

Gerência de QoC: Uma Abordagem Direcionada ao Tratamento de Conflitos de Contexto*

Lucas M. S. Xavier¹, Adenauer C. Yamin¹, Ana M. Pernas¹, Gerson G. H. Cavalheiro¹

¹Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPEL)
Rua Gomes Carneiro, 1 – 96.010-610 – Pelotas – RS – Brasil

{lmdsxavier, adenauer, marilza, gerson.cavalheiro}@inf.ufpel.edu.br

Abstract. *In context-aware systems, imprecise, ambiguous and erroneous information frequently arises from the many imperfections that naturally occur in the context acquisition and distribution processes. The negative impacts of such imperfections may be diminished through adequate modelling and management of Quality of Context (QoC) indicators, which are assessed from specific context metadata. This paper proposes a QoC management framework which deals with QoC metadata in an IoT middleware and uses the resulting indicators to reduce the quantity of conflicts caused by contradictory context information. A case study is presented in which the framework was used to provide QoC awareness to an ubiquitous application managed by the EXEHDA middleware.*

Resumo. *Nos sistemas cientes de contexto, os processos de aquisição e distribuição de informações estão inerentemente sujeitos a imperfeições que levam à produção de dados contextuais imprecisos, ambíguos e/ou errôneos. Tais imperfeições podem ter seus impactos minorados por meio da modelagem e da gerência de indicadores de Qualidade de Contexto (QoC), obtidos a partir de metadados contextuais. A proposta deste trabalho é a concepção de um arcabouço para manipulação de metadados de QoC em um middleware para IoT, com aplicação imediata na redução dos conflitos causados por dados contextuais contraditórios. Em um estudo de caso, emprega-se a solução desenvolvida para prover ciência de QoC a um ambiente ubíquo gerenciado pelo EXEHDA.*

1. Introdução

Sistemas cientes de contexto são capazes de caracterizar a situação de pessoas, lugares, objetos físicos e/ou dispositivos computacionais, empregando tais informações para suporte às aplicações reativas e adaptativas [Yürür et al. 2016]. Esses sistemas encontraram ampla aplicabilidade na computação ubíqua e na IoT, permitindo o enfrentamento de desafios referentes a aspectos como escalabilidade e abstração. Uma plataforma para ciência de contexto deve, entretanto, considerar a possibilidade de que as informações providas por suas fontes de contexto possam ser ambíguas, imprecisas, desatualizadas e/ou errôneas.

Em resposta à necessidade de tratar estas imperfeições, um número crescente de plataformas vêm adotando mecanismos de gerência de Qualidade de Contexto (em inglês, *Quality of Context* ou QoC) [Nazário et al. 2012]. Essas soluções expõem indicadores que sinalizam o grau de imperfeição associado a cada contexto.

*O presente trabalho foi realizado com apoio do Programa Nacional de Cooperação Acadêmica da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES/Brasil.

Assim, este trabalho tem como objetivo central a concepção de um arcabouço para gerência de indicadores de qualidade de contexto, compreendendo: (i) uma proposição de modelos para representação e manipulação das entidades de QoC, baseados em uma generalização dos trabalhos relacionados; e (ii) um conjunto de interfaces para exposição das variáveis de QoC para as aplicações. Adicionalmente, o trabalho visa avaliar a exploração do emprego de indicadores de QoC na sistematização de políticas para redução nos conflitos entre dados contextuais. Em função disso, além dos mecanismos básicos para tratamento de QoC, foi concebida uma estratégia para implementação de políticas baseadas em qualidade, visando a uma redução no número de potenciais conflitos de contexto.

As soluções projetadas foram materializadas como uma extensão das funcionalidades de manipulação de contexto do *middleware* EXEHDA [Lopes et al. 2014]. A implementação foi aplicada em um estudo de caso, no qual as ferramentas propostas foram avaliadas em um cenário que representa um ambiente ubíquo típico.

2. Qualidade de Contexto e Conflitos de Contexto

A definição de qualidade de contexto abrange toda informação que pode caracterizar a adequação de um dado a uma aplicação específica [Krause and Hochstatter 2005]. A título de exemplo, em sistemas cientes de contexto, essas informações frequentemente incluem metadados como: a atualidade do dado; sua precisão; sua importância; e o custo de sua obtenção. A qualidade de contexto distingue-se da Qualidade de Serviço (QoS) porque é intrínseca à informação em si, e não ao serviço que a produziu.

Os metadados de QoC podem, ainda, ser divididos em dois níveis de abstração [Bringel Filho and Agoulmine 2011]. Parâmetros de QoC são informações contextuais diretamente sensorizadas, que podem ser empregadas para extrair indicadores de qualidade. Indicadores de QoC são métricas de qualidade, derivadas dos parâmetros de QoC, que permitem estimar a adequação de um contexto com respeito a um aspecto definido. Como exemplo, o instante de produção de um dado é um parâmetro de QoC; e a atualidade desse dado é um indicador de QoC (derivado de dados como o tempo de vida esperado para a informação e o instante atual). Na literatura da área, os trabalhos existentes propõem listas distintas de indicadores, não havendo consenso quanto a um conjunto básico de aspectos de qualidade que devam ser gerenciados [Nazário et al. 2012].

A representação dos parâmetros e indicadores segue, em geral, a modelagem de contexto utilizada pelo *middleware* para o qual os arcabouços de QoC são concebidos. Dessa forma, são empregados modelos lógicos [Neisse et al. 2014], baseados em ontologia [Bringel Filho and Agoulmine 2011], em linguagens de marcação [Marie et al. 2015] ou em pares chave-valor, assim como para as demais informações contextuais.

Situações de conflito ocorrem em diversas camadas nos sistemas cientes de contexto [Manzoor et al. 2009]. Na camada de aquisição, há a necessidade de integrar os dados de sensores com características distintas de frequência de coleta, precisão e formato de representação. Na camada de processamento, mecanismos de inferência podem chegar a conclusões divergentes dependendo da maior ou menor confiabilidade da fonte de informações escolhida. Na camada de distribuição, a mobilidade dos sensores, o perfil operacional das redes sem fio e a natureza concorrente das aplicações ubíquas resultam em dados redundantes e/ou conflitantes, desperdiçando recursos de rede e processamento. A modelagem e o tratamento desses conflitos pode resultar na mitigação desses problemas.

Diversas técnicas são utilizadas para redução ou resolução de conflitos de contexto. Entre os métodos bem-sucedidos está o uso de indicadores de QoC para arbitrar a situação de conflito em favor da informação de maior qualidade. No trabalho de Manzoor et al. (2009), políticas baseadas em QoC são usadas para resolução de conflitos. Em Bringel Filho e Agoulmine (2011), parâmetros de QoC como *trustworthiness* e *probability of correctness* são integrados aos mecanismos de fusão de informações contextuais. Em Al-Shargabi e Siewe (2013), o histórico de contexto é usado para inferir parâmetros de QoC que alimentam um mecanismo de aprendizado baseado em regras de associação.

3. Trabalhos Relacionados

O OpenCOPI é um *middleware* orientado a serviços que age como integrador de plataformas coexistentes em um ambiente ubíquo [Lopes et al. 2013]. Possui uma arquitetura centralizada, com disseminação de informações no modelo *publish-subscribe*; e emprega ontologias para representação de contexto, bem como de indicadores de QoC e QoS. No OpenCOPI, as capacidades de gerência de QoC são utilizadas para alimentar as decisões internas de composição de serviços; o *middleware* determina dinamicamente o conjunto de serviços mais adequado para compor uma informação ou serviço para a aplicação final, baseando-se em dados de QoC e de custo computacional.

O projeto MobileHealthNet visa ao desenvolvimento de plataformas para redes sociais móveis e colaborativas, voltadas para a área da medicina ubíqua [Pinheiro 2014]. Sua infraestrutura para tratamento de contexto (o MHNCS) adota um modelo de disseminação baseado no padrão *publish-subscribe* e uma arquitetura distribuída. A representação do contexto é orientada a objetos, com utilização de documentos XML para troca ou persistência dos dados contextuais. O *middleware* expõe as interfaces de QoC para as aplicações, permitindo a solicitação de níveis mínimos ou máximos para os indicadores de qualidade, bem como a implantação de regras de processamento baseadas em QoC para disparo de eventos ou filtragem de contexto.

O *framework* MUCONTEXT faz parte do projeto INCOME, implementando modelos e ferramentas para gerência de privacidade e de QoC no contexto mais amplo de provisão de contexto para IoT [Marie et al. 2015]. O arcabouço adota uma arquitetura distribuída, com disseminação no padrão *publish-subscribe*. A modelagem de contexto é baseada no meta-modelo QoCIM e prioriza modelos gráficos, baseados em EMF e UML. Para transmissão e persistência dos dados contextuais, são empregados documentos XML. A gerência de QoC do MUCONTEXT estabelece contratos (representados em XML) com as aplicações, expressando garantias, requisitos, capacidades e limitações quanto aos indicadores de qualidade. Para proteção da privacidade dos usuários, serviços não-críticos podem receber dados com qualidade explicitamente reduzida para impedir a utilização indesejada. Os indicadores de QoC podem ser usados também para filtragem de contexto.

Com relação ao uso de QoC para resolução de conflitos de contexto, as principais técnicas adotadas na literatura [Manzoor et al. 2009, Bringel Filho and Agoulmine 2011, Al-Shargabi and Siewe 2013] estão contempladas na Seção 2.

4. EXEHDA-QoC: Concepção e Integração ao *middleware* EXEHDA

O EXEHDA é um *middleware* para aplicações ubíquas distribuídas, móveis e cientes de contexto. Sua organização é baseada em serviços dinamicamente carregados conforme a

demanda. Seu subsistema de tratamento de contexto possui uma arquitetura em dois níveis: uma camada central que interage com as aplicações (Servidor de Contexto) e uma camada periférica que interopera com sensores e atuadores (Servidor de Borda) [Lopes et al. 2014].

O modelo de qualidade de contexto concebido neste trabalho visa à integração com o *middleware* EXEHDA e se baseia parcialmente nas premissas do trabalho de Bringel Filho e Agoulmine (2011). Considera-se que as especificidades dos processos relacionados a uma publicação podem ser caracterizadas por metadados que são chamados de *parâmetros de QoC*. Um *indicador de QoC* é uma avaliação, baseada em parâmetros de QoC e representada por um número de 0 a 1, a respeito de uma característica específica de uma publicação que influencia na qualidade final dos dados. Admite-se que um indicador pode ser avaliado de várias formas, dependendo dos parâmetros disponíveis, por meio de diferentes *métodos de cálculo*.

A avaliação final de QoC é resultado de uma agregação realizada sobre os diversos indicadores. A maioria das plataformas emprega, para isso, uma média aritmética ponderada, em que as aplicações atribuem pesos aos diversos indicadores. O modelo do EXEHDA-QoC generaliza esse aspecto ao permitir que diversos *agregadores* sejam usados para produzir o valor final de QoC; assim, múltiplos esquemas de avaliação, baseados em diversas funções utilidade, são disponibilizados simultaneamente às aplicações.

Finalmente, um *avaliador de QoC* é um conjunto formado por um algoritmo e uma série de métodos de cálculo para diferentes indicadores. O avaliador determina unicamente qual o algoritmo utilizado, quais indicadores estão disponíveis e qual o método de cálculo utilizado para avaliar cada indicador. O modelo de contexto considerado prevê que os conjuntos de indicadores, parâmetros, algoritmos, métodos de cálculo e avaliadores não são fixos. Dessa forma, está contemplada a possibilidade de extensão das funcionalidades do framework por meio da inclusão de novos métodos ou indicadores.

No modelo exposto, a avaliação de QoC de uma publicação p (opcionalmente condicionada a um contexto de interesse t e a uma aplicação a) procede da seguinte forma:

1. Usa-se o avaliador vigente para determinar os indicadores $i_1 \dots i_n$ que serão considerados e métodos $m_{i_1} \dots m_{i_n}$ que serão empregados para avaliar esses indicadores.
2. Determinam-se os valores $\mu_1 \dots \mu_n$, que representam as avaliações dos indicadores de QoC para a publicação p considerada. Tais valores são dados por $\mu_k = m_{i_k}(p, a, t)$ para $i_k \in (i_1 \dots i_n)$.
3. Os valores de $\mu_1 \dots \mu_n$ e os pesos $w_1 \dots w_n$ são passados como parâmetro para o agregador f_n (determinado pelo avaliador). O valor final de QoC é dado por $QoC_v(p, a, t, (w_1 \dots w_n)) = f_n((\mu_1, w_1) \dots (\mu_n, w_n))$.

A visão de qualidade de contexto considerada foi consolidada na forma de uma biblioteca orientada a objetos e integrada ao modelo de contexto original do *middleware* EXEHDA. O diagrama de classes do modelo resultante é mostrado na Figura 1.

Como exemplo de aplicação dos arcabouços de QoC no aprimoramento das funcionalidades gerais do *middleware*, foi concebida uma abordagem baseada no trabalho de Manzoor et al. (2009) para redução de conflitos de contexto no EXEHDA. A estratégia tem como premissa o uso de políticas baseadas em indicadores de QoC para eliminar informações com alto potencial de conflito. Tais políticas são implementadas por uma série de filtros que atuam em pontos estratégicos da arquitetura do Servidor de Contexto.

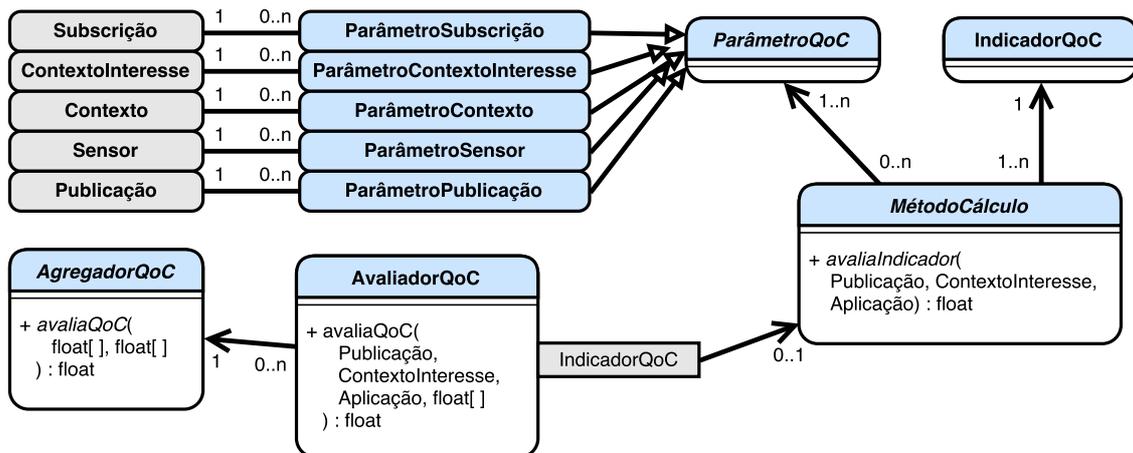


Figura 1. Diagrama de classes do modelo de QoC integrado ao modelo de contexto do EXEHDA

Especificamente, uma política de QoC é uma tupla $\pi = (v, \lambda, (w_1 \dots w_n))$ em que v é um avaliador, $w_1 \dots w_n$ é um conjunto de pesos e λ é um valor limiar mínimo de QoC. Uma publicação p (no escopo de um contexto de interesse t e de uma aplicação a) é aceita pela política π se $QoC_v(p, a, t, (w_1 \dots w_n)) \geq \lambda$; caso contrário, ela é rejeitada. Define-se Π como o conjunto de todas as políticas.

Um repositório de políticas $\pi^*(a, e, pr)$ é uma função que discrimina políticas específicas com base em uma aplicação a e em um escopo e (que é um sensor ou contexto de interesse). É previsto, ainda, o uso de políticas de aplicabilidade geral, que influenciam a todos os contextos de interesse e /ou a todas as aplicações. Nesse caso, emprega-se a notação $\pi^*(a, \phi, pr)$ e/ou $\pi^*(\phi, e, pr)$ e a política é compartilhada por todos os escopos ou aplicações finais. A prioridade pr , que pode ser alta (1) ou baixa (0), indica se essas políticas gerais devem ser substituídas ou aplicadas de forma concomitante às demais políticas mais específicas. Assim, π^* toma a forma $\pi^* : (A \cup \phi) \times (S \cup T \cup \phi) \times \{0, 1\} \rightarrow \Pi$.

Para avaliar se uma publicação p (destinada a uma aplicação a e com escopo delimitado por e) é aceita por um repositório de políticas, forma-se um conjunto julgamento $CJ_{a,e} \subset \Pi$. O conjunto $CJ_{a,e}$ é composto por todas as políticas de π^* que são aplicáveis considerando a e e . Se a publicação em questão é aceita por todas as políticas de $CJ_{a,e}$, a publicação é aceita pelo repositório como um todo; caso contrário, ela é rejeitada. O conjunto julgamento é construído da seguinte forma:

1. se existirem as políticas prioritárias $\pi^*(a, e, 1)$, $\pi^*(\phi, e, 1)$, $\pi^*(a, \phi, 1)$ e $\pi^*(\phi, \phi, 1)$, elas sempre fazem parte de $CJ_{a,e}$;
2. se não existe a política mais específica $\pi^*(a, \phi, 1)$ e existe a política mais geral (de baixa prioridade) $\pi^*(\phi, \phi, 0)$, então $\pi^*(\phi, \phi, 0)$ faz parte de $CJ_{a,e}$;
3. se não existe $\pi^*(a, e, 1)$ e existe $\pi^*(\phi, e, 0)$, então $\pi^*(\phi, e, 0)$ faz parte de $CJ_{a,e}$.

Para aplicação, no *middleware*, do modelo de políticas proposto, foram implantados três repositórios de políticas no Servidor de Contexto. O repositório de políticas de aquisição (π_{aq}^*) é usado para determinar se uma nova publicação possui QoC suficiente para ser persistida no histórico de contextos do EXEHDA. Se a publicação é rejeitada pelo repositório, o dado é descartado sem armazenamento.

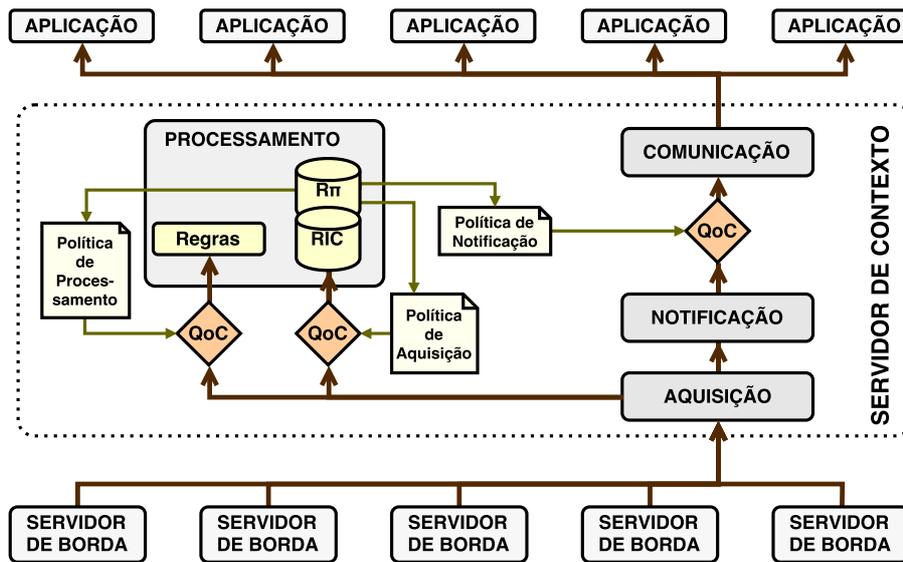


Figura 2. Diagrama indicando a integração dos repositórios de políticas à arquitetura do Servidor de Contexto (SC) do EXEHDA

O repositório de políticas de processamento (π_{proc}^*) é usado para disparo condicional do motor de regras do *middleware*. A rejeição de um contexto implica na suspensão do disparo das regras e rotinas associadas ao seu processamento. Por fim, o repositório de políticas de notificação (π_{not}^*) é empregado para ajustar os valores mínimos de qualidade associados às subscrições realizadas pelas aplicações ubíquas servidas pelo EXEHDA.

Os três repositórios descritos e sua integração à arquitetura do Servidor de Contexto estão delineados na Figura 2. A arquitetura original do *middleware* pode ser obtida em [Lopes et al. 2014]. O diagrama de classes do modelo usado para materializar as políticas de QoC está indicado na Figura 3.

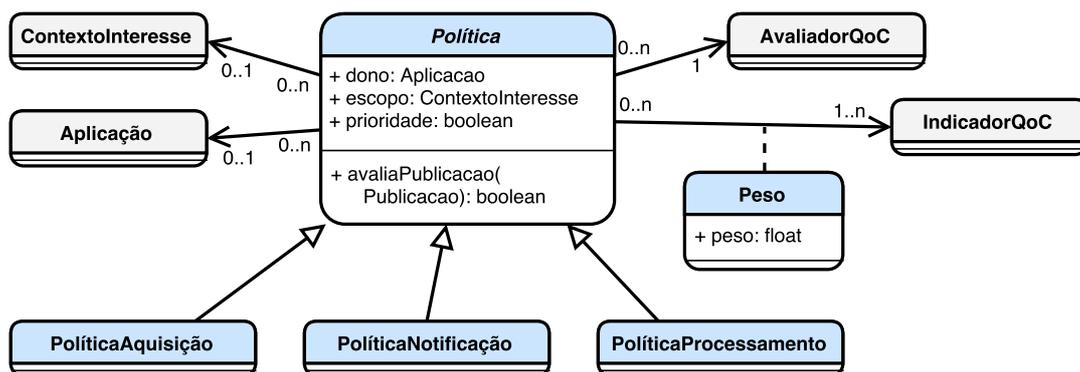


Figura 3. Diagrama de classes do modelo de políticas integrado ao modelo de QoC do EXEHDA

A solução de gerência de QoC proposta prevê que parte da funcionalidade deve ser flexibilizada para atender às especificidades de cada ambiente ubíquo, o que se traduz na possibilidade de incluir novos indicadores, funções de agregação e métodos de cálculo. Apesar disso, é necessário que todo Servidor de Contexto ofereça um conjunto padronizado de funcionalidades básicas, permitindo algum nível de interoperação entre diferentes

Tabela 1. Funções de agregação padrão do EXEHDA-QoC

Agregador	f_{linear}	$f_{mínimo}$	$f_{geométrica}$	$f_{harmônica}$	f_{CES}
$f(\mu_k, w_k) = \dots$	$\sum(\widehat{w}_k \mu_k)$	$\min(\mu_k)$	$\prod \mu_k^{\widehat{w}_k}$	$\left(\sum \frac{\widehat{w}_k}{\mu_k}\right)^{-1}$	$\left[\widehat{w}_k \mu_k^\rho\right]^{\kappa/\rho}$

células ubíquas. Assim, foi definido um conjunto padrão de funcionalidades, obrigatoriamente implementadas, que representam indicadores e formulações comumente encontrados na literatura da área. De forma mais específica, o conjunto padrão compreende:

- **Um conjunto padrão de indicadores de QoC.** A *atualidade* assinala a magnitude do intervalo de tempo transcorrido entre a coleta e o consumo de um dado contextual, diminuindo conforme a informação se torna obsoleta. A *precisão* está associada à magnitude dos erros aleatórios que se manifestam nos processos de aquisição, processamento e distribuição. A *importância* é definida para cada aplicação e é um indicador arbitrário, usado para discriminar dados críticos.
- **Um conjunto padrão de métodos de cálculo,** que inclui um método para cada indicador do conjunto padrão.
- **Um conjunto padrão de funções de agregação.** Em geral, as soluções de QoC existentes empregam uma média aritmética ponderada entre os indicadores. O EXEHDA-QoC flexibiliza os agregadores, permitindo a implementação de funções utilidade diversas. As funções de agregação padrão estão sumarizadas na Tabela 1. Nelas, os valores de \widehat{w}_k representam pesos *normalizados*, proporcionalizados para a soma de todos os pesos seja igual a 1. Nota-se que o agregador f_{CES} é parametrizado por dois valores adicionais (ρ e κ), que controlam o grau de homogeneidade e a elasticidade da função. Além das funções indicadas na Tabela 1, é fornecido um agregador f_{limiar} que retorna 1 se $\forall k(\mu_k \geq w_k)$ e 0 caso contrário.
- **Dois avaliadores padrão,** que empregam as funções de agregação baseadas em limiar (v_{limiar}) e em média aritmética (v_{linear}). Os avaliadores padrão usam os três indicadores e os métodos de cálculo padrão mencionados.

Para transferência de políticas e de informações de QoC entre os componentes do EXEHDA e entre aplicações, as APIs do Servidor de Contexto foram estendidas. Foram definidos dois formatos, baseados em JSON e especificados empregando o padrão JSON Schema, para representar: (i) requisitos e informações gerais de QoC; e (ii) políticas de QoC. Finalmente, para operacionalizar o emprego do arcabouço de QoC desenvolvido, as interfaces gráficas de gerência e configuração do *middleware* foram ampliadas com funcionalidades para cadastro, atualização e visualização das novas entidades definidas.

5. Estudo de Caso

Foi desenvolvido um estudo de caso visando à demonstração das funcionalidades do EXEHDA-QoC em um cenário representativo de um ambiente ubíquo real. O estudo foi baseado na simulação de uma célula ubíqua com agentes, dispositivos e aplicações móveis produzindo e/ou consumindo informações contextuais com características de QoC diversas. O cenário do estudo de caso é baseado no trabalho de Li et al. (2014), no qual sensores tipicamente encontrados em *smartphones* são usados para determinar se um usuário está em um ambiente interno ou externo. Especificamente, são usados os sensores de iluminação, magnetismo e a intensidade de sinal da rede de telefonia celular.

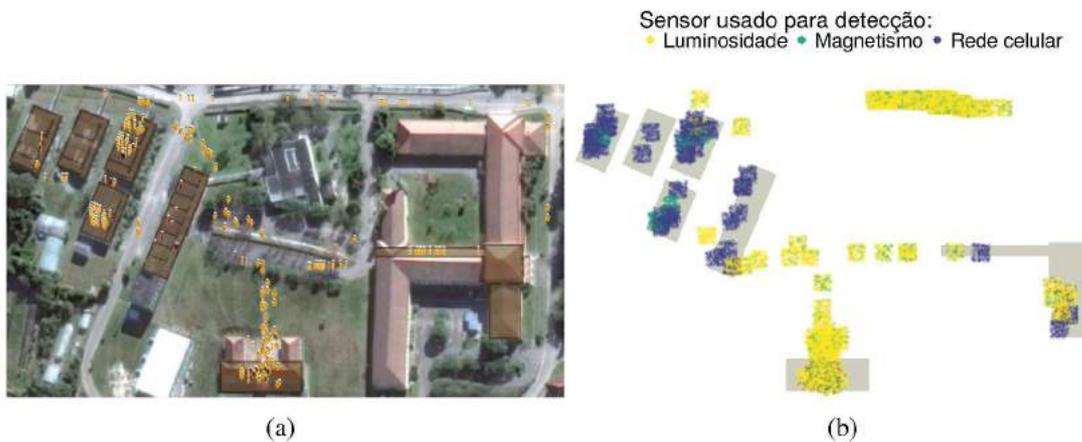


Figura 4. (a) Imagem do ambiente simulado. (b) Sensores empregados para detecção de ambiente interno/externo, de acordo com a localização dos agentes. As áreas em cinza são ambientes internos.

A iluminação é empregada porque a luz solar é significativamente mais intensa que a artificial; entretanto, o sensor pode ser bloqueado pelo usuário a maior parte do tempo. A intensidade do sinal da rede de telefonia móvel apresenta variações abruptas quando o usuário deixa ou ingressa em construções; contudo, está sujeita a efeitos de linha de visão da antena e de propagação do sinal. A intensidade do campo magnético é aproximadamente constante em áreas externas e amplamente variável perto de construções; por isso, permite detectar ambientes externos, mas é insuficiente para diferenciar ambientes internos de suas proximidades. O uso concomitante dos três sensores atenua problemas relacionados à disponibilidade ou à acurácia de um dos tipos de detecção.

O ambiente simulado (Figura 4(a)) considera o seguinte cenário de uso: em um *campus* universitário, os estudantes e professores (referenciados como agentes) utilizam uma aplicação móvel que seleciona conteúdo dinamicamente de acordo com a localização (interna ou externa) do usuário. Os celulares dos agentes enviam as medições dos sensores periodicamente para o EXEHDA. Um indicador de QoC (*confiabilidade*) é atribuído pelo Servidor de Borda de acordo com a acurácia e a disponibilidade previstas para os sensores. Políticas de QoC são usadas na camada de aquisição do Servidor de Contexto para descartar dados potencialmente conflituosos, de acordo com a *confiabilidade*. A aplicação móvel solicita dados de contexto periodicamente, buscando estimativas para determinar se o usuário está em ambiente interno ou externo. Na requisição, são usadas a *confiabilidade* e a *atualidade* para priorizar os dados, baseando a detecção em um ou outro sensor.

O cenário foi simulado com o auxílio da ferramenta Siafu [Martin and Nurmi 2006], sendo utilizada uma planta com sete prédios representando salas de aula e de professores, um auditório, uma cantina e uma parada de ônibus. A simulação representou um dia de atividades, ao longo do qual cerca de 500 agentes se alternaram em atividades acadêmicas e de lazer, circulando entre os prédios. A cada vinte segundos os agentes enviaram informações de contexto para o EXEHDA; nesse instante, a aplicação móvel solicita dados de localização ao *middleware* (em 6,6% dos envios).

Os dados enviados do Servidor de Borda para o Servidor de Contexto são valores entre 0 e 1 que indicam a probabilidade de o agente estar em um ambiente interno, de

Tabela 2. Matriz de confusão para o resultado das solicitações de previsão realizadas pelos agentes.

		Previsto		Total
		Interno	Externo	
Real	Interno	9692	983	10675
	Externo	6	23229	23235
	Total	9698	24212	33910

acordo com as informações de apenas um dos sensores. As políticas de aquisição do Servidor de Contexto rejeitam informações com *confiabilidade* inferior a 0,5, correspondentes a sensores bloqueados ou em condições de coleta inadequadas. Na solicitação das informações, a aplicação móvel demanda o uso do avaliador linear padrão para maximizar o valor de QoC usando pesos de 0,5 para *atualidade* e *confiabilidade*.

O ambiente experimental empregou duas máquinas virtuais: uma para execução do Servidor de Borda e do Sifafu, e a outra para execução do Servidor de Contexto. Ao longo da simulação, que durou doze horas, os agentes produziram aproximadamente quinhentas mil publicações e solicitaram informações contextuais 33910 vezes.

A Figura 4(b) mostra a influência das políticas de aquisição na seleção da fonte contextual mais adequada. No gráfico, estão reproduzidas as localizações das solicitações realizadas pelos agentes, coloridas de acordo com o sensor que foi usado para a detecção (foi adicionado um pequeno deslocamento aleatório aos pontos para permitir a visualização). É possível verificar que, nas áreas externas, a detecção priorizou os sensores de iluminação e magnetismo, enquanto nos ambientes internos o sinal da rede de telefonia móvel foi usado com mais frequência. Assim, os sensores foram empregados predominantemente em ambientes que representam circunstâncias de aquisição favoráveis à sua acurácia.

Além dos impactos na aquisição do contexto, o arcabouço de gerência permitiu a seleção das informações mais apropriadas dentre as disponíveis no repositório de informações contextuais. No caso da aplicação simulada, os requisitos de QoC adotados objetivaram à manutenção de um nível elevado de acurácia para a detecção de ambiente interno. A Tabela 2 aponta a quantidade de previsões corretas e incorretas realizadas pelos mecanismos de provisão de contexto do EXEHDA-QoC.

6. Conclusão e Trabalhos Futuros

Neste trabalho, foram propostos mecanismos para gerência de dados de QoC em plataformas para computação ubíqua. Além da exploração das estratégias básicas para manipulação de QoC presentes na literatura, produziu-se, como contribuição, a concepção de um arcabouço que emprega os indicadores de qualidade para mitigar os impactos negativos causados por informações conflitantes. Os modelos propostos foram materializados por meio de sua implementação e de sua integração à arquitetura do *middleware* EXEHDA, sendo aplicados ao seu subsistema de tratamento de contexto. As funcionalidades do EXEHDA-QoC foram exploradas em um cenário de aplicação, representativo de um subconjunto dos sistemas em funcionamento em um ambiente ubíquo típico.

Entre as frentes de estudo prioritárias para sequência do trabalho, pode-se destacar a introdução de uma representação semântica para os dados de QoC, fazendo uso das

capacidades de modelagem híbrida de contexto do *middleware* EXEHDA. Ainda, pontua-se a integração das funcionalidades de QoC à arquitetura dos Servidores de Borda, permitindo a avaliação de qualidade nos componentes mais próximos da aquisição e da atuação.

Referências

- Al-Shargabi, A. A. and Siewe, F. (2013). Resolving context conflicts using association rules (RCCAR) to improve quality of context-aware systems. In *Computer Science Education, 2013 8th International Conference on*, pages 1450–1455, Colombo. IEEE.
- Bringel Filho, J. and Agoulmine, N. (2011). A quality-aware approach for resolving context conflicts in context-aware systems. In *Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on*, pages 229–236, Melbourne. IEEE.
- Krause, M. and Hochstatter, I. (2005). Challenges in modelling and using quality of context (QoC). In Magedanz, T., Karmouch, A., Pierre, S., and Venieris, I., editors, *Mobility Aware Technologies and Applications*, pages 324–333. Springer, Berlin, Heidelberg.
- Li, M., Zhou, P., Zheng, Y., Li, Z., and Shen, G. (2014). IODetector: A generic service for indoor/outdoor detection. *ACM Trans. Sen. Netw.*, 11(2):28:1–28:29.
- Lopes, F., Delicato, F. C., Batista, T., Cavalcante, E., Pereira, T., Pires, P. F., Ferreira, P., and Mendes, R. (2013). OpenCOPI: middleware integration for ubiquitous computing. *International Journal of Parallel, Emergent and Distributed Systems*, 29(2):178–212.
- Lopes, J., Souza, R., Geyer, C., Costa, C., Barbosa, J., Pernas, A., and Yamin, A. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *Journal of Universal Computer Science*, 20(9):1327–1351.
- Manzoor, A., Truong, H.-L., and Dustdar, S. (2009). Using quality of context to resolve conflicts in context-aware systems. In *Quality of Context: First International Workshop*, pages 144–155. Springer, Berlin, Heidelberg.
- Marie, P., Desprats, T., Chabridon, S., Sibilla, M., and Taconet, C. (2015). From ambient sensing to IoT-based context computing: An open framework for end to end QoC management. *Sensors*, 15(6):14180.
- Martin, M. and Nurmi, P. (2006). A generic large scale simulator for ubiquitous computing. In *Mobile and Ubiquitous Systems - Workshops, 2006. 3rd Annual International Conference on*, pages 1–3, San Jose. IEEE.
- Nazário, D., Dantas, M., and Todesco, J. L. (2012). Taxonomia das publicações sobre qualidade de contexto. *Sustainable Business International Journal*, 20(2012):1–28.
- Neisse, R., Wegdam, M., and van Sinderen, M. (2014). Trust management support for context-aware service platforms. In Aldini, A. and Bogliolo, A., editors, *User-Centric Networking: Future Perspectives*, pages 75–106. Springer International, Cham.
- Pinheiro, D. N. (2014). MHNCS: Um middleware para o desenvolvimento de aplicações móveis cientes de contexto com requisitos de QoC. Dissertação de mestrado, Universidade Federal do Maranhão, São Luís.
- Yürür, Ö., Liu, C. H., Sheng, Z., Leung, V. C. M., Moreno, W., and Leung, K. K. (2016). Context-awareness for mobile sensing: A survey and future directions. *IEEE Communications Surveys Tutorials*, 18(1):68–93.