticas	(DÜRR et al., 2010)
Cálculo de pesos e prioridades para os parâmetros de QoC	(ABID; CHABRIDON, 2011)
São apresentadas fórmulas para o cálculo dos parâmetros, sendo a média ponderada dos parâmetros calculados o valor de QoC	(YASAR et al., 2011)
Probabilidade de ocorrência e cálculo de confiança com base na fusão de observação de sensores múltiplos	(HOSSAIN et al., 2012)
Fórmulas para o cálculo de diversos parâmetros baseado na literatura	(ZHENG; WANG; KERONG, 2012)

Tabela 2: Estratégias para quantificação de parâmetros de QoC (NAZARIO; DANTAS; TODESCO, 2012)

(NEISSE; WEGDAM; SINDEREN, 2008) cita que a confiança é uma relação estabelecida entre duas entidades, e que seu valor quando representado numericamente não expressa essa relação de forma completa. Assim, através da utilização de lógica subjetiva o valor da relação é mapeado para o conjunto [*muito confiável, confiável, inconfiável, muito inconfiável*].

<b>Parâmetros de QoC</b>	<b>Fontes de QoC (MANZOOR; TRUONG; DUSTDAR, 2008)</b>	<b>Fontes de QoC (MIRON et al., 2010)</b>
<i>Validade do dado</i>	HoraDaMedição, HoraAtual	HoraDaCaptura, HoraAtual, TempoDeVida
<i>Confiança</i>	LocalizaçãoDaFonte, LocalizaçãoDaEntidadeInformante, PrecisãoDoSensor	—
<i>Completude</i>	Relação entre o número de atributos preenchidos e o número de atributos existentes	NúmeroDeRespostas, NúmeroDeRequisições
<i>Significância</i>	ValorCrítico	—
<i>Sensitividade</i>	—	NíveisDeDivulgação, NívelDeDivulgaçãoAtual
<i>Segurança para acesso</i>	—	NívelDeSegurançaAtual, NíveisTotaisDeSegurança
<i>Precisão</i>	—	PrecisãoDesejada, PrecisãoAtual

Tabela 3: Informações utilizadas para medir parâmetros de QoC (NAZARIO; DANTAS; TODESCO, 2012)

Parâmetros de QoC devem ser utilizados em serviços sensíveis ao contexto para que se tenha algumas garantias quanto a qualidade de contexto. Porém, um uso efetivo desses parâmetros só ocorre se seus respectivos valores, numéricos ou não, forem estabelecidos. Ademais, os requisitos das aplicações devem ser consultados, pois esses indicam diretamente parte dos parâmetros que devem ser utilizados, ou até mesmo, criados. E apesar de não refletirem todos os aspectos de qualidade desejados pelas aplicações, os parâmetros de qualidade fornecem uma base do que deve ser levado em consideração na concepção de serviços sensíveis ao contexto.

### 3.3 GERENCIAMENTO DE DADOS DE CONTEXTO

O gerenciamento de dados de contexto se refere basicamente a uma questão fundamental: “como representar e processar dados de contexto?”. Primeiramente, os dados de contexto devem ser representados através de alguma técnica de modelagem de dados. A seção 3.2.1 apresenta diversas abordagens apresentadas na literatura. Em segundo lugar, dados de contexto devem ser processados de acordo com as necessidades dos serviços. A seção 3.2.2 são apresentadas as principais operações que podem ser realizadas em dados de contexto para que as necessidades desejadas por serviços sensíveis ao contexto sejam supridas. Na figura 4 apresentamos a classificação proposta por (BELLAVISTA et al., 2013), sob as diferentes abordagens utilizadas para representar e processar dados de contexto.

Modelos de propósito geral fornecem suporte para especificação e representação de aplicações nos mais diversos domínios, eles focam na representação do conhecimento. Por outro lado, mo-

delos de domínio específico representam apenas dados que pertencem a um domínio específico e vertical, e não permitem a especificação de dados genéricos. Porém, devido ao escopo reduzido, podem ser introduzidas operações mais complexas sob os dados. Modelos híbridos se baseiam em dois ou mais modelos, tanto geral como específico, ao mesmo tempo. Finalmente, vários sistemas baseados em informações de contexto não indicam aspectos de representação de dados, por focarem em outros aspectos, tais sistemas foram denominados “sem modelo”.

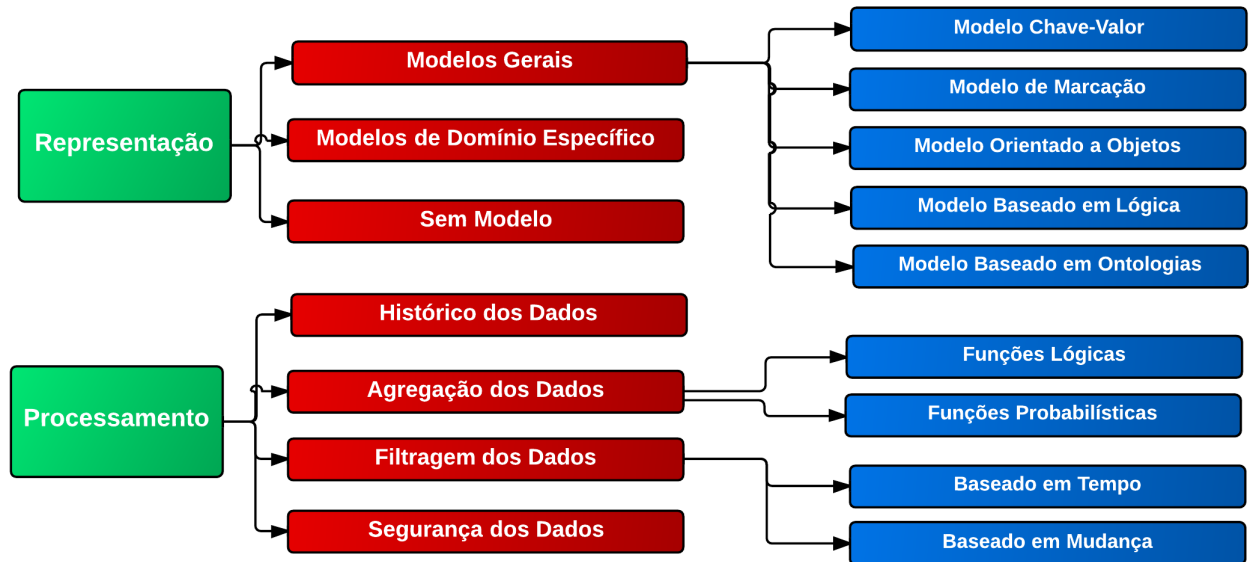


Figura 4: Classificação das abordagens de gerenciamento de dados de contexto (BELLAVISTA et al., 2013)

Na subseção seguinte, são apresentados os modelos de propósitos geral, dado que os *middlewares* de contexto são costumeiramente baseados nesse tipo de abordagem. Além disso, são apresentados alguns exemplos que utilizam modelos de domínio específico.

### 3.3.1 Representação de contexto

Diversas formas de representação de dados de contexto foram sugeridas na literatura, sendo que as principais diferenças concentram-se nos aspectos de expressividade, custo de memorização, e *overhead* de processamento. Como explicitado em (STRANG; LINNHOFF-POPIEN, 2004), a computação ubíqua faz algumas demandas em relação a qualquer abordagem para modelagem de dados de contexto, em termos de:

- *composição distribuída (cd)*: como apresentado na figura 2, os sistemas ubíquos são uma derivação direta de sistemas distribuídos, cujo os quais não demandam uma instância centralizadora que seja responsável pela criação, implementação e manutenção de dados e serviços, em particular especificações de contexto. Ao invés disso, a composição e administração de modelos e dados de contexto variam dinamicamente em termos de tempo, topologias de rede e fontes.

- *validação parcial (vp)*: é altamente desejável ser capaz de validar, mesmo que parcialmente, o conhecimento contextual nas diversas camadas que compõem o sistema. Isso é de suma importância, devido a alta complexidade encontrada nas interações de contexto que fazem com que as abordagens de modelagem de dados sejam suscetíveis a erros.
- *qualidade da informação (qi)*: a qualidade de uma informação fornecida por um sensor se altera ao passar do tempo, bem como a riqueza das informações providas por diferentes sensores para caracterizar uma entidade em um ambiente ubíquo podem diferir. Assim, um modelo de contexto é considerado apropriado para ser utilizado em ambientes ubíquos caso de suporte para criação e avaliação de indicadores de qualidade e riqueza.
- *completude e ambiguidade (ca)*: o conjunto de informações disponíveis para caracterizar entidades relevantes em ambientes ubíquos, usualmente, é incompleta ou ambígua. Isso deve ser contemplado no modelo de contexto através, por exemplo, da composição de dados incompletos.
- *formalismo (for)*: É um desafio constante descrever fatos e relações de contexto de maneira precisa e rastreável. Como exemplo, para realizar a tarefa “imprima esse documento na impressora que está perto de mim”, é necessário ter uma definição precisa dos termos usados na tarefa, por exemplo, o que “perto” significa para “mim”. Além disso, é altamente desejável que cada parte participante de uma interação compartilhe a mesma interpretação dos dados trocados (o chamado entendimento compartilhado).
- *aplicabilidade em ambientes já existentes (app)*: Da perspectiva de implementação, é importante que o modelo de contexto possa ser aplicado nas infraestruturas ubíquas já existentes, a exemplo de *frameworks* de serviço e *WebServices*.

Modelos de propósito geral oferecem diferentes níveis de formalismo e expressividade. A expressividade é estritamente relacionada às possíveis operações sob os dados, modelos mais complexos tendem a padronizar e suprir operações adicionais, como operadores de agregação que derivam novos dados de contexto e operadores de qualidade usados para especificar e gerenciar fatores de QoC. Entretanto, ainda não existe um acordo geral sobre a padronização das operações sob dados de contexto, dificultando assim uma adoção generalizada de ferramentas automáticas para gerenciamento de dados de contexto reutilizáveis. Sistemas sensíveis ao contexto devem adotar algum dos principais modelos aceitos e seguidos, em ordem de complexidade: modelos chave-valor, modelos de esquema de marcação, modelos orientados a objetos, modelos baseados em lógica, e modelos baseados em ontologia (BETTINI et al., 2010) (STRANG; LINNHOFF-POPIEN, 2004). Detalhes sobre os modelos são apresentados a seguir.

Modelos chave-valor possuem uma estrutura simples para modelagem de dados de contexto, pela a exploração de dois fatores: uma chave (nome do atributo) e o seu valor. A simpli-

cidade pode ser colocada como a principal razão para a popularidade dessa abordagem. Porém, esses modelos possuem restrições quanto as diferentes formas de estruturação de dados, além de não prover mecanismos para validade dos mesmos. *Context Toolkit*, um dos mais importantes trabalhos na área de sensibilidade ao contexto, adota essa abordagem para representar tanto dados de contexto quanto os metadados associados com as fontes de informação de contexto (DEY; ABOWD, 2000). *Pervasive Autonomic Context-aware Enviroments* (PACE) se baseia em pares de chave valor para representar dados de contexto que são usados para determinar quais são as preferências dos usuários em relação as ações executadas no contexto pervasivo atual (HENRICKSEN et al., 2005). Protocolo de roteamento baseado em histórico para redes oportunistas (HiBOP) e Roteamento Adaptativo Consciente de Contexto (CAR) são duas soluções bem significantes para redes tolerantes a *delays* (DTN), e utilizam o contexto computacional, de tempo e de usuário para avaliar e selecionar o melhor dado a ser enviado. Para conceber a infraestrutura de roteamento, cada dispositivo móvel distribui os dados de contexto solicitados, descritos como pares de chave-valor, para seus vizinhos (BOLDRINI; CONTI; PASSARELLA, 2008) (MUSOLESI; MASCOLO, 2009).

Modelos de esquema de marcação utilizam representações baseada em XML para modelar estruturas de dados hierárquicas com *tags* de marcação, atributos, e conteúdos. Essa abordagem supera algumas limitações de modelos chave-valor. Por exemplo, eles suportam a possibilidade de: i) validar dados de contexto em termos de esquemas XML, e ii) estruturar dados aninhados através das estruturas XML. Ambiente para gerenciamento de recursos sensíveis ao contexto (CARMEN) explora perfis baseados em XML para descrever informações sobre o contexto computacional e sobre o contexto de usuário (BELLAVISTA et al., 2003). *Context CASTing* (C-CAST) é explicitamente focado nos aspectos de provisionamento de contexto, e define uma linguagem baseada em XML (ContextML) para distribuir os dados de contexto para todo o sistema (KNAPPEYER et al., 2009). *Cooltown* utiliza **Web presences** (identificado por uma URL) para oferecer acesso as informações de contexto desejadas: **Web presences** são interfaces web ligadas por *web hyperlinks* que podem ser navegadas para obter mais informações (DEBATY; GODDI; VORBAU, 2005). COPAL (LI; SEHIC; DUSTDAR, 2010) foca na entrega e processamento de dados de contexto, e representa os dados através de documentos XML. CONSINE (JUSZCZYK et al., 2009) constrói um compartilhamento modular de contexto, cujo o qual os dados de contexto são representados por XML e podem ser consultados através de consultas XPath. Por fim, MANIP não estabelece um modelo de dados de contexto, mas sugere alguns candidatos, como o modelo de informação comum (CIM) e o *Framework* para Descrição de Recursos (RDF), que pertencem as abordagens de (MACEDO et al., 2009), (POWERS, 2009) e (SWEITZER et al., 1999).

Modelos orientados a objetos tomam vantagem dos benefícios fornecidos por uma abordagem orientada a objetos, tipicamente encapsulamento e reusabilidade: cada classe define

um novo tipo de contexto com funcionalidades de acesso associadas. Checagem de tipo e a validação dos dados podem ser asseguradas em tempo de execução e compilação, enquanto os elementos de QoC podem ser mapeados como outros objetos. Além disso, esses modelos facilitam a interação entre serviços e dados de contexto, utilizando as mesmas abstrações providas pelas linguagens de programação orientadas a objetos, facilitando a implementação do código para gerenciar os dados de contexto. Em COSMOS (CONAN; ROUVOY; SEINTURIER, 2007), cada dado de contexto é reificado como um objeto contendo diversos mecanismos para assegurar notificações de mudança baseadas em *push* e *pull*. O projeto *Hydrogen* representa cada tipo de dado de contexto como uma *subclass* de *ContextObject* (HOFER et al., 2003). *middleware* reconfigurável sensível ao contexto (RCSM) explora diretamente uma linguagem de definição de interface (IDL) para sustentar sua abordagem (YAU et al., 2004): Ao utilizar a linguagem o desenvolvedor pode especificar situações de contexto relevantes para a aplicação, as ações a serem tomadas, e o tempo de execução dessas ações. Similarmente, *MobiSoC* e *MoCA* (GUPTA et al., 2009) e (SACRAMENTO et al., 2004) trocam objetos tradicionais definidos pelos desenvolvedores para representar dados de contexto, pertencentes principalmente a dimensão de contexto do usuário.

Modelos baseados em lógica tomam vantagem da alta capacidade de expressividade intrínseca ao formalismo lógico, onde o contexto contém fatos, expressões, e regras, enquanto o conhecimento pode ser extraído por inferência. Tradicionalmente, esses modelos focam em mecanismos de inferência, fornecendo também formalismos apropriados para especificar as regras de inferência. Porém, usualmente os modelos baseados em lógica não oferecem funcionalidades simples para lidar com a validade do dado. A validação, nesse caso, pode ser assegurada mas as regras associadas não são fáceis de especificar e dependem do tipo de lógica adotada. *Mobile Gaia* e *Gaia* representam contexto utilizando predicados de primeira-classe (CHETAN et al., 2005) (RANGANATHAN; CAMPBELL, 2003). Nos dois sistemas, o contexto é representado através de tuplas quaternárias como: *Contexto*(*<TipoDoContexto>*, *<Sujeito>*, *<Relacionador>*, *<Objeto>*), onde *<TipoDoContexto>* é o tipo de contexto que o predicado descreve, *<Sujeito>* é a pessoa, lugar, ou objeto físico de interesse para aquele contexto, *<Relacionador>* relaciona o *<Sujeito>* com o *<Objeto>* utilizando comparadores lógicos, um verbo, ou uma preposição. *CORTEX* e *CASS* utilizam uma abordagem similar para modelagem dos dados (SØRENSEN et al., 2004) (FAHY; CLARKE, 2004) *EgoSpaces* adota um modelo de tuplas, porém não pré-define a estrutura das mesmas (JULIEN; ROMAN, 2006).

Modelos baseados em ontologias representam os dados de contexto através de ontologias e se beneficiam da capacidade de expressar relações complexas. A validade dos dados é usualmente expressada pela imposição de restrições ontológicas. Focando-se nas relações entre entidades, ontologias são muito adequadas para o mapeamento do conhecimento dentro de uma estrutura de dados facilmente utilizável e automaticamente gerenciável. Adicionalmente, a am-

plata adoção de ontologias permite a reutilização dos modelos já definidos em trabalhos anteriores, e a criação de vocabulários de domínios comuns e que podem ser compartilhados. SOCAM compõe uma ontologia genérica com algumas de domínio específico (GU; PUNG; ZHANG, 2005). Quando necessário, a ontologia específica é unificada com a genérica. SOCAM (CHEN; FININ; JOSHI, 2003) classifica os dados como diretos - direto dos sensores ou pré-definidos por usuários - e indiretos - derivados por inferência. *Framework* para gerenciamento de contexto (CMF) pode agregar diferentes ontologias de domínio específico, além de definir diferentes domínios de administração (KRANENBURG et al., 2006). Mesmo que as abordagens por lógica de primeira ordem e ontologias pareçam promissoras, ambientes móveis geralmente evitam essas abordagens, uma vez que os recursos computacionais necessários (memória e CPU) podem não ser aceitáveis devido a limitação dos recursos disponíveis nos dispositivos móveis.

Finalmente, apesar de um único modelo ser capaz de facilmente representar operações para gerenciamento e dados, alguns sistemas tendem a adotar modelos híbridos para captar o melhor dos modelos usados. Solar e SALES representam dados utilizando a abordagem chave-valor juntamente com a abordagem orientada a objetos (CHEN; LI; KOTZ, 2008) (CORRADI; FANELLI; FOSCHINI, 2010). Os pares chave-valor são utilizados para reduzir o *overhead* no gerenciamento dos dados, enquanto a abordagem orientada a objetos facilita o projeto e a implementação dos sistemas, podendo esses serem extensíveis.

Os modelos de domínio específico são menos flexíveis uma vez que são centrados num domínio de uma aplicação em particular, e, ao mesmo tempo, devido à flexibilidade restrita, operadores de agregação mais complexos são usualmente usados por sistemas que utilizam essa abordagem. Por exemplo, os modelos de dados espaciais são amplamente adotados por sistemas de localização para representar objetos do mundo real, a sua localização e realizar consultas sobre contenção, intersecção, e assim por diante de forma rápida e eficiente. Neste caso, a validade dos dados é assegurada mais facilmente, e ferramentas automáticas geralmente estão disponíveis para especificar regras de validação. Entre as soluções encontradas, *MiddleWhere* é um sistema de distribuição de dados de contexto estritamente relacionado com cenários conscientes de localização, por isso, são focados especificamente na dimensão de contexto físico (RANGANATHAN et al., 2004). *MiddleWhere* usa um modelo de dados espaciais assumindo que os objetos do mundo real podem ser apenas pontos, linhas e polígonos, e que os dados de localização geográfica formam uma hierarquia. Com base nessa representação espacial do mundo, *MiddleWhere* responde a consultas baseados em localização geográfica.

Grande parte das soluções encontradas na literatura não possuem uma modelagem de dados, tipicamente porque o foco desses trabalhos é direcionado para aspectos técnicos. A exemplo de *Pervaho* que utiliza eventos para distribuir dados de contexto ao invés de especificar um formato interno para os dados (EUGSTER; GARBINATO; HOLZER, 2008). *HiCon* lida com a agregação de dados de contexto aceitando qualquer tipo de representação de dados (CHO



et al., 2008). Outras soluções focam no design de protocolos para distribuição dos dados, como na solução *Habit* que usa o contexto físico e de usuário para criar rotas de distribuição de dados (MASHHADI; MOKHTAR; CAPRA, 2009).

Para concluir, é possível inferir que a maioria dos sistemas tendem a adotar modelos mais simples, como chave-valor e modelos de esquema de marcação. No entanto, o modelo de dados adotado que deve ser adotada depende, principalmente, dos cenários suportados e das operações de agregação que serão executadas. Ao mesmo tempo, embora quase todos os modelos acima oferecem abstrações suficientes para representar restrições QoC, nenhum deles possuem ferramentas maduras para declarar e para aplicar a qualidade de contexto de maneira efetiva. O enorme escopo de representação dos dados de contexto, os diferentes aspectos semânticos associados aos dados representados, e a ausência de *frameworks* genéricos de QoC, podem ser responsabilizados por essa falta de ferramentas genéricas que tratam da qualidade de contexto. Consequentemente, devido aos problemas acima citados, os sistemas reais tendem a apresentar soluções *ad-hoc* e explorar metadados adicionais para o tratamento da qualidade de contexto.

### 3.3.2 Processamento de dados de contexto

Baseando-se na taxonomia apresentada por (BELLAVISTA et al., 2013), o processamento de dados de contexto envolvem basicamente quatro principais aspectos: histórico de dados de contexto, agregação de dados, filtragem de informações e segurança no acesso. Nessa seção, vamos apresentar com mais detalhes os componentes de processamento e como alguns trabalhos utilizam esses componentes. Antecipadamente, é importante salientar que dependendo do operador de processamento utilizado, essas técnicas são propensas a apresentar sobrecargas muito diferentes nos sistemas que as utilizam.

O histórico dos dados de contexto trata da manutenção de todos os dados relevantes do passado e da possibilidade de recuperação de um histórico de contexto particular. O histórico de dados de contexto impõe exigências sobre os recursos de memória, consoante ao tamanho dos dados e à taxa de produção. Nesse sentido, pode ser difícil manter toda o histórico, especialmente para serviços implementados em ambientes móveis. No entanto, apesar dos recursos necessários, o histórico dos dados de contexto é muito útil, como demonstrado pela novas soluções emergentes que sugerem esse armazenamento. SOCAM mantém o histórico de dados que possuem informações geográficas (GU; PUNG; ZHANG, 2005). Em CORTEX, cada fonte de contexto faz a previsão futura baseado no estado atual e nos dados de contexto armazenados (SØRENSEN et al., 2004). *Cooltown* mantém o histórico de eventos referentes a entidade física (produtora das informações de contexto) (DEBATY; GODDI; VORBAU, 2005). Em C-CAST, o processador de contexto pode implementar um histórico para recorrer a fatos passados (KNAPPMAYER et al., 2009).

As operações de agregação em dados de contexto tratam, usualmente, de funções que realizam a fusão e sobreposição em diferentes dados de contexto. Operações mais específicas dependem diretamente do modelo de contexto adotado, e dado a imprecisão encontrada nos dados de contexto, tais operações devem ser implementadas sob a ótica da qualidade de contexto. As técnicas de agregação encontradas podem ser classificadas como lógicas e probabilísticas, dependendo se o sistema considera que um dado pode estar simplesmente correto, ou que está correto em uma determinada probabilidade (tipicamente menos que 1). Além disso, esquemas híbridos que combinam esses dois tipos de abordagem também podem ser concebidos. As técnicas de agregação podem ser muito exigir muitos recursos computacionais para serem executadas, porém são fundamentais para que a sensibilidade ao contexto ocorra de forma completa, dado que: i) geralmente, o contexto não pode ser definido explicitamente, devido à enorme quantidade de possíveis situações que podem ocorrer e ii) as mudanças de contexto necessitam de atualizações contínuas que devem ser realizadas automaticamente pelo sistema. Acima de tudo, a inteligência artificial provê técnicas, bem como a representação de base lógica padrão e motores de inferência, que podem simplificar a utilização de técnicas de agregação. Consequentemente, a maioria dos sistemas que requerem a agregação de dados dinâmico podem adotar qualquer um dos modelos de lógica, ou ainda, modelos baseados em ontologias que são mais simples de gerenciar e integrar-se com os motores de inferência.

Considerando as funções lógicas, em C-CAST, diferentes provedores de contexto, podem ser configurados para trocar dados de contexto e realizar agregação de dados de maneira distribuída (KNAPMEYER et al., 2009). Context Toolkit possui a habilidade de agregar dados de contexto captados diretamente dos sensores através da utilização dos chamados Context Toolkit widgets (DEY; ABOWD, 2000)s. COPAL possibilita a especificação de operações complexas para processamento de contexto, em particular, é disponibilizado um operador “Agregação” que tem como principal objetivo agregar dados de contexto para um nível mais alto (LI; SEHIC; DUSTDAR, 2010). Em COSINE, serviços de agregação são explicitamente introduzidos para lidar com a agregação de dados. Além disso, se existe uma demanda de contexto que exija a correlação de múltiplos tipos de dados de contexto, o serviço de agregação é capaz de determinar quais os contextos que devem ser agregados, e entregar os dados de forma transparente (JUSZCZYK et al., 2009). O *middleware* COSMOS separa o processo de agregação em três camadas principais: coleta, processamento, e adaptação. Além disso, oferece processadores de contexto que são utilizados para realizar a agregação (CONAN; ROUVOY; SEINTURIER, 2007).

Considerando as funções probabilísticas, *MiddleWhere* encorpora uma gama de técnicas de agregação de dados geográficos (RANGANATHAN et al., 2004). Dados fornecidos por diferentes sensores são armazenados em uma base de dados espacial, e são agregados para fornecer a informação de localização sob uma determinada probabilidade. Ademais, o serviço de

localização elimina conflitos, e responde a requisições sobre regiões espaciais e objetos físicos. MobiSoC foca no aprendizado do estado social do usuário, e introduz diferentes soluções para a descoberta de relações entre pessoas, grupos sociais, e relações de grupo-lugar: os autores apresentam o algoritmo de Identificação de Grupo-Lugar (GPI) para agregar as informações do contexto do usuário e inferir padrões geo-sociais com uma certa probabilidade (GUPTA et al., 2007) (GUPTA et al., 2009).

A filtragem de dados de contexto busca melhorar a escalabilidade de sistema pelo controle e redução da quantidade de dados de contexto transmitidos. Essas técnicas são fundamentais, pois: i) algumas informações de contexto mudam constantemente, e as fontes associadas a produção desses dados podem produzir informações sob uma taxa muito alta, e ii) o fornecimento de dados de contexto deve ser realizado em acordo com as definições de QoC. Os trabalhos encontrados realizam a filtragem dos dados através da abordagem baseado em tempo, isto é, os dados são enviados sob intervalos determinados de tempo, e abordagens baseadas em mudança, isto é, o envio dos dados não acontece enquanto o dado de contexto for igual ou similar ao dado enviado previamente. Qualquer combinação dessas técnicas também pode ser utilizada. Por exemplo, um serviço pode pedir para receber atualizações de localização somente se a nova localização é diferente da anterior, e se pelo menos um intervalo de 10 segundos foi decorrido.

Finalmente, a segurança para acesso a dados de contexto engloba todos os mecanismos para assegurar privacidade, integridade e disponibilidade dos dados. Cenários reais exigem essa segurança pois dados de contexto podem possuir informações sigilosas. Por exemplo, informações sobre a temperatura do ambiente trocadas em forma de texto podem não parecer para o usuário uma violação de privacidade, outras informações como a localização do usuário podem exigir mecanismos para garantia de privacidade. No entanto, pode ser observado que as questões de segurança já foram enfrentadas e resolvidos na literatura, e soluções eficientes para resolver os problemas de segurança, por exemplo, através da exploração de controle de acesso e mecanismos de criptografia, estão disponíveis e podem ser utilizadas para assegurar segurança em dados de contexto. Dos trabalhos pesquisados, CoBrA, Mobile Gaia, Context Toolkit, CORTEX, CASS, SOCAM, MobiSoC, Gaia, e CMF possuem algum suporte a segurança e primitivas de privacidade (CHEN; FININ; JOSHI, 2003) (CHETAN et al., 2005) (DEY; ABOWD, 2000) (SØRENSEN et al., 2004) (FAHY; CLARKE, 2004) (GU; PUNG; ZHANG, 2005) (GUPTA et al., 2007) (RANGANATHAN; CAMPBELL, 2003) (KRANENBURG et al., 2006).

### 3.4 DISTRIBUIÇÃO DE DADOS DE CONTEXTO

A distribuição de dados de contexto busca determinar a melhor forma para o roteamento dos dados em um sistema sensível ao contexto. Por ser dependente da infraestrutura de rede

disponível, as soluções de distribuição de dados podem ser limitadas à rede escolhida. Os fatores que envolvem a distribuição de dados de contexto podem ser organizados em dois principais componentes. O primeiro, chamado de disseminação, considerada as principais políticas para decisão no roteamento dos dados i) qual dado de contexto deve ser distribuído e ii) qual o destino (ou destinos) de um determinado dado. O segundo componente, sobreposição de roteamento, considera que a distribuição dos dados de contexto pode explorar diferentes tipos de redes sobrepostas para conectar e organizar os nós envolvidos. As seções 3.3.1 e 3.3.2 apresentam em mais detalhes esses componentes. A figura 5 apresenta a taxonomia proposta por (BELLAVISTA et al., 2013) para distribuição de dados.

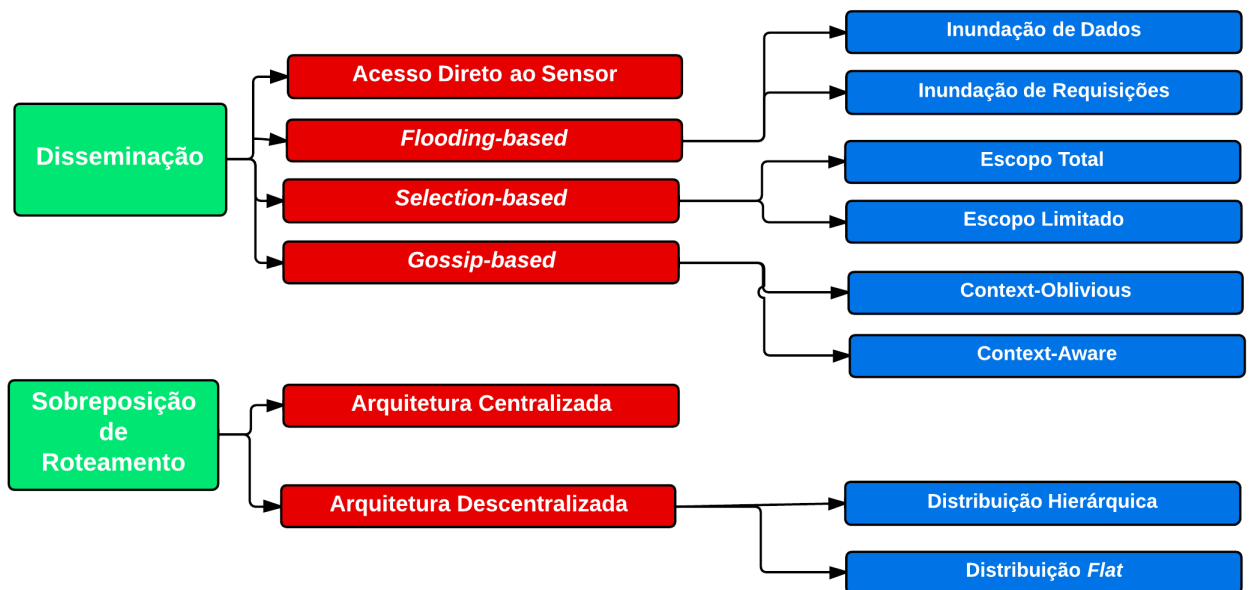


Figura 5: Classificação das abordagens de distribuição de dados de contexto (BELLAVISTA et al., 2013)

### 3.4.1 Disseminação

A disseminação de dados de contexto trata do fluxo de dados entre produtores e consumidores, possuindo impacto direto na escalabilidade do sistema e na disponibilidade dos dados. Diversas soluções foram propostas para disseminação de dados, e podem ser agrupadas em: *flooding-based*, *selection-based*, e *gossip-based*. As duas primeiras categorias caracterizam abordagens determinísticas, onde o coletor definitivamente recebe os dados produzidos por fontes pertencentes ao mesmo sistema de distribuição de dados de contexto. A última categoria é tipicamente de abordagens probabilísticas, onde o coletor pode perder alguns dados de interesse. Algumas soluções, também utilizam abordagens híbridas que misturam as diferentes formas de disseminação.

Abordagens de acesso direto a sensores podem introduzir uma baixa aviabilidade de dados e conflitos espaço-tempo, devido ao fato que os consumidores devem se comunicar diretamente com os fontes para acessar os dados. Entretanto, como principal benefício, essas abordagens possuem um menor custo para organização e suporte. Os serviços de *Cooltown* (DEBATY; GODDI; VORBAU, 2005) acessam diretamente servidores Web para encontrar dados de contexto. Context Toolkit introduz a descoberta para lidar com o registro de fontes de dados de contexto e permitir, assim, uma melhor mobilidade ao dispositivo (DEY; ABOWD, 2000). Através da exploração de uma arquitetura orientada a serviços Web, as requisições em COSINE acessam diretamente os sensores, físicos ou virtuais, pertencentes a rede (JUSZCZYK et al., 2009). SOCAM abastece um serviço para identificar as fontes de dados de contexto, e utiliza o acesso direto aos sensores para captar os dados (GU; PUNG; ZHANG, 2005).

Algoritmos *flooding-based* realizam a disseminação através de operações que alcançam todos os nós contidos em um determinado escopo (ex.: toda a rede). Eles operam tanto por inundação de dados de contexto (*data flooding*) ou por inundação de requisições a dados de contexto (*subscription flooding*). Na inundação de dados de contexto, cada nó realiza um *broadcast* dos dados para que sejam espalhados dentro de todo o sistema, deixando que os consumidores selecionem os dados que desejam localmente. Como exemplo, *Adaptive Traffic Lights* troca os dados de contexto, que são úteis para coordenar o tempo entre as luzes vermelha/amarela/verde para veículos próximos a um cruzamento (GRADINESCU et al., 2007). Em MANIP, cada dado possui uma marcação de localidade física que delimita para aonde o dado será distribuído (MACEDO et al., 2009). Por outro lado, na inundação de requisições, cada nó realiza um *broadcast* de suas requisições para todos os nós existentes para construir uma estrutura de disseminação. Esse esquema propaga as requisições para todos os nós do sistema e assume que cada nó memoriza as requisições de todos os outros nós para realizar localmente a descoberta para onde enviar o dado de contexto. Isto pode reduzir a sobrecarga de largura de banda, divulgando apenas os dados necessários, no entanto, este esquema requer tabelas de roteamento muito grandes, limitando a escalabilidade. Como consequência, nenhum trabalho foi encontrado utilizando essa abordagem.

Algoritmos *selection-based* são tipicamente organizados em duas fases. Na primeira, é deterministicamente construída uma estrutura básica para suporte à disseminação dos dados através da utilização de requisições de contexto. Na segunda, os dados são disseminados somente através da estrutura básica, e fica limitada a enviar dados apenas aos nós interessados. Para construir novas estruturas, os nós devem trocar informações de controle, introduzindo assim sobrecargas na comunicação. Abordagens *selection-based* oferecem dois escopos de visibilidade diferentes as requisições: escopo de todo o sistema e escopo limitado. No primeiro caso, o processo de disseminação assegura que cada requisição será visível por todo sistema, garantindo que todos os consumidores de interesse possam receber o dado. No segundo caso, o

processo de disseminação limita as requisições para um subconjunto de nós, de modo a assegurar princípios de localidade e escalabilidade do sistema. Entretanto, devido ao alcance limitado, é possível que alguns consumidores de interesse não sejam encontrados.

Começando pelas abordagens que possuem o alcance de todo o sistema, CASS utiliza um servidor central para memorizar dados de contexto, e os nós móveis escutam todos os eventos de contexto (FAHY; CLARKE, 2004). C-CAST e COPAL implementam nós centrais que recebem as requisições, e possuem a visibilidade de todo o sistema (KNAPPMEYER et al., 2009) (LI; SEHIC; DUSTDAR, 2010). CMF adota uma infraestrutura fixa, na qual nós distribuídos são responsáveis por coordenar o fornecimento de dados aos nós móveis (KRANENBURG et al., 2006). Pervaho utiliza um sistema para submissão de requisições baseado em localização que possuam as seguintes restrições: cada publicação e requisição possui um escopo visível, e a publicação é entregue para um requisitor ativo somente se o publicador e o requisitor estão na intersecção desses dois escopos (EUGSTER; GARBINATO; HOLZER, 2008).

Por outro lado, considerando as abordagens com escopo limitado, em CORTEX os objetos interagem através da utilização de um serviço de descoberta e de um módulo para submissão dos dados. Devido à camada de implantação, as interações entre os objetos são limitadas, concretizando assim um escopo de visibilidade limitada (SØRENSEN et al., 2004). EgoSpaces também introduz uma abordagem *selection-based* baseada em agentes (JULIEN; ROMAN, 2006). SALES desenvolve a abordagem em uma arquitetura hierárquica estruturada que utiliza princípios de localidade física (CORRADI; FANELLI; FOSCHINI, 2010). Para realizar o roteamento dos dados, SALES introduz requisições de contexto, onde cada requisição captura as necessidades particulares de contexto, e dissemina para parte da arquitetura distribuída.

Algoritmos *gossip-based* disseminam dados de maneira probabilística deixando que cada nó reenvie os dados para um conjunto de nós vizinhos randomicamente selecionados. Dado que essas abordagens não necessitam de infraestruturas complexas de roteamento, na verdade apenas a visibilidade para quais vizinhos o dado deve ser enviado, os protocolos *gossip-based* atendem redes instáveis e com mudanças repentinas, como exemplo de MANETs (FRIEDMAN et al., 2007). Vale ressaltar que, se for ajustada, este tipo de abordagem é capaz de garantir alta confiabilidade e baixa latência, apesar de sua simplicidade. Entretanto, podem exibir um comportamento, em tempo de execução, que dependem estritamente da densidade e mobilidade do nó, e isso pode levar a um desempenho não tão estável. Mais detalhadamente, protocolos *gossip-based* podem ser classificados em duas grandes categorias: *context-oblivious* e *context-aware*.

Protocolos *context-oblivious* usualmente se baseiam em retransmissões randômicas e não consideram nenhuma informação de externa de contexto para modelar o seu comportamento (SASSON; CAVIN; SCHIPER, 2003). Entre as abordagens que utilizam esses protocolos, sistemas puramente probabilísticos simplesmente reenviam o dado recebidos com uma probabi-

lidade de retransmissão, que pode ser diferente para cada nó vizinho e depende da densidade de nó local ou das informações da vizinhança (HAAS; HALPERN; LI, 2006) (CARTIGNY; SIMPLOT, 2003) (DRABKIN et al., 2007) (TILAK; MURPHY; HEINZELMAN, 2003). Por outro lado, em sistemas de disseminação baseadas em contadores, a cada intervalo de tempo um nó recebe novos dados. O nó aguarda um *delay* randômico para ouvir possíveis retransmissões realizadas pelos nós vizinhos, ao final do tempo de *delay* o nó reenvia o dado se, e somente se, tiver ouvido um número de retransmissões menor que um determinado valor limite (CARTIGNY; SIMPLOT, 2003) (HAAS; HALPERN; LI, 2006). É importante salientar que nas abordagens *context-oblivion* a percentagem de nós que recebem os dados aumenta repentinamente quando se aproxima de um limite específico, que é dependente da densidade do nó. Assim, pode-se concluir que o principal benefício destas abordagens está no fato de que elas não demandam grandes recursos computacionais, nem uma estrutura de disseminação pré-definida, pois simplesmente seleciona aleatoriamente os nós na vizinhança. Porém, uma vez que a largura de banda sem fio pode ser inutilmente utilizada por transmitir dados desnecessários, nenhum dos sistemas pesquisou adotou essa abordagem.

Protocolos *context-aware*, selecionam os vizinhos para enviar os dados através da utilização de alguma informação de contexto externa, geralmente pertencente a uma dimensão de contexto diferente. Como exemplo, algumas abordagens utilizam a distância entre os nós (contexto físico) para criar réplicas distantes da original (MIRANDA et al., 2007). Outras abordagens utilizam similaridade social, como membros de uma mesma classe (contexto de usuário), para selecionar os vizinhos que receberão os dados. Em suma, abordagens de disseminação *context-aware* reduzem o número de dados enviados desnecessariamente, porém requerem uma coordenação mais “pesada” para analisar e trocar dados de contexto. HiBOp e CAR exploram as informações de contexto para encaminhamento das mensagens, como por exemplo, selecionar o melhor remetente durante o roteamento de mensagens (BOLDRINI; CONTI; PASSARELLA, 2008) (MUSOLESI; MASCOLO, 2009).

Abordagens híbridas também são encontradas. Active Highways coletam dados de sensores locais/remotos dos veículos, e através de servidores fixos auxiliam os veículos em trânsito, a abordagem *selection-based* com escopo limitado é usada para disseminar os dados entre os servidores (IFTODE et al., 2008). Gaia usa tanto o acesso direto aos sensores quanto as políticas *selection-based* para o escopo de todo o sistema (RANGANATHAN; CAMPBELL, 2003). Finalmente, em MobEyes, dados podem ser coletados por sensores ou captados de outros veículos, os veículos próximos trocam dados utilizando uma abordagem *flooding-based*, onde cada veículo realiza um *broadcast* de dados de contexto captados pelos sensores locais ou recebidos de outros veículos (LEE et al., 2009).

### 3.4.2 Sobreposição de Roteamento

A sobreposição de roteamento trata da organização dos nós envolvidos na disseminação de dados. As arquiteturas podem ser classificadas como centralizadas e descentralizadas. As abordagens centralizadas incluem qualquer possível implementação concentrada (i.e.: servidor único e clusterizado), enquanto as arquiteturas descentralizadas podem ser divididas em duas categorias: distribuição hierárquica e distribuição *flat*. Essas duas categorias podem ajudar a satisfazer o princípio da localidade física, por exemplo, garantindo que cada nó trate apenas dos lugares físicos mais próximos e facilmente acessíveis, além de melhorar a escalabilidade, mesmo que seja exigida uma sobrecarga adicional para gerenciamento dos nós.

Arquiteturas centralizadas são usualmente implementadas em conjunto com infraestruturas fixas sem fio. De maneira detalhada, C-CAST, CoBrA, CASS, e COPAL adotam uma infraestrutura sem fio com um servidor centralizado que armazena dados de sensores e os envia a serviços sensíveis ao contexto em nós móveis (CHEN; FININ; JOSHI, 2003) (FAHY; CLARKE, 2004) (KNAPPMEYER et al., 2009) (LI; SEHIC; DUSTDAR, 2010). Pervaho LPSS, também, utiliza uma infraestrutura sem fio com um servidor JMS centralizado (RICHARDS; MONSON-HAEFEL; CHAPPELL, 2009) (EUGSTER; GARBINATO; HOLZER, 2008). Gaia utiliza uma infraestrutura sem fio com um servidor centralizado que faz buscas de informações de contexto e faz acesso a dados de contexto (RANGANATHAN et al., 2004). MobiPADS também possui uma abordagem centralizada, que utiliza Mobilelets, isto é, entidades que podem migrar para instalar novos códigos e transferir tarefas (CHANG; SHIN; CHUNG, 2007). COSMOS foca principalmente em problemas no processamento de contexto, e assume que todos as fontes de informações de contexto são localmente implementadas no nó que está executando o serviço sensível ao contexto. Assim, esse trabalho foi categorizado como centralizado pois cada nó pode ser visto como um servidor central construindo o seu próprio sistema de distribuição de dados de contexto (CONAN; ROUVOY; SEINTURIER, 2007).

Por outro lado, existem as arquiteturas distribuídas do tipo *flat*, começando por infraestruturas sem fio fixas. Active Highways é implementado em servidores que recebem dados veículos e sensores, que se encontram nas rodovias, coordenados para obter informações do tempo de duração de viagens (IFTODE et al., 2008). Aura faz a migração de aplicações entre diferentes nós para suportar usuários realizando tarefas (SOUSA; GARLAN, 2002). CARMEN possui uma abordagem similar cuja a qual *proxies* móveis seguem o usuário enquanto está em *roaming* (BELLAVISTA et al., 2003). CMF possui um conjunto de nós que são coordenados para enviar dados de contexto de acordo com as necessidades dos serviços (KRANENBURG et al., 2006). Cooltown utiliza a conexão sem fio com a Internet para acessar diferentes aplicações que fornecem dados de contexto (DEBATY; GODDI; VORBAU, 2005). COSINE adota serviços Web distribuídos que provem acesso para os sensores, sendo que a rede final é



do tipo *flat* (JUSZCZYK et al., 2009).

Além disso, para configurações *ad-hoc*, HiBOp, CARISMA, Context Toolkit, CORTEX, Hydrogen, EgoSpaces, MobEyes, MANIP, Habit, CAR, Migratory Services, e RCSM compartilham informações de contexto com os dispositivos fisicamente próximos, caracterizando a abordagem *ad-hoc flat* (BOLDRINI; CONTI; PASSARELLA, 2008) (CAPRA; EMMERICH; MASCOLO, 2003). (DEY; ABOWD, 2000) (SØRENSEN et al., 2004) (HOFER et al., 2003) (JULIEN; ROMAN, 2006) (LEE et al., 2009) (MACEDO et al., 2009) (MASHHADI; MOKHTAR; CAPRA, 2009) (MUSOLESI; MASCOLO, 2009) (RIVA et al., 2007) (YAU et al., 2004). Adicionalmente, mesmo que adotados protocolos de clusterização para gerenciar a rede, do ponto de vista da sobreposição de roteamento, tanto REDMAN como Mobile Gaia adotam uma arquitetura distribuída do tipo *flat* para gerenciar dados de contexto (CHETAN et al., 2005) (BELLAVISTA; CORRADI; MAGISTRETTI, 2005).

Para finalizar, as arquiteturas hierárquicas distribuídas são usualmente encontradas em sistemas que adotam de maneira conjunta a infraestrutura fixas sem fio e a comunicação sem fio *ad-hoc*. De fato, SALES explora os dois tipos de configurações sem fio, e organiza os nós pertencentes ao sistema hierarquicamente para assegurar o princípio da localidade física (CORRADI; FANELLI; FOSCHINI, 2010). Similarmente, HiCon possui uma arquitetura em árvore para sobreposição de roteamento, onde os nós intermediários realizam a agregação de dados para reduzir o número de dados de contexto trocados (CHO et al., 2008).

### 3.5 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo foram apresentados os conceitos que envolvem a sensibilidade ao contexto, bem como os aspectos relativos ao processamento e disseminação de dados de contexto. As abordagens sensíveis ao contexto são muito interessantes para aplicações e/ou serviços que desejam se adaptar de acordo com as alterações que ocorrem no ambiente de execução. Além disso, a sensibilidade ao contexto também permite uma melhor utilização dos recursos computacionais onde os serviços e/ou aplicações estão sendo executados. Sendo assim, para ambientes que possuem limitação quanto aos recursos computacionais disponíveis (ex.: ambientes ubíquos móveis), a sensibilidade ao contexto passa a ser não só interessante, mas altamente desejável.



## 4 PROQUALI: PRODUTOR DE CONTEXTO CONSCIENTE DE QUALIDADE

Nesse capítulo é apresentado a abordagem consciente de qualidade para provedores de contexto em ambientes ubíquos móveis. A proposta visa explorar os aspectos de processamento e armazenamento de dispositivos pervasivos para garantir uma melhor qualidade de contexto, provendo informações pré-validadas às camadas consumidoras.

A proposta foi dividida em três seções: **Políticas de Qualidade de Contexto**, **Processo de Filtragem** e **Provedor de Contexto Consciente de Qualidade**. Na primeira seção, são definidas as políticas de qualidade de contexto, e suas subcategorias. Já a seção Processo de Filtragem, explica o processo utilizado pelos provedores de contexto para aprimorar a qualidade nas informações de contexto, utilizando exemplos para ilustrar algumas situações onde o processo de filtragem pode ser realizado. Finalmente, na seção seguinte, apresentamos o modelo do provedor de contexto consciente de qualidade, bem como as funcionalidades do seus componentes.

Na seção seguinte, são feitas algumas considerações sobre a proposta apresentada, levantando questões que suportam o desenvolvimento desse trabalho de pesquisa. E por fim, na última seção, apresentamos os trabalhos relacionados com essa dissertação com o intuito de fornecer uma base de comparação com o trabalho proposto.

### 4.1 POLÍTICAS DE QUALIDADE DE CONTEXTO

Assumindo que cada aplicação tem objetivos específicos e dado a gama de cenários onde aplicações sensíveis ao contexto podem ser implementadas, não é viável garantir qualidade de contexto, nos mais diversos tipos de aplicações, através de políticas baseadas em apenas alguns parâmetros de qualidade. Por essa razão, nessa seção é definido um conjunto de políticas de qualidade que podem ser utilizadas em diferentes cenários sensíveis ao contexto. As políticas foram divididas em dois grupos, no primeiro, são apresentadas as políticas orientadas a parâmetros de qualidade, que similar as políticas propostas na literatura, utilizam parâmetros de QoC para garantir a qualidade da informação. O segundo grupo, políticas orientadas à aplicação, recorrem aos requisitos das aplicações para delimitar o escopo das informações que devem ser enviadas às aplicações.

#### 4.1.1 Políticas orientadas à parâmetros de qualidade

Políticas orientadas à parâmetros de qualidade exploram parâmetros de QoC para assegurar qualidade em dados de contexto. Diferentes parâmetros de QoC podem ser utilizados para

criar políticas que qualifiquem as informações desejadas pela aplicação, em outras palavras, esse trabalho sugere que várias políticas sejam concebidas conjuntamente com o intuito de avaliar os dados de contexto e garantir qualidade. As políticas podem ser estáticas ou dinâmicas, dependendo de como os parâmetros de qualidade são quantificados.

**Definição 1** *Uma política de QoC estática PQE é uma tupla  $PQE = \{c, q, o, l\}$  onde  $c$  é um parâmetro de contexto definido no modelo de contexto,  $q$  é o parâmetro de qualidade de contexto definido no modelo de QoC,  $l$  é o valor limitante de  $q$  e  $o$  é um operador de comparação.*

Políticas estáticas devem ser utilizadas quando o valor do parâmetro de QoC pode ser pré-definido. Por exemplo, uma aplicação que utiliza a informação provida por um acelerômetro de um dispositivo móvel para detectar movimento, baseando-se na frequência do movimento humano (aproximadamente 0-10 Hz) (LESTER; HANNAFORD; BORRIELLO, 2004). Supondo que para não descaracterizar as inferências feitas pela aplicação a precisão no dado capturado não possa ultrapassar 0.5 Hz, a política  $PQE = \{frequencia, precisao, >, 0.5\}$  poderia ser utilizada no dispositivo para eliminar informações desnecessárias. Ou ainda, o dispositivo deve enviar informações se e somente se a significância da informação de frequência for maior que 90%, outra política de QoC poderia ser definida como  $PQE = \{frequencia, significancia, >, 0.90\}$ . Porém alguns parâmetros de qualidade podem ser calculados em tempo de execução, através de diferentes formas de quantização. Para tal, definimos políticas dinâmicas.

**Definição 2** *Uma política dinâmica PQD é uma tupla  $PQD = \{c, q, r, o\}$  onde  $c$  é um parâmetro de contexto definido no modelo de contexto,  $q$  é o parâmetro de qualidade de contexto definido no modelo de QoC,  $r$  é a relação definida para quantificar o parâmetro de QoC e  $o$  é um operador de comparação.*

Políticas dinâmicas, usualmente, demandam um alto poder computacional, por essa razão devem ser manipuladas com cuidado em dispositivos pervasivos. Caso os parâmetros de qualidade demande uma quantificação através de abordagens complexas (algoritmos genéticos, lógica fuzzy, etc...), o servidor de contexto deve ser requisitado para fornecer essa informação, caso isso não seja possível, as políticas baseadas nesses parâmetros devem ser utilizadas nas camadas presentes no servidor de contexto.

#### 4.1.2 Políticas orientadas à aplicação

Políticas orientadas à aplicação (PA) são utilizadas para delimitar o escopo das informações desejadas pelas aplicações. A ideia que envolve essas políticas consiste em aumentar o número de informações relevantes fornecidas à aplicação. Então, o conjunto de políticas orientadas a aplicação pode ser usado para restringir o espaço amostral de uma determinada informação de

contexto, definindo efetivamente o que é de interesse para cada aplicação. Uma política de aplicação pode ser do tipo estática, se o valor limitante for pré-determinado. Ou dinâmica, caso o valor limitante seja constantemente atualizado.

**Definição 3** *Uma política de aplicação estática PAE é uma tupla  $PAE = \{c, o, l\}$ , onde  $c$  é um parâmetro de contexto definido no modelo de contexto,  $o$  é um operador de comparação e  $l$  é o valor limitante pré-determinado de  $c$ .*

Retomando o exemplo do acelerômetro, sob o ponto de vista prático, não há a necessidade do dispositivo enviar informação quando não ocorre movimento humano, então as políticas de aplicação  $PAE = \{frequencia, \leq, 10\}$  e  $PAE = \{frequencia, \geq, 0\}$ , poderiam ser definidas e usadas para reduzir a quantidade de dados enviados pelo dispositivo móvel. Algumas vezes os valores limitantes não podem ser estabelecidos previamente por serem sensíveis a mudanças ao longo do tempo, por exemplo, suponha agora que o acelerômetro seja utilizado para detectar e classificar diferentes tipos de movimento, de fato, quanto mais informações foram obtidas melhor será a caracterização dos tipos de movimento. Com o histórico dos dados torna-se possível limitar de maneira dinâmica o escopo de interesse para aplicação. Por essa razão, definimos políticas orientadas a aplicação dinâmicas.

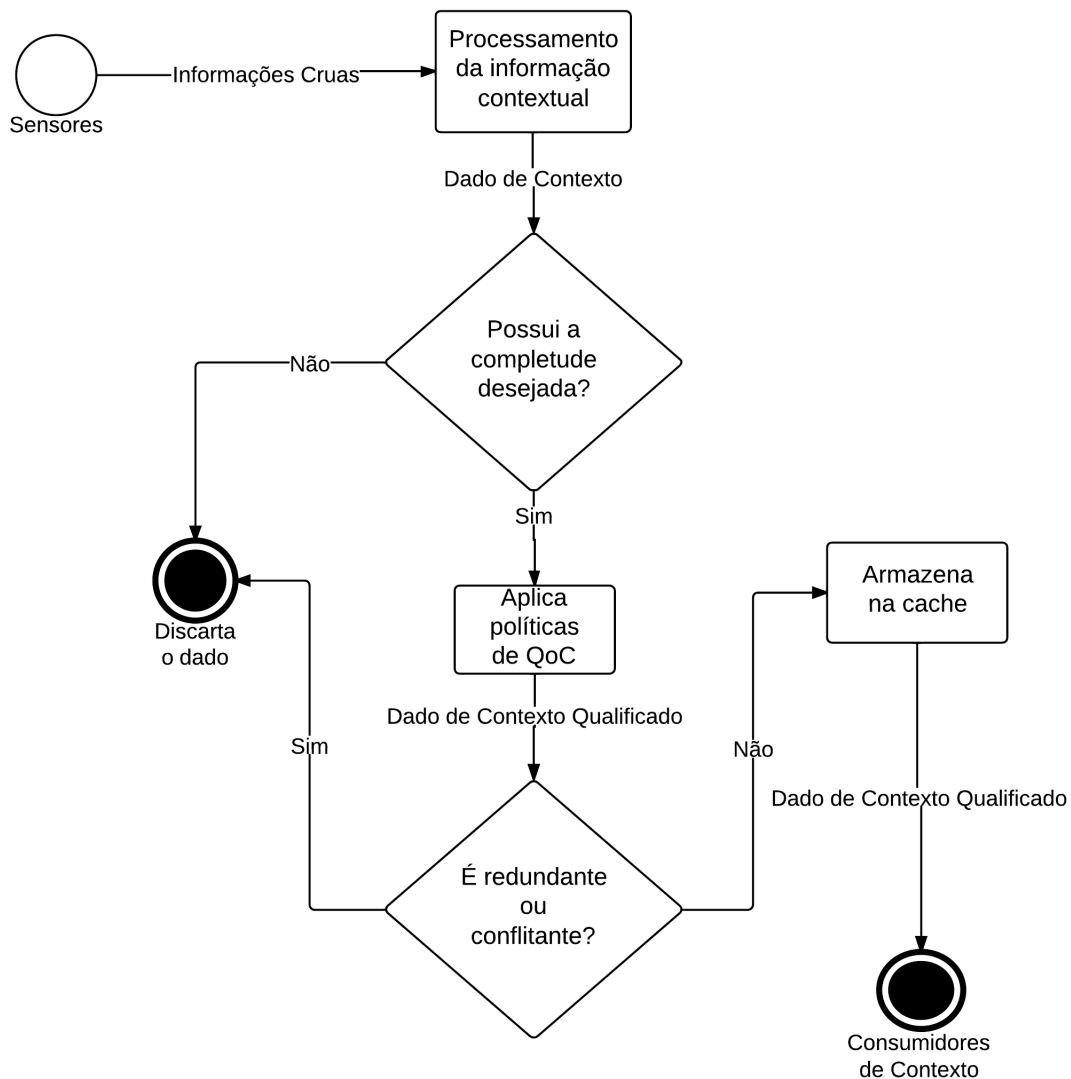
**Definição 4** *Uma política de aplicação dinâmica PAD é uma tupla  $PAD = \{c, o, l_i, r\}$  onde  $c$  é um parâmetro de contexto definido no modelo de contexto,  $o$  é um operador de comparação,  $l_i$  é o valor limitante inicial e  $r$  é a relação atualizadora de  $l$ .*

Apesar de possuírem um potencial interessante, essas políticas não foram exploradas nesse trabalho, dado que as técnicas para aprendizado de máquina, existentes hoje em dia, demandam recursos computacionais nem sempre disponíveis em dispositivos pervasivos. Outras abordagens podem ser utilizadas, como utilizar o servidor de contexto para realizar as inferências sobre o histórico dos dados, porém como o foco desse trabalho é em dispositivos pervasivos, essa abordagem será desenvolvida em trabalhos futuros.

De uma maneira geral, um conjunto de políticas deve ser definido para cada parâmetro presente no modelo de contexto, para que ocorram garantias reais de qualidade nas informações produzidas pelos dispositivos. Nesse sentido, um parâmetro pertencente ao modelo de contexto pode ser limitado em termos da necessidade da aplicação e avaliado por vários parâmetros de QoC, para que seja reportado apenas o que a aplicação necessita.

## 4.2 PROCESSO DE FILTRAGEM

Nessa seção são apresentadas todas as etapas do processo de filtragem de informações de contexto juntamente com exemplos, para clarificar sua abrangência. A figura 6 apresenta o esquema de filtragem.



S

Figura 6: Processo de filtragem de informações de contexto

(SILVA; DANTAS, 2013b)

O processo de filtragem se inicia quando o provedor de contexto recebe a informação “crua” dos sensores. Essa informação é, então, usada para criar um dado de contexto sem nenhuma garantia de qualidade, porém, o novo dado criado já é interoperável em todas as camadas do *middleware* de contexto. Subsequentemente, o dado de contexto é avaliado em termos de completude, assegurando que toda a informação necessária está disponível. Por exemplo, uma aplicação que demanda informações sobre a temperatura em um determinado lugar e em um determinado tempo. O dado de contexto, provido pelo dispositivo pervasivo, deve conter todas as informações solicitadas pela aplicação, isto é, temperatura, hora do dia e localização geográfica. Caso contrário, o dado de contexto é eliminado.

Após assegurar que toda a informação está disponível, são aplicadas as políticas de qualidade (tanto orientadas a parâmetros, quanto orientadas a aplicação) para assegurar QoC no dado de contexto. Suponha, agora, que a aplicação utilizada como exemplo é um sistema de

alerta que notifica os usuários quando a temperatura excede  $c$  graus Celsius em uma sala de *data center*. Algumas políticas que poderiam ser usadas para atingir os objetivos da aplicação, seriam:

- $\{temperatura, precisao, \leq, 1\}$ : A precisão na captação da informação de temperatura deve ser no máximo de 1 grau Celsius.
- $\{temperatura, significancia, >, 0.9\}$ : A significância da informação de temperatura deve ser maior que 90%. Por exemplo, se a temperatura capturada em um determinado tempo  $t_n$  é  $\gg$  que a temperatura capturada em  $(t_{n-1}, t_{n-2}, \dots)$  e  $(t_{n+1}, t_{n+2}, \dots)$ , essa informação possui uma baixa significância e deve ser descartada.
- $\{temperatura, >, 18\}$ : Supondo que o valor limite  $c$  seja 18, a aplicação não necessita enviar informações de temperatura abaixo de 18 graus Celsius.

Após a aplicação das políticas de qualidade, o dado de contexto é confrontado com os dados recém enviados para que sejam identificadas redundâncias e conflitos. Voltando ao exemplo e assumindo que a sala do *data center* tem uma temperatura estável por um certo período  $p$  do dia. Durante  $p$ , os dados criados possuem informações de temperatura similares, e não há a necessidade de enviar essas informações redundantes. Por outro lado, se a temperatura exceder um determinado limite, a aplicação deve enviar a informação até que a situação seja resolvida. Nessa caso, os dados redundantes são desejáveis, dado que uma condição crítica deve ser identificada pela a aplicação. Portanto, a caracterização de uma situação redundante para a aplicação deve ser feita, também, através de políticas de qualidade.

Como mencionado nos capítulos anteriores, a informação provida por sensores pode diferir da realidade, levando-se em consideração a imprecisão intrínseca encontrada em muitos desses dispositivos. Tal informação pode causar situações de conflito. Usando o mesmo exemplo anterior, imagine que a temperatura da sala do *data center* está estável em  $c$  graus Celsius em um determinado momento  $t_n$  em  $p$ . E que os sensores capturam a informação de temperatura com o valor de  $c + l$  graus Celsius em  $t_n$ , onde  $l$  é o valor medido excedente causado pela imprecisão do dispositivo. Se  $c + l > q$ , onde  $q$  é o valor limite para alertar os usuários, uma situação de conflito pode ocorrer e, portanto, a informação de contexto deve ser previamente eliminada.

Agora imagine que a sala do *data center* possui sensores de diferentes fornecedores, que também são responsáveis por informar a temperatura da sala. Se a informação provida por dois sensores diferentes divergirem, uma situação de conflito ocorre, e deve ser resolvida através do uso de políticas de QoC. Mas nesse caso, o servidor de contexto deve ser responsável por identificar e eliminar o dado conflitante. Isso é justificado pelo fato de que nessa abordagem, e também nas demais encontradas na literatura, os diferentes provedores de contexto não conhecem uns aos outros (trabalhos futuros).

Os exemplos acima foram usados para caracterizar algumas situações onde o processo de filtragem pode ser realizado. As situações mencionadas acima podem ser projetadas para aplicações que possuem um número maior de variáveis, habilitando a ocorrência de um maior número de situações indesejadas. Isso implica que cada aplicação deve possuir políticas específicas para lidar com conflitos e redundâncias, para que a qualidade de contexto seja, de fato, assegurada.

O processo de filtragem busca reduzir o número de dados enviados através da rede, e como consequência um menor poder computacional é requerido ao servidor para processar dados de contexto. Além disso, operações de refinamento e inferência sob dados de contexto podem trazer melhores resultados, dado que, com a filtragem, são utilizados dados não-redundantes e que foram validados através da necessidade da aplicação. Por uma questão de clareza, não está sendo sugerido aqui que toda a qualidade de contexto deva ser assegurada pelos provedores de informação de contexto, mas um balanço na distribuição de responsabilidades entre os provedores de contexto e os consumidores de contexto, no sentido de otimizar a utilização dos recursos computacionais disponíveis.

#### 4.3 MODELO DO PROVEDOR DE CONTEXTO

Nessa seção são apresentados os componentes do proposto provedor de contexto descrevendo suas funcionalidades. O provedor de contexto consciente de qualidade procura explorar a capacidade de processamento e armazenamento dos dispositivos ubíquos/pervasivos com a finalidade de assegurar uma melhor qualidade de contexto.

O provedor de contexto provê dados de contexto pré-validados para os seus consumidores, sejam eles servidores de contexto ou as próprias aplicações, removendo redundâncias e conflitos nas informações de contexto. O modelo proposto é apresentado na figura 7.

##### 4.3.1 Criador de Dados de Contexto

O componente criador de dados de contexto (CDC) lida com o hardware específico do dispositivo pervasivo. Esse componente tem a função de capturar as informações cruas de contexto providas pelos sensores do dispositivo e transforma-las em um dado de contexto interoperável para a *middleware* de contexto. CDC utiliza as definições do modelo de contexto para padronizar a informação recém recebida. Algumas informações de contexto podem ser providas pelas próprias aplicações, CDC também é responsável por interoperabilizar essas informações.

Em ambientes heterogêneos, informações “cruas” de contexto são providas por uma série de diferentes dispositivos ubíquos, cada um possuindo seu próprio padrão de dados. CDC garante que o provedor de contexto só envia a informação que pode ser manipulada em todas as





são utilizadas pelo IQoC para eliminar dados de contexto incompletos.

### 4.3.3 Validador de Contexto

O validador de contexto (VC) valida as informações de contexto através do uso de políticas de QoC. Para o correto uso das políticas, é necessário que os parâmetros de QoC já estejam quantificados. Concomitantemente, VC também é responsável por quantizar parâmetros de qualidade. O processo de filtragem pode ser realizado de formas diferentes, dependendo do tipo da política (estática ou dinâmica) em que o parâmetro está sendo utilizado.

- *Políticas de QoC estáticas*: Nesse tipo de política, os valores dos parâmetros de QoC são pré-definidos. VC, então, relaciona o parâmetro de QoC com a informação recém recebida através do operador de comparação definido na política, para avaliar a informação em termos de qualidade de contexto.
- *Políticas de QoC dinâmicas*: Aqui os valores dos parâmetros de QoC são calculados dinamicamente baseado na relação definida na própria política. Após calculado, o operador de comparação e o parâmetro de qualidade se usados para avaliar a informação em termos de qualidade.
- *Políticas de aplicação estáticas*: Essas políticas funcionam similarmente as políticas de QoC estáticas, porém não são utilizados os parâmetros de QoC para a comparação, e sim os valores de interesse para a aplicação.
- *Políticas de aplicação dinâmicas*: O valor de interesse da aplicação é atualizado através da relação definida na própria política e comparado com o parâmetro de contexto a ser validado.

Após assegurar qualidade no novo dado de contexto, VC consulta a cache de contexto coletando todos os dados enviados recentemente para encontrar redundâncias, a aplica as políticas de qualidade para remove-las. Algumas aplicações podem requerer dados redundantes para realizar análises futuras sob uma situação de contexto. Por essa razão, a caracterização de dados redundantes é realizada através de políticas de QoC. Dessa forma, cada aplicação pode ter sua própria definição de dados de contexto redundantes.

O processo de filtragem busca melhorar a qualidade dos dados que são transmitidos, certificando que somente dados relevantes são enviados as aplicações. Entretanto, apesar do constante crescimento na capacidade de processamento dos dispositivos pervasivos, os algoritmos de filtragem devem ser projetados considerando as limitações dos dispositivos móveis. Algoritmos de alta complexidade devem ser executados nas camadas do servidor de contexto. Portanto, quando algum parâmetro não é pré-definido e demanda alto poder de processamento para ser

calculado, as camadas do servidor são acionadas para prover a informação através da camada de comunicação do dispositivo, se necessário.

#### **4.3.4 Cache de Contexto**

A cache de contexto (CaC) armazena em tempo de execução dados de contexto válidos. Ademais, a cache se comunica com a camada de comunicação do dispositivo (Distribuidor de dados de contexto), transferindo e requisitando dados de contexto válidos e informações de qualidade, quando necessário. CaC também atua como um gerenciador de persistência no dispositivo móvel, explorado o espaço de armazenamento do dispositivo para salvar dados de contexto, reduzindo, assim, o número de requisições feitas ao servidor de contexto. A cache de contexto realiza operações no repositório de contexto para deletar dados inválidos e atualizar valores nas informações armazenadas.

Em ambientes ubíquos móveis, usualmente acontece de o provedor de contexto ser também um consumidor. Aplicações podem realizar as operações sensíveis ao contexto mais rapidamente utilizando a informação armazenada no próprio dispositivo. Além disso, dispositivos móveis encaram diferentes contextos computacionais, cujo os quais podem causar problemas de conectividade, então a cache pode ser explorada para armazenar os dados até que a conexão seja restabelecida e o dado possa ser enviado.

Outro ponto importante está relacionado ao reprocessamento desnecessário que não deve acontecer em ambientes ubíquos dada a limitação energética dos dispositivos móveis. Parâmetros de QoC já quantificados devem ser mantidos e gerenciados pela CaC até que novos valores sejam calculados impedindo, assim, o retrabalho. Parâmetros de QoC que requerem cálculos dinâmicos e complexos, devem ser quantizados pelo servidor de contexto, e cabe CaC solicitar e armazenar esses valores com o intuito de acelerar o processo de filtragem dos dados de contexto.

#### **4.3.5 Distribuidor de Dados de Contexto**

DDC distribui dados de contexto para os consumidores. Além disso, o distribuidor de dados de contexto também pode receber dados das camadas de comunicação do servidor, por exemplo quando existe algum impedimento para a quantificação da informação no provedor de contexto. Esse componente interage com a cache de contexto em duas situações diferentes: i) no caso de qualquer limitação na conexão que impeça o envio dos dados de contexto, CDD armazena na cache os dados; ii) quando CaC solicita uma informação para ser utilizada pelo validador de contexto, CDD envia uma requisição para o servidor de contexto solicitando o dado faltante. O distribuidor de dados de contexto implementa todos os protocolos de coordenação e

disseminação requeridos para transportar os dados de contexto aos nós interessados.

O distribuidor de dados de contexto foi projetado para garantir qualidade de contexto na distribuição de dados (ex.: tempo de entrega do dado, confiabilidade, ...). Alguns parâmetros de QoC são definidos através de aspectos temporais (ex.: atualidade, prazo de validade,...), e *delays* na distribuição do dado podem alterar a validade desses parâmetros de qualidade. Usualmente, a distribuição de dados de contexto é realizada através de infraestruturas sem fio de melhor-esforço que podem introduzir *delays* e perda de pacotes (BELLAVISTA et al., 2013), nesse sentido esse componente foi concebido para evitar imprecisões adicionais nos dados de contexto.

A proposta apresentada tem como objetivo estabelecer a qualidade de contexto, também, no provedor de contexto. Isso pode ser alcançado explorando as facilidades providas por dispositivos pervasivos para realizar operações que assegurem a correta distribuição de dados com qualidade para as aplicações sensíveis ao contexto. Além disso, foram definidas políticas de QoC com a intenção de caracterizar o escopo de quais informações são relevantes para cada aplicação. De uma maneira mais abrangente, esse proposta visa melhorar a qualidade dos dados transmitidos entre produtores e consumidores de contexto.

#### 4.4 CONSIDERAÇÕES SOBRE A PROPOSTA

Nessa seção é discutida a abrangência da proposta em relação a sua contribuição para infraestruturas sensíveis ao contexto. Dessa forma, é considerado que os aspectos levantados nesse trabalho de pesquisa devam ser contemplados na concepção de *middlewares* de contexto. Como definido em (KJÆR, 2007), um *middleware* é um sistema de software que provê uma camada abstrata entre o sistema operacional e as aplicações em um sistema distribuído. Mais especificamente, o *middleware* de contexto tem como objetivo gerenciar, de maneira transparente, as principais fases que envolvem o provimento de dados de contexto para consumidores sensíveis ao contexto, isto é, representação, agregação, distribuição, garantia de qualidade, etc. (BALDAUF; DUSTDAR; ROSENBERG, 2007). Dessa forma, a figura 8 apresenta a sugestão desse trabalho para a arquitetura lógica de sistemas sensíveis ao contexto.

Dispositivos pervasivos são usualmente utilizados como produtores de informações “cruas” de contexto. Entretanto, sugerimos nessa proposta que *middlewares* de contextos devem usufruir da infraestrutura de hardware dos dispositivos, considerando o poder computacional encontrado em tais equipamentos. De forma mais explícita, explorar as facilidades dos dispositivos pervasivos em um *middleware* de contexto contribui para a melhora da qualidade dos serviços providos, em termos de:

- *Interoperabilidade*: Dados providos por dispositivos pervasivos em ambientes heterogêneos são criados e distribuídos segundo os padrões específicos de cada dispositivo. O dispo-

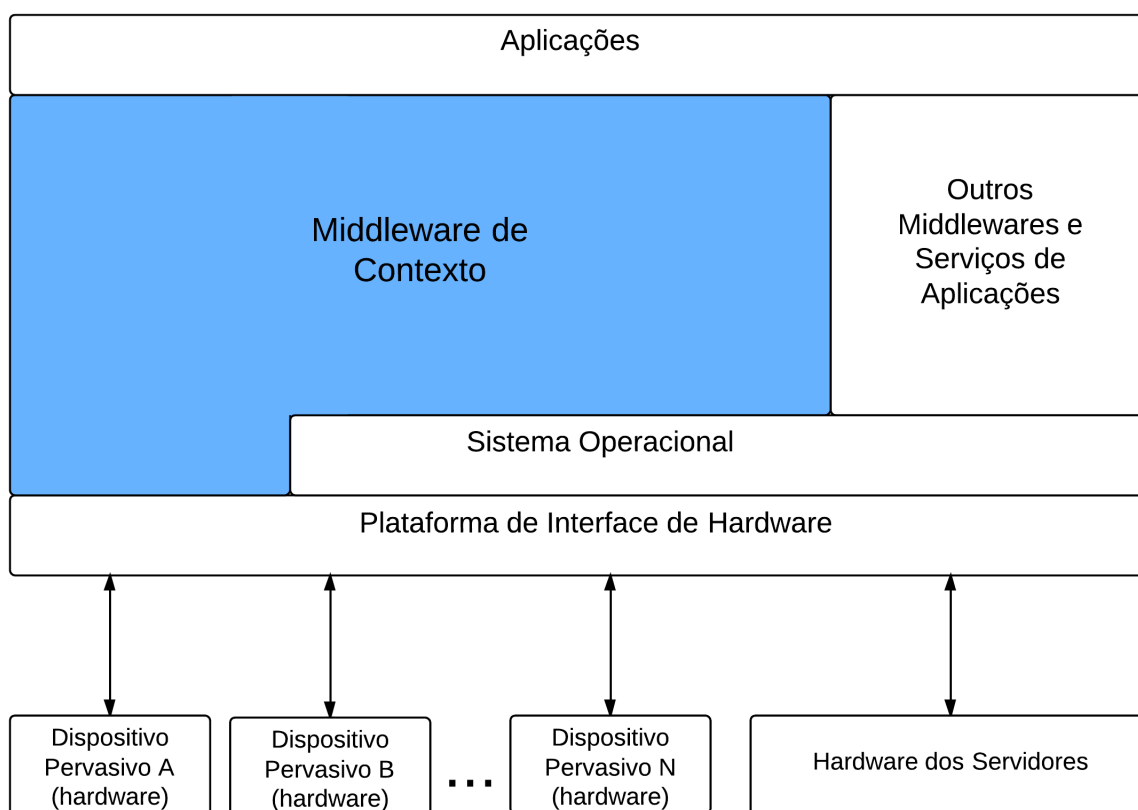


Figura 8: Arquitetura lógica de sistemas sensíveis ao contexto

sitivo, então, pode ser explorado para prover informações de contexto padronizadas e interoperáveis em todo *middleware* de contexto, através de modelos de dados de contexto pré-definidos no próprio dispositivo.

- *Distribuição*: A comunicação sem fio é suscetível à alguns problemas tais como desconexões temporárias, mudanças topológicas frequentes, e limitações na garantia de entrega dos dados. Protocolos de coordenação e disseminação de dados devem ser implementados no dispositivo pervasivo, bem como métodos e algoritmos, com o intuito de distribuir propriamente os dados de contexto.
- *Qualidade de Contexto*: Tipicamente, provedores de contexto provêm informações sob um escopo definido. Algoritmos dedicados e políticas baseadas em parâmetros de qualidade podem ser utilizados para qualificar os dados providos, levando-se em consideração as limitações de cada dispositivo.
- *Armazenamento*: Dispositivos móveis, atualmente, possuem uma considerável capacidade de armazenamento. Os processos de adaptação podem ocorrer mais rapidamente se os dados de contexto forem armazenados no próprio dispositivo, já que o acesso aos dados seria realizado localmente. Ademais, nos casos de limitação na conexão, o espaço de armazenamento pode ser utilizado para resguardar os dados de contexto até que esses possam ser enviados, evitando assim, perda de informações.

Dado as facilidades providas pelos dispositivos pervasivos, entende-se que um *middleware* de contexto deve explorar esses equipamentos para que o gerenciamento no provimento dos dados de contexto seja realizado em sua completude. Além disso, o controle dos dados desde a captura, possibilita ao *middleware* descartar dados indesejados antecipadamente, assim, ocupando de maneira otimizada os recursos computacionais disponíveis. Dessa maneira, *middlewares* que possuem consciência da infraestrutura computacional disponível, tendem a prover dados de contexto com qualidade para as aplicações que suportam.

Outro ponto discutido nessa proposta está relacionado, justamente, com a melhora na qualidade nos dados de contexto. A proposta defende que aperfeiçoar a qualidade de contexto não se limita a avaliar parâmetros de qualidade e garantir a distribuição dos dados de contexto. Os requisitos das aplicações devem ser levados em consideração para que seja possível a utilização de dados com qualidade de contexto. Ou seja, mesmo com a validação feita através de parâmetros de qualidade, as informações de contexto podem não ser desejadas pelas aplicações, sendo que tal informação pode ser conflitante e/ou redundante. Resumidamente, um conflito ocorre quando dois (ou mais) dados de contexto fornecem informações divergentes sobre uma mesma situação, já um dado redundante é identificado por possuir a mesma carga semântica de outro dado já existente.

Um exemplo será utilizado para elucidar os conceitos de conflito e redundância em dados de contexto na perspectiva da aplicação. Então, imagine uma aplicação que pretende utilizar informações de contexto para identificar padrões nas trajetórias dos usuários. Um certo usuário fornece informações para aplicação através do GPS do carro e do *smartphone*. No tempo  $t_1$ , o GPS do carro informa que o usuário está na posição geográfica  $(l_1, l_2)$ , e o *smartphone*, ao mesmo tempo  $t_1$ , informa que o usuário está em  $(k_1, k_2)$ , onde  $k_1 \neq l_1$  e  $k_2 \neq l_2$ . Essa situação caracteriza um conflito e deve ser resolvida. Perceba que essa situação pode ocorrer mesmo que os dados tenham sido avaliados por parâmetros de contexto e tenham sido distribuídos com qualidade.

Utilizando a mesma aplicação como exemplo, imagine um usuário parado em um determinado local (ex: durante o trabalho) durante  $h$  horas. O GPS irá fornecer informações com uma carga semântica muito parecida sobre a localização geográfica desse usuário durante  $h$ , em outras palavras, um conjunto de dados redundantes será enviado, consumindo recursos computacionais desnecessariamente. A proposta considera, então, que a qualidade de contexto só pode ser plenamente alcançada se os modelos e *middlewares* de contexto considerarem os requisitos das aplicações para qualificar e distribuir as informações, i.e, comunicando apenas dados que contribuem para o bom funcionamento dos serviços sensíveis ao contexto.

## 4.5 TRABALHOS CORRELATOS

São apresentados, nessa seção, os trabalhos relacionados com a pesquisa conduzida nessa dissertação, bem como a descrição e caracterização dos aspectos utilizados para estabelecer a comparação entre os diferentes trabalhos. Dessa forma, são apresentadas as principais características utilizadas nos trabalhos para descrever um provedor de contexto capaz de fornecer informações de qualidade para seus consumidores. As características foram agrupadas em quatro diferentes categorias:

- **Representação de dados de contexto:** A definição e representação é de fundamental importância para a comunicação e processamento de dados de contexto. Porém, deve-se considerar que por um lado a comunicação de dados de contexto em redes sem fio, entre dispositivos móveis limitados, exige uma representação leve, enquanto que as operações de inferência sobre as situações do indivíduo e do ambiente demandam uma representação através de modelos mais ricos em informação. Além disso, como observado em (STRANG; LINNHOFF-POPIEN, 2004), a representação de contexto para operações semânticas mais exigentes requer uma modelagem capaz de: permitir a validação parcial da informação, independentemente das inter-relações complexas; permitir uma expressividade rica em informações e formalismo para a compreensão compartilhada; indicações para a qualidade da informação.
- **Processamento de dados de contexto:** Diversas operações englobam o conjunto de demandas de processamento para dados de contexto. Como apresentado em 4, as operações de processamento podem ser classificadas em diferentes grupos, e que possuem os seguintes desafios para a implantação em ambientes ubíquos móveis:
  - *operações sob o histórico dos dados:* dependem diretamente da limitada capacidade de armazenamento (temporário ou permanente) dos dispositivos ubíquos móveis.
  - *operações de raciocínio:* devem ser implementadas com o consentimento das limitações de processamento e memória dos dispositivos ubíquos móveis.
  - *operações de filtragem:* devem ser definidos parâmetros para filtragem dos dados em termos da semântica dos dados, dos requisitos da aplicação, e da qualidade de contexto.
  - *operações para segurança dos dados:* as questões de segurança em geral, e em particular as questões de privacidade, devem estar presentes no esquema de roteamento ciente de contexto, uma vez que, eventualmente, os dados de contexto podem incluir informações sensíveis sobre o usuário.

- **Disseminação de dados de contexto:** A disseminação de dados de contexto não é só comprometida pela infraestrutura de rede escolhida para propagação dos dados. A desconexão do próprio dispositivo ubíquo pode prejudicar o correto funcionamento dos consumidores dos dados. Assim um provedor de contexto deve possuir mecanismos capazes de efetuar a sua entrada e saída da rede, ou seja, a disseminação deve ser desacoplada das outras fases do processo de provimento dos dados. Ademais, a abordagem escolhida de disseminação dos dados reflete diretamente na escalabilidade do sistema, que um fator fundamental para a implantação de sistemas de grande porte.
- **Qualidade de contexto:** Como explicitado na seção 3.2 desse documento, garantias de qualidade de contexto são fundamentais para o correto funcionamento das aplicações e serviços sensíveis ao contexto. Dessa forma, se faz necessário o projeto de módulos capazes de atribuir aspectos de qualidade à informação captada pelo dispositivo ubíquo, bem como operações que realizem validações baseadas em informações de QoC, garantindo assim um melhor aproveitamento das informações providas pelo dispositivo. Além disso, ao prover apenas dados com qualidade, o dispositivo ubíquo permite que as operações sob dados de contexto sejam realizadas sob dados já qualificados, o que pode melhorar o resultado das operações em relação aos objetivos dos serviços sensíveis ao contexto.

A proposta de (BOLDRINI; CONTI; PASSARELLA, 2008) denominada HiBOP adota uma abordagem de filtragem para trocar dados de contexto e manter a infraestrutura de roteamento. Assim, são realizadas operações de inferência nos dados recebidos para que sejam descobertos novos comportamentos dos usuários. Os dispositivos móveis enviam e recebem dados de contexto para os vizinhos utilizando diferentes abordagens de filtragem de dados (ex.: baseada em tempo, baseada em mudança). Os dados são representados em um modelo chave-valor. A qualidade de contexto não é contemplada nesse modelo, porém como são definidos diversos parâmetros e esses são utilizados para realizar a validação dos dados, pode-se dizer que existe suporte para a definição de parâmetros de QoC, apesar de isso não ser dito claramente no trabalho.

Solar (CHEN; LI; KOTZ, 2008) foca na escalabilidade para ambientes pervasivos, utilizando técnicas de filtragem para minimizar o envio dos dados. Dessa forma, a congestão na largura de banda e a sobrecarga no dispositivo são reduzidas. Porém, nenhum outro tipo de processamento é realizado nos dados de contexto. Além disso, apesar de ser citada na pesquisa a qualidade de contexto não é contemplada de maneira definitiva no processo de filtragem de dados. Os dados são representados tanto por modelo chave-valor, como pela abordagem orientada a objetos.

Em (KNAPPEMEYER et al., 2009) é apresentado um modelo para provedores de contexto que possui suporte para as principais funcionalidades desejadas em um provedor de contexto, isto é, representação, processamento, disseminação e qualidade de contexto. Para representar



os dados de contexto foi definida uma linguagem (ContextML), que é por definição pertencente ao grupo de representação denominado esquema de marcação. Quanto ao processamento dos dados, é fornecido suporte para as operações de filtragem e agregação dos dados. Além disso, o modelo também suporta o armazenamento de dados permitindo assim operações de histórico. Para disseminação dos dados foram definidas duas interfaces, uma para aquisição das informações dos sensores e outra para o envio dos dados de contexto. A qualidade de contexto não é parte integrante do modelo proposto porém a linguagem de representação permite a inserção de alguns parâmetros de qualidade, mais especificamente, resolução, precisão e grau de incerteza.

No *framework* COPAL proposto em (LI; SEHIC; DUSTDAR, 2010) os dispositivos ubíquos são publicadores de contexto. Os chamados *Wrappers* são responsáveis por “esconder” os dispositivos, e fornecer as informações de contexto. Os *Wrappers* possuem uma camada de comunicação que contempla a heterogeneidade dos sensores e trata de problemas relativos a conectividade. Porém, o provedor de contexto de COPAL não possui uma forma de representação definida, bem como operações para o processamento dos dados. Além disso, não são caracterizados aspectos de qualidade de contexto. Todas essas funções são excedidas por um centralizador de informações de contexto, que recebe a informação dos diversos dispositivos que compõem o sistema.

Já em (MIRON et al., 2010) os provedores de contexto possuem suporte a representação de dados através de ontologias. Os dados podem ser processados através de métodos de agregação que são definidos, em parte, por aspectos de qualidade de contexto. O modelo também prevê o armazenamento de dados de contexto no provedor. Os provedores realizam a comunicação com os sensores e enviam os dados para um centralizador de informações de contexto, porém, o componente de disseminação não é desacoplado no modelo. Além disso, as informações dos provedores não atingem diretamente as aplicações e serviços sensíveis ao contexto, o centralizador, então, é o único consumidor direto das informações fornecidas pelos provedores de contexto. A qualidade de contexto é contemplada pelo suporte a parâmetros de qualidade, que podem ser calculados no próprio dispositivo e são inseridos nos dados de contexto para futuras validações.

Dado as características citadas e a descrição dos trabalhos correlatos, apresentamos na tabela 4 uma comparação entre os trabalhos relacionados e a proposta dessa dissertação. A tabela comparativa indica se os modelos dos provedores de contexto definidos nos trabalhos suportam os principais aspectos relacionados ao provimento de dados de contexto. É importante ressaltar que estamos tratando exclusivamente do provimento de dados por dispositivos ubíquos, pois alguns trabalhos suportam as características indicadas em diferentes camadas de contexto (ex.: camadas do servidor, camadas de *brokers* de contexto).

Referência	Representação	Processamento				Disseminação	Qualidade de Contexto			
		<i>Histórico</i>	<i>Raciocínio</i>	<i>Filtragem</i>	<i>Segurança</i>		<i>Parâmetros</i>	<i>Políticas</i>	<i>Operações</i>	<i>Req. Aplicações</i>
(CHEN; LI; KOTZ, 2008)	sim	não	não	sim	não	sim	não	não	não	não
(BOLDRINI; CONTI; PASSARELLA, 2008)	sim	sim	sim	sim	não	sim	sim <sup>1</sup>	não	não	não
(KNAPPEMEYER et al., 2009)	sim	sim	sim	sim	não	sim	sim <sup>1</sup>	não	não	não
(MIRON et al., 2010)	sim	sim	sim	não	não	não	sim	não	sim	não
(LI; SEHIC; DUSTDAR, 2010)	não	não	não	não	não	sim	não	não	não	não
<b>ProQuali</b>	<b>sim</b>	<b>sim</b>	<b>sim</b>	<b>sim</b>	<b>não</b>	<b>sim</b>	<b>sim</b>	<b>sim</b>	<b>sim</b>	<b>sim</b>

Tabela 4: Comparação entre os provedores de contexto propostos

Para concluir essa seção caracterizamos a contribuição desse trabalho, em relação aos demais apresentados, como sendo o provimento de dados com qualidade de contexto. Na abordagem proposta a qualidade de contexto é contemplada em todos os componentes do modelagem permitindo que o provedor de contexto pré-valide as informações com garantias de qualidade, através de parâmetros de qualidade, políticas de qualidade, operações de qualidade e requisitos das aplicações. A proposta contemplou as principais características necessárias para se estabelecer um provedor de contexto, utilizando da qualidade de contexto para fortalecer tais aspectos.

#### 4.6 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo foi apresentada a abordagem ProQuali: Produtor de Contexto Consciente de Qualidade. A proposta objetiva contemplar as principais fases relativas ao provimento de dados de contexto. Para tal, ProQuali se baseia nos conceitos estabelecidos na literatura para o provimento de dados de contexto, isto é, captura, processamento, qualificação e distribuição dos dados. Além disso, a proposta também define um processo de filtragem que utiliza de políticas de QoC para assegurar a qualidade dos dados providos. Assim, a modelagem busca utilizar de maneira otimizada os recursos energéticos dos dispositivos ubíquos através da redução da quantidade de dados enviados e processados.

Em relação aos trabalhos relacionados, ProQuali apresenta um diferencial quanto a garantia de qualidade de contexto nos dados providos pelos produtores de contexto. Apesar das demais propostas possuírem suporte a alguns parâmetros de contexto e a processos de filtragem e raciocínio, nenhum dos provedores de contexto pesquisados possuem garantias de QoC em todas as etapas do provimento de dados de contexto. Dessa forma, a modelagem sugerida nesse capítulo juntamente com as políticas e o processo de filtragem contemplam a qualidade de contexto da captura a disseminação dos dados, possibilitando aos dispositivos pervasivos produzirem dados de contexto com consciência de qualidade.

## 5 AMBIENTE E RESULTADOS EXPERIMENTAIS

Nesse capítulo, descrevemos o ambiente e os resultados experimentais relacionados aos nossos testes. A abordagem proposta foi implementado em uma aplicação que tinha por objetivo descobrir e representar as trajetórias dos usuários. Para a avaliação da proposta foram analisados os seguintes aspectos: i) se houve, ou não, redução nos dados enviados pelos provedores de contexto utilizando políticas orientadas à aplicação e orientadas à parâmetros de QoC, ii) o impacto dos recursos computacionais do dispositivo móvel pelo uso da abordagem proposta e iii) se as trajetórias foram melhor representadas devido ao uso do ProQuali.

### 5.1 DESCRIÇÃO DO AMBIENTE EXPERIMENTAL

A figura 9 ilustra o cenário de aplicação dos testes. O ambiente experimental simulado é composto por diversos dispositivos ubíquos enviando informações para um servidor (consumidor de contexto). Cada dispositivo envia informações sobre a localização do usuário, e o servidor processa e envia as informações para as aplicações de interesse.

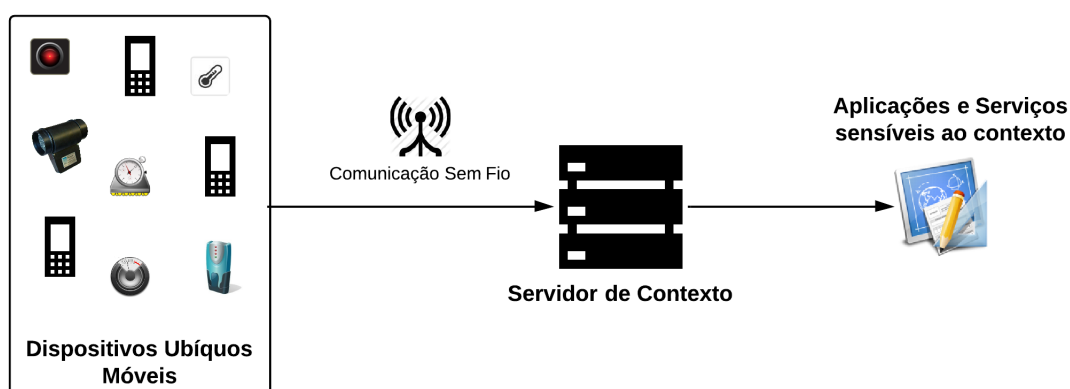


Figura 9: Figura ilustrativa do ambiente experimental

O ambiente experimental utilizado em todos os testes é apresentado na tabela 6. Os provedores de contexto foram implementados em máquinas virtuais e o emulador de Android foi utilizado para simular o dispositivo móvel. O esquema lógico do *framework* utilizado nos experimentos, foi baseado no trabalho de (MIRON et al., 2010) e, é apresentado na figura 10. O *framework* é composto por provedores de contexto (definidos na proposta), e por camadas no servidor que são descritas a seguir:

- **Coletor de Contexto:** coleta e agrega as informações enviadas pelos provedores de contexto, construindo um contexto global do usuário. Comunica-se diretamente com o Avaliador de QoC com o intuito de verificar as dimensões de qualidade que não puderam ser

	Sistema Operacional	Processador	Memória	Armazenamento	Base de dados	Linguagem de programação	Ambiente
Provedores de Contexto	Android 4.1	Quad-core 1.4 GHz	512MB	16GB	GvSig 0.3	Java 1.7	Android Emulador
Servidor de Contexto	Ubuntu 12.04 64-bit	Intel i5 2.5GHz	4GB	1TB	PostGIS 2.0	Java 1.7	Dell Vostro 3460

Tabela 5: Ambiente experimental

concebidas pelos provedores de contexto. Os contextos globais dos usuários são salvos no Repositório Global de Usuários.

- **Racionalizador de Contexto:** executa tarefas de inferência e derivação em dados de contexto. Cria novos contextos baseados nas informações disponíveis, e infere comportamentos futuros. Além disso, novas características e informações são descobertas por esse componente. Algoritmos de alta complexidade são executados pelo racionalizador de contexto.
- **Ofuscador de Contexto:** reforça as políticas de privacidade de usuários em informações de contexto, executando processos de ofuscação e anonimização. As políticas de privacidade são usadas com a finalidade de limitar e/ou generalizar o nível de divulgação das informações de contexto.
- **Avaliador de QoC:** é o serviço responsável por avaliar a qualidade de contexto no servidor. Os parâmetros de QoC são quantizados dinamicamente para avaliar todas as dimensões de qualidade dos dados de QoC. O processo de qualidade realizado nesse componente pode ser dividido em duas etapas: i) Os aspectos de qualidade que não foram validados nos provedores de contexto, são aqui avaliados; ii) A qualidade de contexto é reavaliada nas informações inferidas/derivadas pelo Racionalizador de Contexto. As políticas de qualidade são utilizadas para filtrar e qualificar os dados de contexto.
- **Fornecedor de Contexto:** responde diretamente as requisições das aplicações. O componente busca dados com maior qualidade de contexto e os fornece às aplicações solicitantes.

No ambiente simulado, foram usados dados reais de GPS coletados de trajetórias de quatro diferentes veículos. Os veículos número 1 e 2 foram monitorados durante três horas não consecutivas gerando, cada um, um total de 10800 dados de contexto. O veículo 3 foi monitorado durante uma hora e meia, em uma trajetória única, gerando 5400 dados de contexto. O veículo 4 foi monitorado durante uma semana num total de 5 horas e 23 minutos gerando 19380 dados de contexto. O intervalo de precisão do GPS é de 3m-5km, que é justificado pela quantidade de construções ao redor, perdas na conexão, e limitação do próprio dispositivo. Por

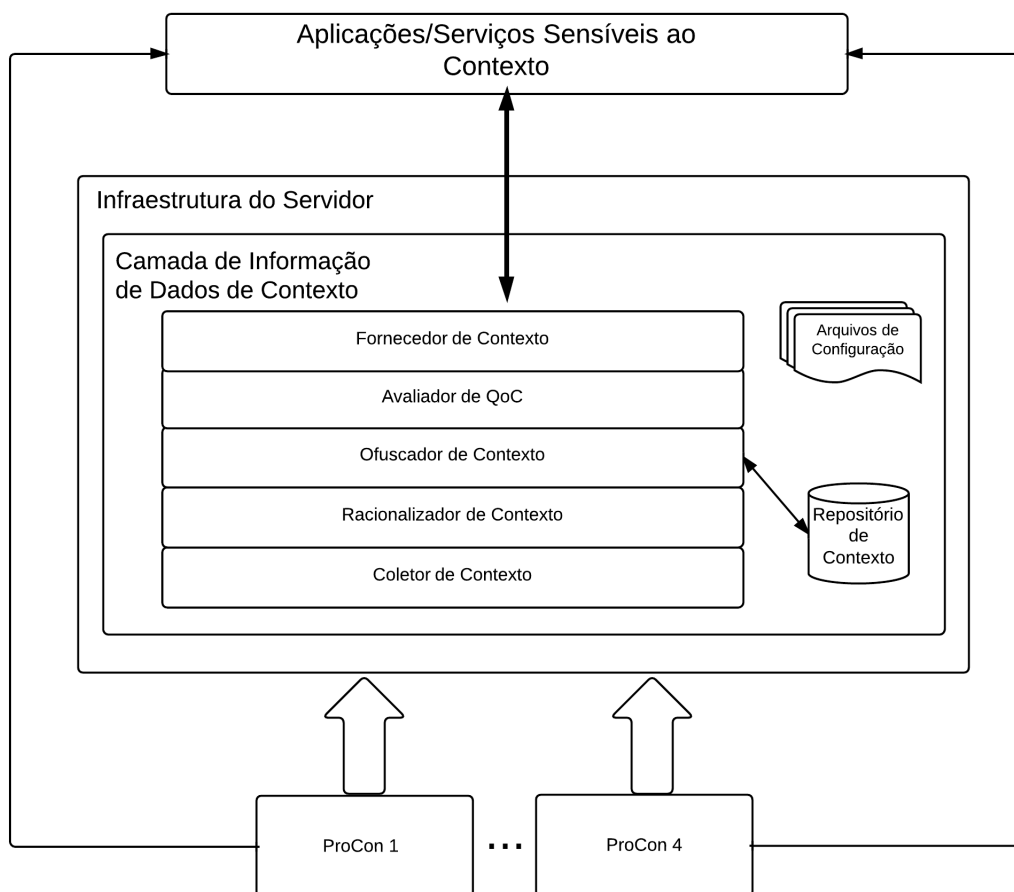


Figura 10: Arquitetura implementada para os experimentos

essa razão, esse cenário produz uma grande quantidade de dados redundantes e conflitantes (ex.: quando o veículo estaciona ou quando existe alguma interferência no sinal do satélite), caracterizando uma aplicação real para a abordagem proposta.

#### Listing 5.1: Método para Resolver Conflitos

```

void resolveConflito(Contexto co1, Contexto co2){
    pol[] = getPoliticadeConflito(co1);
    contextoValido = aplicaPolitica(co1, co2, pol[]);
    enviaParaCache(contextoValido);
}

```

Para remover redundâncias e conflitos foram implementados algoritmos baseados no trabalho de (MANZOOR; TRUONG; DUSTDAR, 2009), o pseudo-código é apresentado nos listings 5.1, 5.2 e 5.3. O método *Equals* compara todas as dimensões de contexto entre dois dados de contexto distintos, e verifica se esses dados representam informações redundantes. Já o método *Remove Redundâncias* compara o dado recentemente criado com todos os dados que estão na cache. Uma vez identificada a redundância, o algoritmo calcula a diferença de tempo entre dois objetos de contexto e adiciona ao tempo de parada  $t_s$  (SILVA; DANTAS, 2013a) do

dado válido. Após essa operação o dado redundante é eliminado. O método *Resolve conflito* é utilizado quando existe alguma indefinição na comparação entre dados de contexto. O método usa as políticas de qualidade, que caracterizam redundâncias e conflitos, para resolver inconsistências entre dados de contexto.

Listing 5.2: Método Equals

```
boolean equals(Contexto co1, Contexto co2){
    if(co1.getLocalizacaoGeografica().equals(co2.
        getLocalizacaoGeografica())){
        if(co1.getTodosContextos().equals(co2.getTodosContextos())
        ){
            return true;
        } else
            resolveConflito(co1, co2);
        return false;
    }
}
```

Foram simulados quatro dispositivos pervasivos enviando informações ininterruptamente para um servidor de contexto. Os provedores de contexto foram configurados para fornecer dados em intervalos de 1 segundo. Além da informações geográfica, os provedores de contexto também forneceram informações sobre os contexto de tempo e contexto computacional.

Listing 5.3: Método para Remover Redundâncias

```
void removeRedundancia(Contexto co){
    contextoNaCache = getCacheContexto();
    for i from 0 to tamanhoDaCache {
        if(cache[i].equals(co)){
            t = getDiferencaTemporal(cache[i], co);
            cache[i].adicionaTempoDeParada(t);
            deleta(co)
        }
        else
            enviaParaCache(co);
    }
}
```

As políticas adotadas para esse experimento foram: i) a precisão da informação do GPS deve estar presente no dado de contexto, ii) a precisão dos dados de GPS devem ser de no máximo 50 metros, iii) a diferença de distância entre duas informações de contexto consecutivas deve ser maior que 200 metros e iv) a diferença de tempo entre duas informações de contexto consecutivas deve ser maior que 10 segundos. Sendo que, i) e ii) são políticas orientadas à

parâmetros de qualidade; iii) e iv) são políticas orientadas à aplicação. O esquema do arquivo de políticas é apresentado no *listing 5.4*.

Listing 5.4: Representação XML de um arquivo de políticas de qualidade

```
<?xml version= "1.0"?>
<Politicass>
  <Politica>
    <NomeParametro>Latitude</NomeParametro>
    <NomeParametro>Longitude</NomeParametro>
    <ParametroDeQualidade>
      <Nome>Precisao</Nome>
      <E>
        <MenorQue>50</MenorQue>
        <Nulo>falso</Nulo>
      </E>
    </ParametroDeQualidade>
  </Politica>

  <Politica>
    <NomeParametro>DiferencaDistancia</NomeParametro>
    <MaiorQue>200</MaiorQue>
    <Nulo>falso</Nulo>
  </Politica>

  <Politica>
    <NomeParametro>DiferencaTempo</NomeParametro>
    <MaiorQue>10</MaiorQue>
  </Politica>
</QoCPolicies>
```

## 5.2 ANÁLISE DOS RESULTADOS

A figura 11 apresenta o número de dados de contexto enviados em 6 horas de simulação. Tanto as políticas orientadas à aplicação quanto as políticas orientadas à parâmetros de qualidade reduziram o número de dados de contexto transmitidos. A intenção foi apresentar a efetividade das políticas separadamente, dado que as mesmas não interferem umas nas outras.

Os experimentos mostraram uma redução de até 52% na quantidade de dados enviados aos consumidores de contexto, mostrando que implementação da proposta juntamente com a definição de políticas de qualidade podem reduzir significativamente a quantidade de dados enviados pelos dispositivos ubíquos móveis. Para um melhor entendimento da significância dos dados apresentados, a seguir são caracterizadas as condições (localização) aonde esses veículos estavam se locomovendo e são conduzidas as análises sobre os resultados.

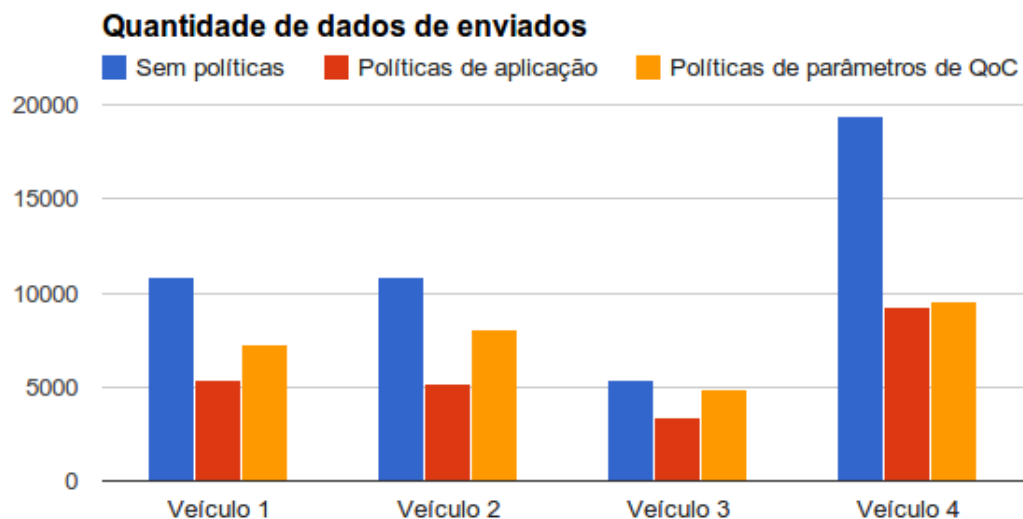


Figura 11: Quantidade de dados de contexto filtrados

Os dados coletados dos veículos 1 e 2 são provenientes da região de Machu Picchu no Peru. Nessa região os dados providos pelo GPS tendem a informar localizações divergentes da onde o GPS realmente se encontra. Isso se deve principalmente pelo relevo da região que causa interferências na triangulação dos satélites, aumentando a imprecisão nos dados captados. O que é interessante perceber, em relação a esses dois veículos, é que a utilização de políticas de aplicação resultou em uma quantidade maior de dados filtrados se comparada a aplicação das políticas de parâmetros de QoC, indicando que mesmo que qualidade de contexto esteja assegurada, o dado pode não ser interessante para a aplicação.

Os dados do veículo 3 foram coletados durante um treinamento de corrida de longa distância em Florianópolis, mais especificamente, na avenida beira-mar norte da cidade. Diferentemente da região de Machu Picchu, a avenida beira-mar norte está localizada em um centro urbano e com uma menor grau de interferências causado pelo relevo, assim, em relação aos outros veículos, uma porcentagem menor de dados foram filtrados. Mas, novamente, as políticas orientadas as aplicações tiveram uma boa efetividade na filtragem dos dados de contexto. Esse cenário pode ser transposto para aplicações de tempo de real (ex. monitoramento de veículos em tempo real), onde uma grande quantidade de informações são produzidas, e caso não sejam definidas políticas orientadas a aplicação, existem grandes chances de ocorrerem sobrecargas nos servidores de contexto devido a quantidade de dados que devem ser comunicados, processados e armazenados.

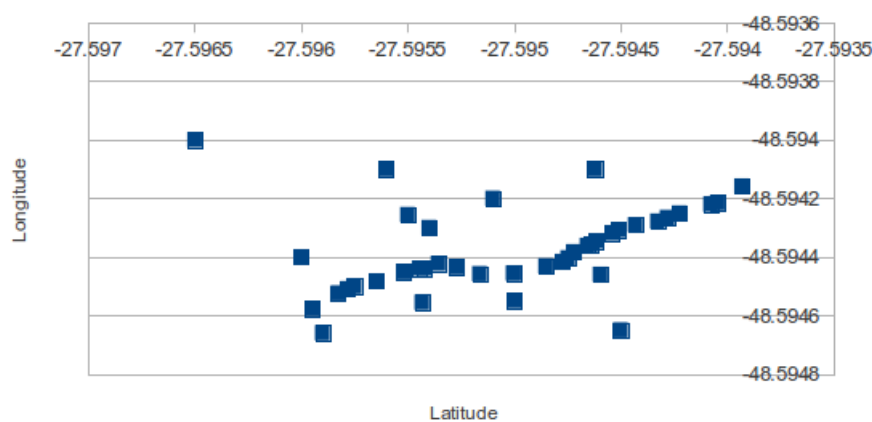
Já para o veículo 4 foi utilizada uma abordagem diferente. Os dados foram coletados durante uma semana na cidade de Florianópolis acompanhando o cotidiano de um indivíduo. Além disso, o usuário tinha a liberdade de randomicamente ligar e desligar o GPS enquanto se locomovia. Os dados foram coletados tanto em regiões que possuíam interferências quanto em regiões de maior precisão na cidade. Nesse caso, as políticas orientadas à parâmetros e orientadas a aplicação obtiveram uma taxa de filtragem muito parecida (aprox. 51%). Indicando que,



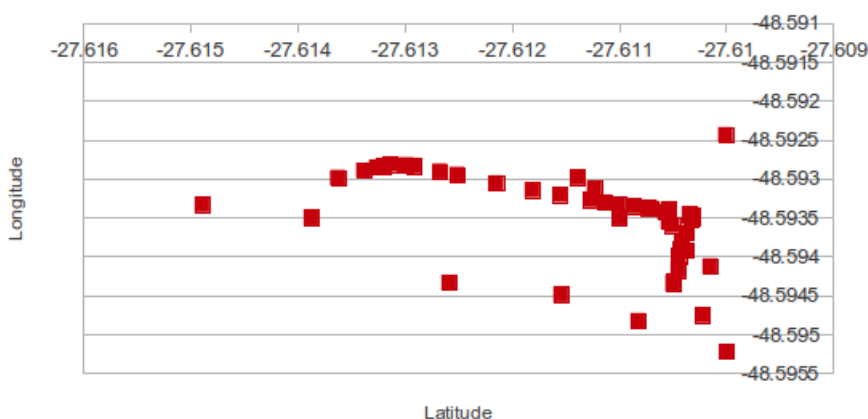
	Quantidade Total de Dados	Quantidade de Dados Filtrados (App)	Quantidade de Dados Filtrados (QoC)
Veículo 1	10800	5476 (50.7%)	3496 (32.37%)
Veículo 2	10800	5632 (52.14%)	2755 (25.09%)
Veículo 3	5400	1975 (36.57%)	515 (9.53%)
Veículo 4	19380	10115 (52.39%)	9816 (50.65%)

Tabela 6: Porcentagem de dados filtrados

de fato, as políticas de qualidade (orientadas a aplicações e orientadas a parâmetros de contexto) devem ser utilizadas conjuntamente para uma melhor garantia de qualidade de contexto nos dados providos por dispositivos ubíquos móveis.



(a) Trajetória não filtrada: veículo 1



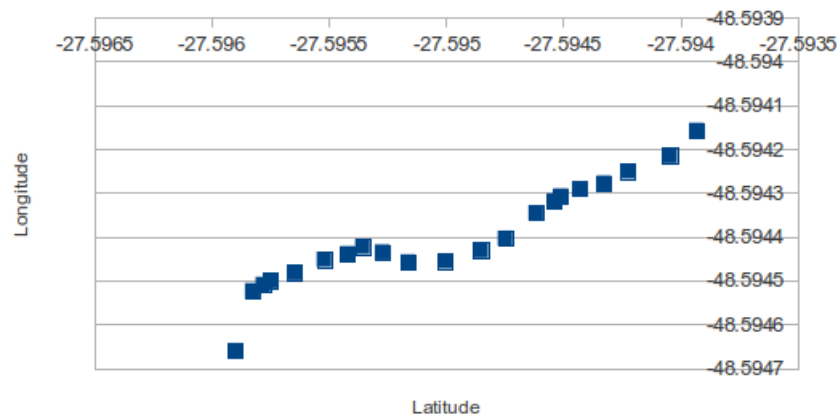
(b) Trajetória não filtrada: veículo 4

Figura 12: Trajetórias não filtradas

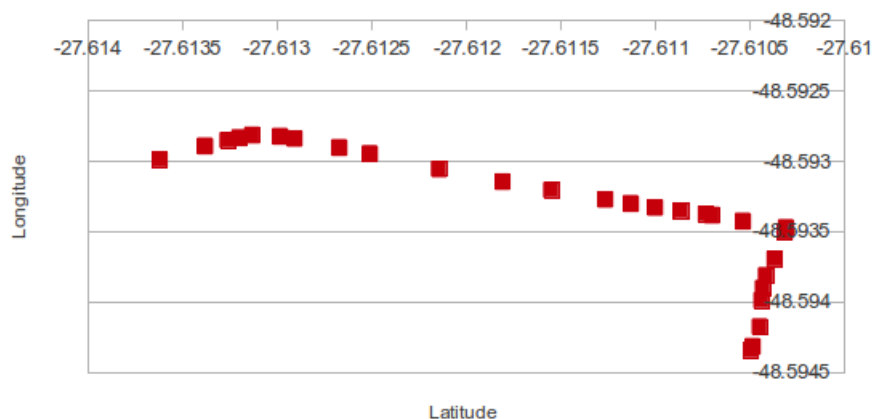
A figura 12 mostra trechos das trajetórias sem nenhum processo de filtragem de contexto. Dados não validados tendem a representar trajetórias que diferem da real trajetória feita pelo usuário. As duas figuras mostram partes das trajetórias que possuem situação de parada do veículo e perda de sinal. A figura 12a faz parte da trajetória do veículo 1, e a figura 12b compõe

a trajetória realizada pelo veículo 4. Pode se perceber que nesses casos, os dados não filtrados parecem não representar uma trajetória, ou parte dela.

A figura 13 apresenta as mesmas rotas da figura 12, entretanto, com dados validados. A quantidade de dados utilizados para representar as trajetórias reduziu em aproximadamente 40%. Os pontos fora da trajetória foram largamente eliminados, e a representação das trajetórias, com os dados validados, fornecem uma melhor referência da trajetória realizada pelo o usuário. Além disso, em consequência a redução de envio, a quantidade de dados processados pelos algoritmos de inferência diminuiu sensivelmente.



(a) Trajetória filtrada: veículo 1



(b) Trajetória filtrada: veículo 4

Figura 13: Trajetórias filtradas

Essa análise pontual dos trechos das trajetórias tem o intuito de verificar se o processo de filtragem proposto contribui de forma direta para os fins das aplicações consumidora dos dados de contexto. Assim, através desses resultados, pode-se inferir que ao enviar dados pré-validados, ou seja, que possuem a qualidade desejada, as aplicações e serviços sensíveis ao contexto tendem a melhor atingir seus objetivos. De maneira específica, os dados de contexto filtrados fornecidos pelos dispositivos pervasivos forneceram um conjunto de informações ca-

paz de representar de uma melhor forma as trajetórias feitas pelos usuários.

A fim de verificar o impacto causado no dispositivo ubíquo móvel pela implementação da abordagem proposta, foi verificada a sobrecarga de processamento durante o envio de dados para os consumidores de contexto. Assim, é comparado a carga de processamento do dispositivo enviando dados sem a abordagem proposta e com a abordagem proposta (ProQuali). A figura 14 apresenta a curva de processamento média (dos 4 dispositivos) durante o tempo da simulação. A abordagem ProQuali exigiu, em média, 26% a mais da capacidade de processamento do dispositivo.

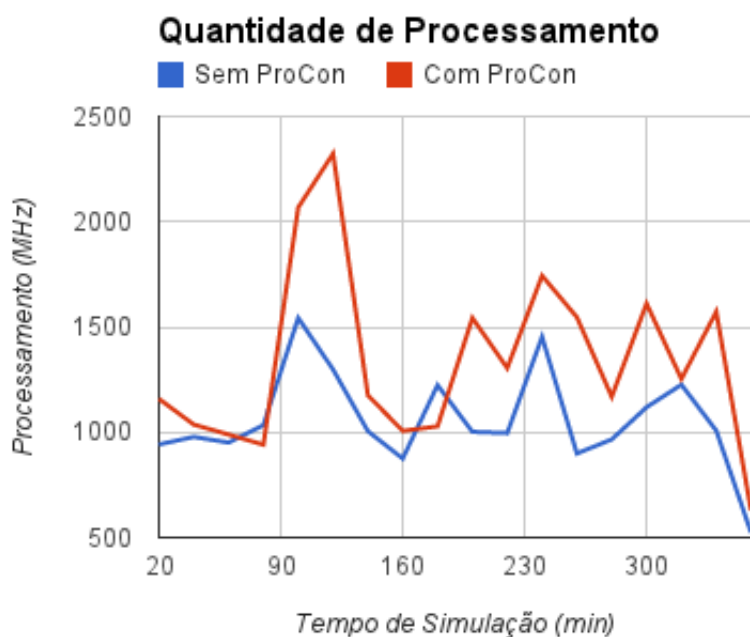


Figura 14: Sobrecarga pela utilização do ProQuali

O aumento na demanda de processamento ocasionado pela abordagem ProQuali contribui para otimização na utilização dos recursos computacionais disponíveis. Essa afirmação pode parecer de certa forma contraditória, mas de fato utilizar de maneira balanceada os recursos disponíveis em uma infraestrutura distribuída colabora para a otimização do sistema como um todo. De maneira mais detalhada, estamos tratando da otimização sob a perspectiva de utilização de recursos, sendo assim, tanto a sub-utilização quanto a sobrecarga de determinados componentes da infraestrutura podem caracterizar um uso não otimizados de recursos computacionais. Assim, através da utilização do ProQuali foi possível explorar os recursos disponíveis nos dispositivos pervasivos sem sobrecarregá-los. E, além disso, ao transferir parte da responsabilidade do processamento para os dispositivos, os consumidores (ex.: servidor de contexto) reduzem a carga de processamento seja pela redução de dados recebidos seja pelo pré-processamento realizado nos dados de contexto.

Dado que a comunicação sem fio é um dos principais fatores que influenciam no tempo de vida a bateria dos dispositivos ubíquos móveis, foi realizada uma medição para verificar se

a utilização do ProQuali impactava diretamente na eficiência energética do dispositivo. Apresentamos na figura 15 o consumo de bateria ocorrido durante toda a simulação. A abordagem ProQuali aumentou o tempo de vida da bateria em aproximadamente 28%.

O consumo de bateria é um fator determinante quando tratamos de dispositivos móveis. E apesar do aumento na quantidade de processamento, o consumo de bateria foi menor. Como a comunicação sem fio é uma das funcionalidades que mais consomem recursos energéticos dos dispositivos, e dado que a implementação da abordagem reduziu a quantidade de dados transmitidos em até 50%, foi obtida uma melhor performance energética nos dispositivos ubíquos móveis.

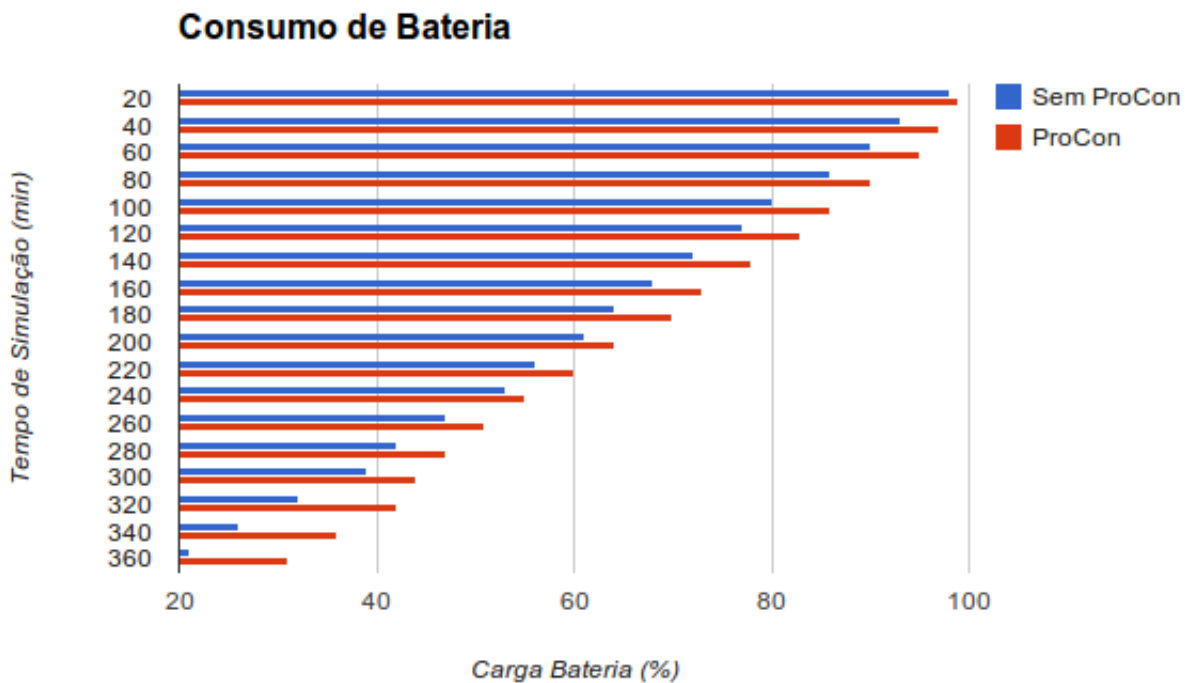


Figura 15: Consumo de bateria do dispositivo

Essa combinação de resultados provê suporte para a premissa conceitual que a qualidade de contexto só pode ser totalmente atingida se os requisitos de cada aplicação forem projetados no modelo de contexto, e os dispositivos móveis forem explorados para assegurar uma melhor qualidade nos dados processados. O comportamento geral dos experimentos mostra que a abordagem reduz o número de dados transmitidos e também aperfeiçoa a representação das trajetórias. Também pôde ser observado que as políticas definidas em termos da aplicação tendem a melhor atender requisitos dos serviços sensíveis ao contexto, diminuindo a quantidade de processamento desnecessário realizado pela infraestrutura computacional.

### 5.3 CONSIDERAÇÕES SOBRE O CAPÍTULO

Esse capítulo apresenta o ambiente experimental e a análise dos resultados gerados durante a execução do protótipo simulado. Quanto ao ambiente experimental, foi utilizada uma arquitetura distribuída onde os dispositivos pervasivos enviam dados de contexto para um servidor de contexto. Os dispositivos contem uma implementação do ProQuali, que explora as políticas de qualidade definidas no formato XML. Já o servidor de contexto possui camadas que são responsáveis por captar os dados enviados pelos dispositivos e agregar os dados que representam uma mesma trajetória. Vale ressaltar que o foco dessa dissertação está voltado para qualidade dos dados enviados pelos provedores de contexto, assim, as camadas do servidor de contexto foram implementadas para realizar apenas operações simples de agregação de dados. Dessa forma, fica mais destacada a efetividade da abordagem ProQuali no provimento de dados com qualidade de contexto.

Em relação aos resultados obtidos, pode ser verificado um melhor aproveitamento dos recursos disponíveis no dispositivo pervasivo. O tempo de vida da bateria aumentou em aproximadamente 28%, o que é justificado, principalmente, pela redução de dados comunicados pelos dispositivos. Essa redução é consequência do processo de filtragem realizado. Tal processo, exige um maior poder de processamento dos dispositivos, porém sem sobrecarregá-los. Assim, o processamento do sistema como um todo encontra um equilíbrio, isto é, explora toda infraestrutura computacional sem sobrecarregar e nem subutilizar seus componentes. Ademais, a redução de dados enviados também contribui para o aumento na escalabilidade do servidor de contexto. Através desses resultados, pode-se verificar que a abordagem ProQuali possui características favoráveis no que diz respeito ao provimento de dados de contexto para aplicações e serviços sensíveis ao contexto.



## 6 CONCLUSÕES, LIMITAÇÕES E TRABALHOS FUTUROS

### 6.1 CONCLUSÕES

O trabalho de pesquisa desenvolvido nessa dissertação propõe uma abordagem de filtragem consciente de qualidade de contexto para ambientes ubíquos móveis (ProQuali). ProQuali baseou-se no fato de que dispositivos pervasivos são partes essenciais para produção de dados de contexto e, por conseguinte, também devem ser responsáveis por inserir a qualidade de contexto nos dados. ProQuali é composto por componentes que tratam das principais fases relativas ao provimento de dados de contexto com qualidade. Mais especificamente, ProQuali coleta as informações de contexto enviadas pelos sensores, padroniza essas informações em dados interoperáveis, injeta parâmetros de qualidade, avalia a qualidade dos dados de contexto, e dissemina o dado aos consumidores. Além disso, ProQuali prevê o armazenamento dos dados em tempo de execução com a finalidade de minimizar o processamento e envio de dados redundantes, e garantir a disponibilidade do dado caso existam problemas de conectividade no momento do envio.

A proposta também define um processo de filtragem de dados de contexto que tem como principal objetivo eliminar os dados que não possuem a qualidade desejada pelas aplicações e serviços sensíveis ao contexto. O processo usufrui dos componentes modelados no dispositivo e realiza a filtragem em todas as informações de contexto recém captadas, eliminando informações de contexto sem qualidade, redundantes e conflitantes, antes que os dados sejam enviados para os consumidores. O processo se baseia em políticas de qualidade que tem como principal objetivo delimitar o escopo de interesse das aplicações e indicar os fatores que influenciam na qualidade das informações solicitadas pelos diversos serviços sensíveis ao contexto. Nesse sentido, a proposta indica que a qualidade das informações de contexto não é totalmente garantida apenas pela avaliação de parâmetros de QoC, e propõe que os requisitos das aplicações sejam levados em consideração para que haja uma efetiva garantia de qualidade de contexto.

De maneira mais ampla, o trabalho sugere que parte do *middleware* de contexto deva estar presente nos dispositivos ubíquos móveis que compõe a infraestrutura computacional sensível ao contexto. Garantindo, assim, que o *middleware* de contexto, que por definição é responsável por suprir as aplicações com dados de contexto qualificados, tenha controle de todo o processo de provimento dos dados, da captura até o envio as aplicações e serviços. Além disso, a proposta também argumenta que o *middleware* não deve ser “cego” em relação as aplicações e serviços que são por ele suportadas, isto é, o entendimento, por parte do *middleware*, de quais informações que são requeridas pelas aplicações, tende a melhorar a qualidade dos dados pro-

vidos aos seus consumidores.

Em relação aos objetivos definidos, ProQuali obteve sucesso em garantir a qualidade de contexto nas informações providas por dispositivos ubíquos. As políticas de qualidade definidas em termos de parâmetros de QoC e requisitos das aplicações forneceram subsídios para o processo de filtragem eliminar dados não desejáveis. Com a redução dos dados de contexto, a escalabilidade do sistema aumentou significativamente melhorando, assim, a distribuição dos dados dentro do sistema. Além disso, ao comunicar menos dados, os dispositivos obtiveram uma melhor performance energética, permitindo que outros aspectos do dispositivo, como processamento e armazenamento, fossem melhor explorados. Dessa forma, a abordagem melhorou qualidade de contexto dos dados transmitidos e otimizou a utilização dos recursos computacionais presentes nos dispositivos ubíquos móveis.

Os resultados experimentais, indicaram que a quantidade de dados enviados pelos dispositivos reduziu significativamente, implicando em melhorias na exploração dos recursos do dispositivo ubíquo e nos objetivos da aplicação em questão. Mais especificamente, o tempo de vida da bateria apresentou uma melhora em torno de 28% e as trajetórias foram melhor representadas utilizando-se a abordagem proposta. De fato, foi identificado uma sobrecarga no processamento dos dados de contexto, porém esse aumento era esperado dado as novas operações e funções inseridas no dispositivo ubíquo. Assim, voltando à hipótese colocada no início deste estudo, podemos afirmar que a proposta possui um potencial para uma melhor garantia de qualidade de contexto em ambiente ubíquos.

## 6.2 LIMITAÇÕES

Nessa seção apresentamos as limitações em relação aos experimentos realizados e descritos nessa dissertação. A primeira limitação é consequência direta da estratégia de pesquisa utilizada, qual seja, estudo de caso. Dessa forma, os experimentos não contemplaram diferentes arquiteturas distribuídas onde a abordagem ProQuali poderia também ser utilizada. De fato, dispositivos ubíquos móveis podem ser utilizados para compor diferentes infraestruturas computacionais, dado a facilidade provida pela comunicação sem fio de disseminar informações com uma menor limitação geográfica. A figura 16 ilustra uma arquitetura distribuída onde os nós se comunicam diretamente, sem a utilização de um centralizador.

A abordagem apresentada na figura 16 é interessante para aplicações e/ou serviços sensíveis ao contexto que são executados no próprio dispositivo. Assim, o *middleware* de contexto seria distribuído entre os nós executores, onde cada nó é produtor e consumidor de dados de contexto. Esse tipo de arquitetura possui um potencial de utilização a medida que os dispositivos ubíquos móveis crescem em poder de processamento e armazenamento, constituindo, assim, um cenário de aplicação para a abordagem ProQuali.



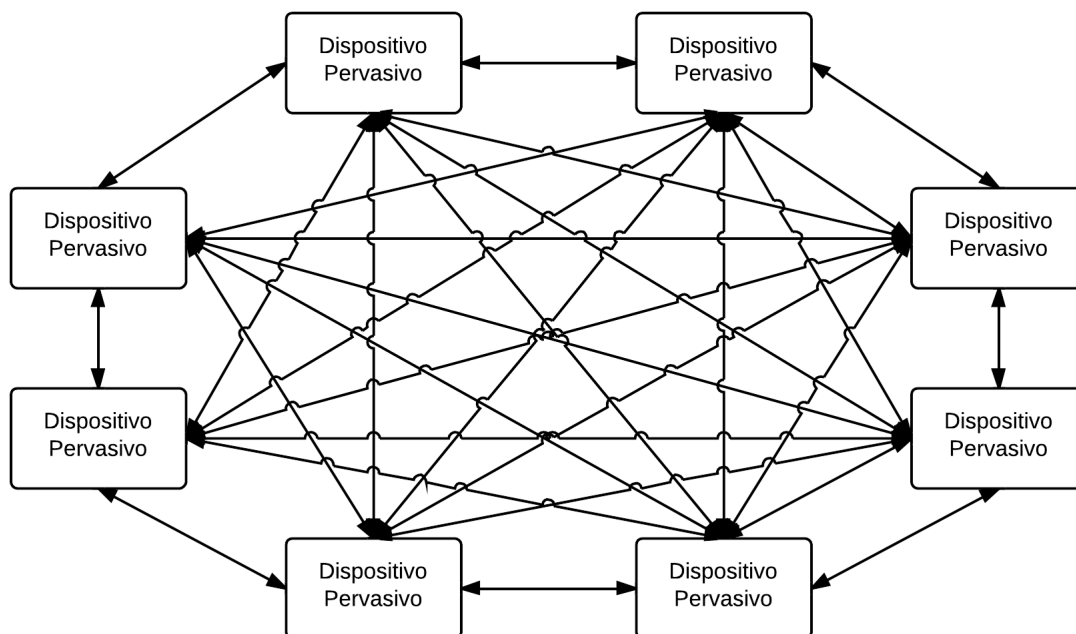


Figura 16: Exemplo de arquitetura descentralizada para aplicações sensíveis ao contexto

Outro esquema de arquitetura que poderia explorar a abordagem é apresentado na figura 17. Nesse caso, existe a figura de nós centralizadores porém os dispositivos ubíquos também se comunicam entre si. Dessa forma, os nós ubíquos podem enviar informações diretamente para as aplicações, e as demandas por operações mais robustas sob os dados de contexto (ex.: agregação, inferência, ...) ficam a cargo dos nós centralizadores de informação.

Assim sendo, diferentes arquiteturas distribuídas podem ser concebidas para configurar infraestruturas computacionais que suportem aplicações sensíveis ao contexto. E apesar de o trabalho de pesquisa não contemplar experimentos em diferentes esquemas computacionais, a validade dos resultados permanece, já que um estudo de natureza exploratória objetiva familiarizar-se com os fenômenos que estão sendo investigados, de modo que as pesquisas subsequentes possam ser concebidas com uma maior compreensão e precisão. De maneira mais direta, ProQuali estabelece as principais fases no provimento de dados de contexto com qualidade, fornecendo meios para a concepção de arquiteturas mais complexas e com diferentes focos de aplicação

Outra limitação dessa dissertação se refere ao nicho de aplicações utilizado nos experimentos. O estudo de caso tratou de uma aplicação que recebia informações de contexto e buscava descobrir padrões de comportamento nas trajetórias dos usuários. A primeira dificuldade consistia em identificar a rota realizada pelo usuário que por muitas vezes era mal representada dado a falta de qualidade nos dados providos pelos dispositivos. Essa aplicação constitui um cenário de aplicação da abordagem, porém, uma gama de aplicações sensíveis ao contexto podem ser concebidas utilizando ProQuali. Porém, esse trabalho não buscava o desenvolvi-

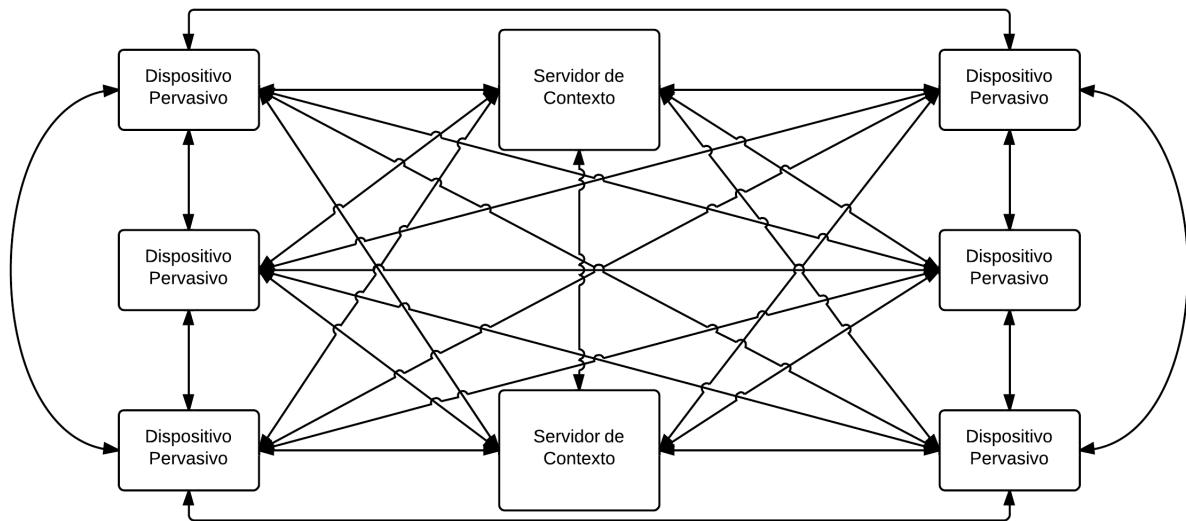


Figura 17: Exemplo de arquitetura para aplicações sensíveis ao contexto

mento de aplicações sensíveis ao contexto, e sim fornecer uma melhor infraestrutura para que as aplicações pudessem atingir seus objetivos.

### 6.3 TRABALHOS FUTUROS

Tendo posto as conclusões e limitações desse trabalho, pode-se enumerar alguns caminhos de pesquisa que poderiam favorecer o aperfeiçoamento da proposta exposta nessa dissertação, explicitados como:

- Investigar os aspectos semânticos que envolvem dados de contexto com o intuito de definir políticas de qualidade mais robustas, e além disso constituir abordagens para a descoberta de valores de parâmetros de qualidade que estejam cientes das limitações encontradas em dispositivos ubíquos móveis.
- Desenvolver modelos para representação de informações de contexto com uma maior granularidade, com o intuito de melhorar os procedimentos de exportação das informações de contexto nos dispositivos perversivos.
- Conduzir testes experimentais em arquiteturas diferentes, com um maior número de aplicações e em ambientes mais heterogêneos, com o objetivo de identificar possíveis melhorias e limitações da modelagem proposta.
- Realizar estudos mais profundos quanto a performance das implementações da abordagem, verificando a sobrecarga adicionada nas diferentes etapas da filtragem dos dados dados de contexto, além de análises sobre a performance energética do dispositivo durante a transmissão dos dados de contexto em diferentes arquiteturas computacionais.

- Projetar interfaces naturais, para que os usuários possam definir políticas de qualidade de maneira mais intuitiva, fornecendo informações de interesse para as aplicações.



## REFERÊNCIAS BIBLIOGRÁFICAS

10746.1-4, I. *Open Distributed Processing - Reference Model*. 1995–1998.

ABID, Z.; CHABRIDON, S. A fine-grain approach for evaluating the quality of context. In: IEEE. *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*. [S.l.], 2011. p. 444–449.

ABOWD, G. D. et al. Towards a better understanding of context and context-awareness. In: SPRINGER. *Handheld and ubiquitous computing*. [S.l.], 1999. p. 304–307.

AKYILDIZ, I. F. et al. A survey on sensor networks. *Communications magazine, IEEE, IEEE*, v. 40, n. 8, p. 102–114, 2002.

BALASUBRAMANIAN, N.; BALASUBRAMANIAN, A.; VENKATARAMANI, A. Energy consumption in mobile phones: a measurement study and implications for network applications. In: ACM. *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. [S.l.], 2009. p. 280–293.

BALDAUF, M.; DUSTDAR, S.; ROSENBERG, F. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, Inderscience, v. 2, n. 4, p. 263–277, 2007.

BARTOLINI, A.; RUGGIERO, M.; BENINI, L. Visual quality analysis for dynamic backlight scaling in lcd systems. In: EUROPEAN DESIGN AND AUTOMATION ASSOCIATION. *Proceedings of the Conference on Design, Automation and Test in Europe*. [S.l.], 2009. p. 1428–1433.

BELLAVISTA, P. et al. A survey of context data distribution for mobile ubiquitous systems. *ACM Computing Surveys*, v. 45, n. 1, p. 1–49, 2013.

BELLAVISTA, P.; CORRADI, A.; MAGISTRETTI, E. Redman: An optimistic replication middleware for read-only resources in dense manets. *Pervasive and Mobile Computing*, Elsevier, v. 1, n. 3, p. 279–310, 2005.

BELLAVISTA, P. et al. Context-aware middleware for resource management in the wireless internet. *Software Engineering, IEEE Transactions on*, IEEE, v. 29, n. 12, p. 1086–1099, 2003.

BERNAL, J. F. M. et al. Towards an efficient context-aware system: problems and suggestions to reduce energy consumption in mobile devices. In: IEEE. *Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR), 2010 Ninth International Conference on*. [S.l.], 2010. p. 510–514.

BETTINI, C. et al. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, Elsevier, v. 6, n. 2, p. 161–180, 2010.

BHARATHIDASAN, A.; PONDURU, V. A. S. Sensor networks: An overview. *paper, University of California, Davis, CA (August 2002)*, 2002.

BOLCHINI, C. et al. Context information for knowledge reshaping. *International Journal of Web Engineering and Technology*, Inderscience, v. 5, n. 1, p. 88–103, 2009.

- BOLDRINI, C.; CONTI, M.; PASSARELLA, A. Exploiting users social relations to forward data in opportunistic networks: The hibop solution. *Pervasive and Mobile Computing*, Elsevier, v. 4, n. 5, p. 633–657, 2008.
- BRGULJA, N. et al. Measuring the probability of correctness of contextual information in context aware systems. In: IEEE. *Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on*. [S.l.], 2009. p. 246–253.
- BROWN, P. J.; BOVEY, J. D.; CHEN, X. Context-aware applications: from the laboratory to the marketplace. *Personal Communications, IEEE*, IEEE, v. 4, n. 5, p. 58–64, 1997.
- BUCHHOLZ, T.; KÜPPER, A.; SCHIFFERS, M. Quality of context: What it is and why we need it. In: *Proceedings of the workshop of the HP OpenView University Association*. [S.l.: s.n.], 2003. v. 2003.
- CAPRA, L.; EMMERICH, W.; MASCOLO, C. Carisma: Context-aware reflective middleware system for mobile applications. *Software Engineering, IEEE Transactions on*, IEEE, v. 29, n. 10, p. 929–945, 2003.
- CARTIGNY, J.; SIMPLOT, D. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In: IEEE. *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*. [S.l.], 2003. p. 10–pp.
- CERI, S. et al. Model-driven development of context-aware web applications. *ACM Transactions on Internet Technology (TOIT)*, ACM, v. 7, n. 1, p. 2, 2007.
- CHANG, H.; SHIN, S.; CHUNG, C. Context life cycle management scheme in ubiquitous computing environments. In: IEEE. *Mobile Data Management, 2007 International Conference on*. [S.l.], 2007. p. 315–319.
- CHEN, G.; KOTZ, D. et al. *A survey of context-aware mobile computing research*. [S.l.], 2000.
- CHEN, G.; LI, M.; KOTZ, D. Data-centric middleware for context-aware pervasive computing. *Pervasive and mobile computing*, Elsevier, v. 4, n. 2, p. 216–253, 2008.
- CHEN, H.; FININ, T.; JOSHI, A. An intelligent broker for context-aware systems. In: *Adjunct proceedings of Ubicomp*. [S.l.: s.n.], 2003. v. 3, p. 183–184.
- CHETAN, S. et al. Mobile gaia: a middleware for ad-hoc pervasive computing. In: IEEE. *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*. [S.l.], 2005. p. 223–228.
- CHO, K. et al. Hicon: a hierarchical context monitoring and composition framework for next-generation context-aware services. *Network, IEEE*, IEEE, v. 22, n. 4, p. 34–42, 2008.
- CONAN, D.; ROUYOY, R.; SEINTURIER, L. Scalable processing of context information with cosmos. In: SPRINGER. *Distributed Applications and Interoperable Systems*. [S.l.], 2007. p. 210–224.
- CORRADI, A.; FANELLI, M.; FOSCHINI, L. towards adaptive and scalable context aware middleware. *International Journal of Adaptive, Resilient and Autonomic Systems (IJARAS)*, IGI Global, v. 1, n. 1, p. 58–74, 2010.
- COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. *Sistemas distribuídos: conceitos e projeto*. [S.l.]: Grupo A, 2007.

- CULLER, D. E. et al. A network-centric approach to embedded software for tiny devices. In: SPRINGER. *Embedded Software*. [S.l.], 2001. p. 114–130.
- DEBATY, P.; GODDI, P.; VORBAU, A. Integrating the physical world with the web to enable context-enhanced mobile services. *Mobile Networks and Applications*, Kluwer Academic Publishers, v. 10, n. 4, p. 385–394, 2005.
- DEY, A. *Providing architectural support for building context-aware applications*. Tese (Doutorado) — Georgia Institute of Technology, 2000.
- DEY, A. et al. Distributed mediation of ambiguous context in aware environments. In: ACM. *Proceedings of the 15th annual ACM symposium on User interface software and technology*. [S.l.], 2002. p. 121–130.
- DEY, A. K. Context-aware computing: The cyberdesk project. In: *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*. [S.l.: s.n.], 1998. p. 51–54.
- DEY, A. K.; ABOWD, G. D. The context toolkit: Aiding the development of context-aware applications. In: *Workshop on Software Engineering for wearable and pervasive computing*. [S.l.: s.n.], 2000. p. 431–441.
- DRABKIN, V. et al. Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks. In: IEEE. *Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on*. [S.l.], 2007. p. 13–22.
- DÜRR, F. et al. Reference model for the quality of context information. 2010.
- ESTRIN, D. et al. Next century challenges: Scalable coordination in sensor networks. In: ACM. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. [S.l.], 1999. p. 263–270.
- EUGSTER, P.; GARBINATO, B.; HOLZER, A. Design and implementation of the pervaho middleware for mobile context-aware applications. In: IEEE. *e-Technologies, 2008 International MCETECH Conference on*. [S.l.], 2008. p. 125–135.
- EUGSTER, P. T.; GARBINATO, B.; HOLZER, A. Middleware support for context-aware applications. In: *Middleware for Network Eccentric and Mobile Applications*. [S.l.]: Springer, 2009. p. 305–322.
- FAHY, P.; CLARKE, S. Cass—a middleware for mobile context-aware applications. In: CITESEER. *Workshop on Context Awareness, MobiSys*. [S.l.], 2004.
- FORMAN, G. H.; ZAHORJAN, J. The challenges of mobile computing. *Computer*, IEEE, v. 27, n. 4, p. 38–47, 1994.
- FRIEDMAN, R. et al. Gossiping on manets: the beauty and the beast. *ACM SIGOPS Operating Systems Review*, ACM, v. 41, n. 5, p. 67–74, 2007.
- GIAFFREDA, R.; BARRIA, J. Service delivery in collaborative context-aware environments using fuzzy logic. In: IEEE. *Communications, 2007. ICC'07. IEEE International Conference on*. [S.l.], 2007. p. 2045–2049.
- GRADINESCU, V. et al. Adaptive traffic lights using car-to-car communication. In: IEEE. *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*. [S.l.], 2007. p. 21–25.

- GROSSMANN, M. et al. An abstract processing model for the quality of context data. In: *Quality of Context*. [S.l.]: Springer, 2009. p. 132–143.
- GU, T.; PUNG, H. K.; ZHANG, D. Q. A service-oriented middleware for building context-aware services. *Journal of Network and computer applications*, Elsevier, v. 28, n. 1, p. 1–18, 2005.
- GUPTA, A. et al. Mobisoc: a middleware for mobile social computing applications. *Mobile Networks and Applications*, Kluwer Academic Publishers, v. 14, n. 1, p. 35–52, 2009.
- GUPTA, A. et al. Automatic identification of informal social groups and places for geo-social recommendations. *International Journal of Mobile Network Design and Innovation*, Inderscience, v. 2, n. 3, p. 159–171, 2007.
- HAAS, Z. J.; HALPERN, J. Y.; LI, L. Gossip-based ad hoc routing. *IEEE/ACM Transactions on Networking (ToN)*, IEEE Press, v. 14, n. 3, p. 479–491, 2006.
- HENRICKSEN, K.; INDULSKA, J. Modelling and using imperfect context information. In: IEEE. *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*. [S.l.], 2004. p. 33–37.
- HENRICKSEN, K. et al. Middleware for distributed context-aware systems. In: *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*. [S.l.]: Springer, 2005. p. 846–863.
- HOFER, T. et al. Context-awareness on mobile devices-the hydrogen approach. In: IEEE. *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*. [S.l.], 2003. p. 10–pp.
- HOSSAIN, M. A. et al. Adaptive interaction support in ambient-aware environments based on quality of context information. *Multimedia Tools and Applications*, Springer, p. 1–24, 2012.
- IFTODE, L. et al. Active highways (position paper). In: IEEE. *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*. [S.l.], 2008. p. 1–5.
- JULIEN, C.; ROMAN, G.-C. Egospaces: Facilitating rapid development of context-aware mobile applications. *Software Engineering, IEEE Transactions on*, IEEE, v. 32, n. 5, p. 281–298, 2006.
- JUSZCZYK, L. et al. Adaptive query routing on distributed context-the cosine framework. In: IEEE. *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*. [S.l.], 2009. p. 588–593.
- KAKOUSIS, K.; PASPALLIS, N.; PAPADOPOULOS, G. A. A survey of software adaptation in mobile and ubiquitous computing. *Enterp. Inf. Syst.*, Taylor & Francis, Inc., Bristol, PA, USA, v. 4, n. 4, p. 355–389, nov. 2010. ISSN 1751-7575. <<http://dx.doi.org/10.1080/17517575.2010.509814>>.
- KIM, S.; LEE, H. The impact of organizational context and information technology on employee knowledge-sharing capabilities. *Public Administration Review*, Wiley Online Library, v. 66, n. 3, p. 370–385, 2006.
- KIM, Y.; LEE, K. A quality measurement method of context information in ubiquitous environments. In: IEEE. *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*. [S.l.], 2006. v. 2, p. 576–581.



- KJÆR, K. E. A survey of context-aware middleware. In: ACTA PRESS. *Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering*. [S.l.], 2007. p. 148–155.
- KLEIN, N.; DAVID, K. Time locality: A novel parameter for quality of context. In: IEEE. *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*. [S.l.], 2010. p. 277–280.
- KNAPPMEYER, M. et al. A context provisioning framework to support pervasive and ubiquitous applications. In: *Smart Sensing and Context*. [S.l.]: Springer, 2009. p. 93–106.
- KRANENBURG, H. V. et al. A context management framework for supporting context-aware distributed applications. *Communications Magazine, IEEE*, IEEE, v. 44, n. 8, p. 67–74, 2006.
- KRAUSE, M.; HOCHSTATTER, I. Challenges in modelling and using quality of context (qoc). In: *Proceedings of the Second international conference on Mobility Aware Technologies and Applications*. Berlin, Heidelberg: Springer-Verlag, 2005. (MATA'05), p. 324–333. ISBN 3-540-29410-4, 978-3-540-29410-8.
- LEE, U. et al. Bio-inspired multi-agent data harvesting in a proactive urban monitoring environment. *Ad Hoc Networks*, Elsevier, v. 7, n. 4, p. 725–741, 2009.
- LESTER, J.; HANNAFORD, B.; BORRIELLO, G. "are you with me?--using accelerometers to determine if two devices are carried by the same person. In: *Pervasive computing*. [S.l.]: Springer, 2004. p. 33–50.
- LEVIS, P. et al. Tinyos: An operating system for sensor networks. In: *Ambient intelligence*. [S.l.]: Springer, 2005. p. 115–148.
- LI, F.; SEHIC, S.; DUSTDAR, S. Copal: An adaptive approach to context provisioning. In: IEEE. *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*. [S.l.], 2010. p. 286–293.
- MACEDO, D. et al. A distributed information repository for autonomic context-aware manets. *Network and Service Management, IEEE Transactions on*, IEEE, v. 6, n. 1, p. 45–55, 2009.
- MANZOOR, A.; TRUONG, H.; DUSTDAR, S. On the evaluation of quality of context. *Smart Sensing and Context*, Springer, p. 140–153, 2008.
- MANZOOR, A.; TRUONG, H.; DUSTDAR, S. Quality aware context information aggregation system for pervasive environments. p. 266–271, 2009.
- MANZOOR, A.; TRUONG, H.-L.; DUSTDAR, S. Quality of context: Models and applications for context-aware systems in pervasive environments. *The Knowledge Engineering Review, Special Issue on Web and Mobile Information Services*, 2011.
- MARTIN, H. et al. Qacbac: an owner-centric qoc-aware context-based access control model for pervasive environments. In: ACM. *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*. [S.l.], 2008. p. 30–38.
- MASHHADI, A. J.; MOKHTAR, S. B.; CAPRA, L. Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks. In: IEEE. *World of Wireless, Mobile and Multimedia Networks & Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a*. [S.l.], 2009. p. 1–6.

- MIRANDA, H. et al. An algorithm for dissemination and retrieval of information in wireless ad hoc networks. In: *Euro-Par 2007 Parallel Processing*. [S.l.]: Springer, 2007. p. 891–900.
- MIRON, A. et al. Modeling and measuring quality of context information in pervasive environments. In: IEEE. *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. [S.l.], 2010. p. 690–697.
- MURMURIA, R. et al. Mobile application and device power usage measurements. In: IEEE. *Software Security and Reliability (SERE), 2012 IEEE Sixth International Conference on*. [S.l.], 2012. p. 147–156.
- MUSOLESI, M.; MASCOLO, C. Car: context-aware adaptive routing for delay-tolerant mobile networks. *Mobile Computing, IEEE Transactions on*, IEEE, v. 8, n. 2, p. 246–260, 2009.
- NAZARIO, D.; DANTAS, M.; TODESCO, J. Taxonomia das publicações sobre qualidade de contexto. *SBIJOURNAL*, n. 20, 2012.
- NEISSE, R.; WEGDAM, M.; SINDEREN, M. V. Trustworthiness and quality of context information. p. 1925–1931, 2008.
- PITOURA, E.; SAMARAS, G.; SAMARAS, G. *Data management for mobile computing*. [S.l.]: Kluwer Boston, 1998.
- POWERS, S. *Practical rdf*. [S.l.]: O'Reilly, 2009.
- RAGHUNATHAN, V. et al. Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, IEEE, v. 19, n. 2, p. 40–50, 2002.
- RANGANATHAN, A. et al. Middleware: a middleware for location awareness in ubiquitous computing applications. In: SPRINGER-VERLAG NEW YORK, INC. *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. [S.l.], 2004. p. 397–416.
- RANGANATHAN, A.; CAMPBELL, R. H. A middleware for context-aware agents in ubiquitous computing environments. In: SPRINGER. *Middleware 2003*. [S.l.], 2003. p. 143–161.
- RICHARDS, M.; MONSON-HAEFEL, R.; CHAPPELL, D. A. *Java message service*. [S.l.]: O'Reilly Media, Inc., 2009.
- RIVA, O. et al. Context-aware migratory services in ad hoc networks. *Mobile Computing, IEEE Transactions on*, IEEE, v. 6, n. 12, p. 1313–1328, 2007.
- RYAN, N. S.; PASCOE, J.; MORSE, D. R. Enhanced reality fieldwork: the context-aware archaeological assistant. In: TEMPUS REPARATUM. *Computer applications in archaeology*. [S.l.], 1998.
- SACRAMENTO, V. et al. Moca: A middleware for developing collaborative applications for mobile users. *Distributed Systems Online, IEEE*, IEEE, v. 5, n. 10, p. 2–2, 2004.
- SASSON, Y.; CAVIN, D.; SCHIPER, A. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In: IEEE. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*. [S.l.], 2003. v. 2, p. 1124–1130.
- SATYANARAYANAN, M. Pervasive computing: Vision and challenges. *Personal Communications, IEEE*, IEEE, v. 8, n. 4, p. 10–17, 2001.

- SCHILIT, B.; ADAMS, N.; WANT, R. Context-aware computing applications. In: IEEE. *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on.* [S.l.], 1994. p. 85–90.
- SHEIKH, K.; WEGDAM, M.; SINDEREN, M. van. Middleware support for quality of context in pervasive context-aware systems. In: IEEE. *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on.* [S.l.], 2007. p. 461–466.
- SILVA, C.; DANTAS, M. A context-aware approach on semantic trajectories. In: *25th International Conference on Software Engineering and Knowledge Engineering.* [S.l.: s.n.], 2013.
- SILVA, C.; DANTAS, M. Quality-aware context provider: A filtering approach to context-aware systems on ubiquitous environment. In: *9th International Conference on Wireless and Mobile Computing, Networking and Communications.* [S.l.: s.n.], 2013.
- SØRENSEN, C.-F. et al. A context-aware middleware for applications in mobile ad hoc environments. In: ACM. *Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing.* [S.l.], 2004. p. 107–110.
- SOUSA, J. P.; GARLAN, D. Aura: an architectural framework for user mobility in ubiquitous computing environments. Kluwer, 2002.
- SRISATHAPORNPHAT, C.; JAIKAEAO, C.; SHEN, C.-C. Sensor information networking architecture. In: IEEE. *Parallel Processing, 2000. Proceedings. 2000 International Workshops on.* [S.l.], 2000. p. 23–30.
- STRANG, T.; LINNHOFF-POPIEN, C. A context modeling survey. In: *Workshop Proceedings.* [S.l.: s.n.], 2004.
- SWEITZER, J. et al. *Common Information Model: Implementing the Object Model for Enterprise Management.* [S.l.]: John Wiley and Sons, Chichester, 1999.
- TANENBAUM, A. S.; STEEN, M. V. *Distributed systems.* [S.l.]: Prentice Hall, 2002.
- TANG, S.; YANG, J.; WU, Z. A context quality model for ubiquitous applications. In: IEEE. *Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on.* [S.l.], 2007. p. 282–287.
- TILAK, S.; MURPHY, A.; HEINZELMAN, W. Non-uniform information dissemination for sensor networks. In: IEEE. *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on.* [S.l.], 2003. p. 295–304.
- VANROMPAY, Y.; MEHLHASE, S.; BERBERS, Y. An effective quality measure for prediction of context information. In: IEEE. *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on.* [S.l.], 2010. p. 13–17.
- WEISER, M. The computer for the 21st century. *Scientific american*, Nature Publishing Group, v. 265, n. 3, p. 94–104, 1991.
- WIDYA, I.; BEIJNUM, B.-J. van; SALDEN, A. Qoc-based optimization of end-to-end m-health data delivery services. In: IEEE. *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on.* [S.l.], 2006. p. 252–260.

YASAR, A. et al. When efficiency matters: Towards quality of context-aware peers for adaptive communication in vanets. In: IEEE. *Intelligent Vehicles Symposium (IV), 2011 IEEE*. [S.l.], 2011. p. 1006–1012.

YAU, S. S. et al. Development and runtime support for situation-aware application software in ubiquitous computing environments. In: IEEE. *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*. [S.l.], 2004. p. 452–457.

YOON, C. et al. Appscope: Application energy metering framework for android smartphone using kernel activity monitoring. In: *USENIX ATC*. [S.l.: s.n.], 2012.

ZHENG, D.; WANG, J.; KERONG, B. Evaluation of quality measure factors for the middleware based context-aware applications. In: IEEE. *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on*. [S.l.], 2012. p. 403–408.

ZIMMER, T. Quality of context: Handling context dependencies. In: *Proceedings of 2nd International Workshop on Personalized Context Modeling and Management for UbiComp Applications, California, USA*. [S.l.: s.n.], 2006b.

ZIMMER, T. et al. Qoc: Quality of context-improving the performance of context-aware applications. *Advances in Pervasive Computing*, Citeseer, v. 207, n. 2, p. 209–214, 2006a.

ZIMMERMANN, A.; LORENZ, A.; OPPERMANN, R. An operational definition of context. *Modeling and using context*, Springer, p. 558–571, 2007.

## **APÊNDICE A - Trabalhos Publicados**



Este apêndice apresenta os trabalhos publicados ao longo do desenvolvimento dessa dissertação.

#### A.1 SEKE 2013 - INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING

•**Título:** A context-aware approach on semantic trajectories

•**Resumo:** Context-sensitive systems can use geographical position at a specific time to extract knowledge and determine patterns in user trajectories. However, this data can be unreliable or inaccurate, thus undermining the trajectories pattern inference process. Some researches adopt a procedure to eliminate redundancy and conflicts through the utilization of a centralized inference server. Other studies adopt a postprocessing procedure which is characterized by adding semantics to the spatial-temporal data. Both efforts consume unnecessary computational resources. Context data represents the state of an entity in a given time period, it provides information that allow to infer user behavior patterns with a set of semantics. In this research, it is proposed a differentiated context-aware approach to filter redundant and conflicting context information. The proposal aims to improve the inference process that determines patterns on user trajectories. The approach uses the processing power and storage capabilities of mobile devices to conduct operations to validate context data, thus eliminating useless information. Experimental results show an improvement both in the representation of spatial-temporal data and also on the semantic load that characterize the user trajectories

•**Evento:** The 25th International Conference on Software Engineering and Knowledge Engineering

•**Local:** Boston, USA

•**Data:** 27-29 de Junho

•**Autores:** Caio Silva e M.A.R. Dantas

•**Estrato Qualis/CAPES:** B1

#### A.2 WIMOB 2013 - INTERNATIONAL CONFERENCE ON WIRELESS AND MOBILE COMPUTING, NETWORKING AND COMMUNICATIONS

•**Título:** Quality-Aware Context Provider: A filtering approach to context-aware systems on ubiquitous environment

•**Resumo:** In ubiquitous environment context awareness approach plays a vital role in adaptability to users requirements. Context-sensitive systems meet specific requirements correctly, only if context information possesses the desired quality of context (QoC). Context data can be unreliable or inaccurate, thus undermining the potential of context-aware services. Some researches adopt a conflict resolution approach through the utilization of few QoC parameters. Other studies define policies based on QoC parameters to resolve conflicts. Both efforts do not exploit application requirements to ensure quality of context. In this research work it is proposed a QoC filtering approach. Our proposal intends to use in a better fashion the context provider, sending only relevant data to the context server. Besides that we present QoC policies, based in terms of application requirements and QoC parameters, which are used in a validation process to filter context data. The

experimental simulation indicates a reduction in the amount of data sent and also an improvement on the quality of context on transmitted data.

●**Evento:** The 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications

●**Local:** Lyon, França

●**Data:** 7-9 de Outubro

●**Autores:** Caio Silva e M.A.R. Dantas

●**Estrato Qualis/CAPES:** B1