UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

VALDERI REIS QUIETINHO LEITHARDT

UbiPri – Middleware para Controle e Gerenciamento de Privacidade em Ambientes Ubíquos

Tese apresentada como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Cláudio Fernando Resin Geyer
Instituto de Informática – UFRGS – Brasil.

Co-orientador: Prof. Dr. Jorge Miguel Sá Silva
Departamento de Engenharia Informática (DEI)
Universidade de Coimbra – Portugal.

Porto Alegre
2015

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Leithardt, Valderi Reis Quietinho
UbiPri – Middleware para Controle e Gerenciamento de Privacidade em Ambientes Ubíquos / Valderi Reis Quietinho Leithardt. – Porto Alegre: Programa de Pós-Graduação em Computação 2015.
120 f.:il.
Tese (Doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2015. Orientador: Claudio Fernando Resin Geyer; Co-orientador: Jorge Miguel Sá Silva.
1.Computação Ubíqua. 2. Middleware 3.Taxonomia. 4. Privacidade.
I. Valderi Reis Quietinho Leithardt, Claudio Fernando Resin Geyer. II. Jorge Miguel Sá Silva. UbiPri – Middleware para Controle e Gerenciamento de Privacidade em Ambientes Ubíquos.

AGRADECIMENTOS

Agradeço primeiramente a Deus por estar sempre ao meu lado em todos os momentos da minha vida, por se fazer presente na finalização deste trabalho, que também contou com ajuda de várias pessoas. São tantos os agradecimentos que me antecedo desculpand-me por não colocar o nome de todas as pessoas queridas que me auxiliaram ao longo desta trajetória. Dentre as quais minhas filhas Jheine e Nathaly, a vocês minha eterna gratidão por tantos momentos que deixamos de estar juntos por conta das atividades que envolviam o desenvolvimento desta tese. Aos meus Pais que na medida do possível, me apoiaram em seguir nos estudos e a continuar trilhando meu futuro. Agradeço a minha esposa Daiana Rosso Leithardt pela motivação e paciência nesses últimos meses que antecederam a finalização desta tese. De forma alguma poderia deixar de agradecer aos inesquecíveis e inseparáveis amigos Julio C.S Anjos, Guilherme Antonio Borges, Anubis Graciela Rossetto, Carlos Oberdan Rolim e demais colegas da sala 205 do instituto de informática da Universidade Federal do Rio Grande do Sul (UFRGS). Não posso deixar de agradecer ao Grupo de Processamento Paralelo e Distribuídos da UFRGS, em especial ao meu Orientador Professor Doutor Claudio Fernando Resin Geyer que foi muito além de um orientador de doutorado, se mostrou um ser humano incrível em momentos cruciais. Ao sempre disposto e prestativo Co – Orientador Professor Dr. Jorge Miguel Sá Silva do departamento de engenharia informática da Universidade de Coimbra Portugal, onde pude desenvolver parte de minha tese realizando doutorado Sanduíche. Neste mesmo departamento, além do grande aprendizado que obtive tive o privilégio de conhecer e trabalhar com colegas fantásticos como Ricardo Mendão, David Nunes e Duarte Raposo, boa e incomparável experiência obtida. Agradeço também aos alunos, orientandos e também bolsistas que trabalharam comigo no decorrer desta tese e também contribuíram para a conclusão. Agradeço a Federação das Indústrias do Estado do Rio Grande do Sul (FIERGS), ao Serviço Nacional de Aprendizagem Indústrial (SENAI) pela disponibilidade e oportunidade de coordenar o Grupo de Pesquisas em Processamento Paralelo e Distribuído Inteligente (GPPD-i). E por fim, os agradecimentos para a Coordenação de Aperfeiçoamento de Nível Superior (CAPES), ao Programa Ciências sem Fronteiras CsF e ao Conselho Nacional de Desenvolvimento Cientifico e Tecnológico (CNPq) pelo aporte financeiro no decorrer deste trabalho.

RESUMO

Atualmente em vários países já existem mais dispositivos e meios de comunicações que habitantes e a medida que a tecnologia avança a troca de informação tende a aumentar exponencialmente. Com isso, ganha destaque a área denominada computação ubíqua, que visa o desenvolvimento de aplicativos para automatizar processos, antes manuais, a fim de proporcionar conforto, rapidez e conexão aos usuários com seus dispositivos. Nos estudos realizados decorrer desta tese identificou–se a necessidade de desenvolver e controlar informações privadas fundamentadas no local, aqui denominado ambiente ubíquo. O problema de pesquisa identificado foi a grande heterogeneidade de dispositivos e comunicações nestes sistemas, tornando-os vulneráveis e expondo os dados de seus usuários. Assim, observou-se a necessidade de um modelo taxonômico de privacidade que engloba características necessárias para controlar e gerenciar a privacidade de dados em ambientes ubíquos. A partir dessa taxonomia desenvolveu–se um protótipo com base em um middleware estruturado em camadas necessárias para prover os controles e gerenciamentos necessários nestes ambientes. Os primeiros testes e resultados se mostraram promissores, tendo seus resultados publicados em conferências internacionais da área que nortearam os estudos para uma melhoria do tratamento e filtragem de dados. Também foi possível a ampliação dos controles e gerenciamento de parâmetros automáticos com aumento e redução de definição do tipo de perfil do usuário em adição são apresentados os resultados obtidos em diferentes cenários de uso e aplicação. Para tanto, o protótipo desenvolvido permite selecionar opções de variáveis atribuídas individualmente a cada ambiente de acordo com suas necessidades, com isso, a solução proposta visa ser empregada no gerenciamento de privacidade em ambientes ubíquos. Os resultados obtidos nos testes realizados comprovam a viabilidade e contribuição cientifica do modelo desenvolvido. O aplicativo UbiPri foi disponibilizado para utilização no google play store, podendo ser instalado e configurado na plataforma android.

Palavras-chave: Computação Ubíqua. Middleware. Taxonomia. Privacidade.

UBiPri - Middleware Control and Privacy Management in Ubiquitous Environments

ABSTRACT

Currently in many countries there are already more devices and communication means inhabitants and as technology advances the exchange of information tends to increase exponentially. As a result, stands out the area called ubiquitous computing, which aims to develop applications to automate processes before hand in order to provide comfort, speed and connecting users with their devices. In studies carried out throughout this thesis it identified the need to develop and control private information based on location, here called ubiquitous environment. The identified research problem was the great heterogeneity of devices and communications in these systems, making them vulnerable and exposing the data of its users. Thus, there was the need for a taxonomic model of privacy that encompasses features needed to control and manage data privacy in ubiquitous environments. From this taxonomy developed a prototype based on a middleware structured in layers to provide the necessary controls and managements required in these environments. The first tests and results were promising, with the results published in international conferences in the area that guided the studies for improved treatment and filtering of data. It was also possible the expansion of controls and management parameters with automatic increase and decrease setting in the user profile type in addition the results obtained are presented in different usage scenarios and application. Thus, the prototype allows you to select options variables individually assigned to each environment according to their needs, with it, the proposed solution is intended to be used in the privacy management in ubiquitous environments. The results obtained in the tests prove the feasibility and scientific contribution of the developed model. The UbiPri application was made available for use in the google play store and can be installed and configured on android platform.

Keywords: Ubiquitous Computing. Middleware. Taxonomy. Privacy.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
CORBA	*Common Object Request Broker Architecture*
CTI	Centro de Tratamento Intensivo
DDR-RAM	*Double Data Rating - Random Access Memory*
DSP	*Digital Signal Processing*
E/S	Entrada/Saída
GIOP	*General Inter ORB Protocol*
GPS	Sistema de Posicionamento Geográfico
GTSH	*Gator Tech Smart House*
HW	*Hardware*
I/O	*Input/output*
IP	*Internet Protocol*
MW	*Middleware*
NFS	*Network File System*
ORB	*Object Request Broker*
PC	*Personal Computer*
PDA	*Personal Digital Assistant*
QoS	*Quality of Software*
RFID	*Radio Frequency Identification*
RSSFs	Redes de Sensores sem Fios
SDP	*Service Discovery Protocol*
SO	Sistema Operacional
SW	*Software*
TCP/IP	*Transmission Control Protocol / Internet Protocol*
UPnP	*Universal Plug and Play*
USB	*Universal Serial Bus*
UFRGS	Universidade Federal do Rio Grande do Sul
XML	*Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Com a crescente evolução e proliferação dos dispositivos móveis, diversos grupos de pesquisa estão direcionando seus projetos a um novo cenário computacional. Este novo cenário contempla novos tipos de aplicações mais complexas atuando em ambientes inteligentes altamente dinâmicos e também conhecidos como computação ubíqua. Considerada como a Terceira Era da Informática, a computação ubíqua surgiu em 1991 com a visão de Weiser. De forma visionária, Weiser vislumbrava a ideia de que no futuro os computadores habitariam os mais triviais objetos, como etiquetas de roupas, canetas, interruptores de luz, etc. Esses diversos dispositivos computacionais interagiriam naturalmente com as pessoas e entre si, tornando-se parte do ambiente. No entanto, em 1991, a tecnologia era insuficiente para concretizar essa visão segundo SAHA e MUKHERJEE (2003).

Outro conceito relacionado é a Computação Pervasiva, que segundo Araújo (2003) implica que o computador está embarcado no ambiente de forma invisível para o usuário. Nesta visão, o computador tem a capacidade de obter informação do ambiente, no qual está inserido, e utilizá-la para dinamicamente construir novos modelos computacionais de forma a atender às necessidades do dispositivo ou usuário. Assim, Araújo (2003) descreve que a computação ubíqua integra a mobilidade em larga escala com a funcionalidade da computação pervasiva, isto é, qualquer dispositivo computacional, sob posse de um usuário em movimento, pode construir dinamicamente modelos computacionais do ambiente em que está inserido e configurar seus serviços dependendo da necessidade. Conforme apresentado na Figura 1.1. A ideia básica é que a computação move-se para fora das estações de trabalho e PCs e torna-se ubíqua no cotidiano dos indivíduos e a evolução dos dispositivos e sistemas computacionais.

Figura 1.1 - Evolução Computacional adaptado de Weiser.

Hoje são constantes a busca pela fusão da mobilidade da computação móvel com a capacidade de interação com o meio, agregada pela computação pervasiva e ubíqua. Assim, pesquisadores visualizam a computação pervasiva e ubíqua como sinônimos. O autor ainda ressalta que a diferenciação entre os termos ubíquo e pervasivo está relacionado ao fato de que um dispositivo que está embutido em um ambiente não necessariamente é móvel. Assim, o termo computação ubíqua é utilizado para denotar o alto grau de dispositivos embarcados da computação pervasiva juntamente com o alto grau de mobilidade da computação móvel (ARAÚJO, 2003).

Outro fato que se observa e que comprova essa similaridade entre computação pervasiva e ubíqua, consiste em que após uma década de progressos na computação, a miniaturização dos dispositivos de identificação por Rádio Frequência (RFID) avançou em larga escala, conforme Zhu, Mutka e Ni (2005). Outras evoluções ocorreram na comunicação com as redes sem fios em conjunto com o Bluetooth, apontada em (RFID JOURNAL, 2014), em conjunto com Wi-Fi, Ad-Hoc e redes inteligentes conforme descrito em Souza et al. (2013). Assim, a proposta de Weiser vem pouco a pouco se materializando. No entanto, esse progresso implicou em três desafios à computação ubíqua.

O primeiro se caracteriza pela interação de uma grande quantidade de dispositivos heterogêneos, de acordo com Chen, Finn, Joshi (2003), esses dispositivos podem variar desde servidor de propósito geral, possuindo alto poder computacional até minúsculos sensores. Além disso, esses dispositivos podem ser conectados com outros dispositivos de diversas formas, tais como: com fio, sem fio, infraestruturada ou ad hoc. Desta forma, é possível compartilhar as diversas funcionalidades dos dispositivos, ocasionando uma alta heterogeneidade no ambiente. O segundo desafio está relacionado à mobilidade do usuário, sendo um dos quesitos fundamentais da computação ubíqua que é permitir a onipresença, ou seja, fornecer ao usuário acesso computacional em qualquer lugar, em qualquer momento, conforme Rodrigues (2006). Essa mobilidade, acrescida às variações na comunicação da rede, torna o ambiente com disponibilização de recursos e serviços ainda mais complexo devido a sua heterogeneidade. O terceiro desafio diz respeito à invisibilidade. A ideia é que as aplicações da computação ubíqua continuamente satisfaçam as expectativas do usuário de forma independente, sem que haja interação. Desta forma, pode-se interagir com o usuário quase no nível subconsciente, conforme Saha e Mukherjee (2003), permitindo alcançar a ideia de interação natural com as pessoas, de acordo com Satyanarayanan (2001). Para tanto, as aplicações ubíquas devem ser conscientes de contexto. O conhecimento das informações e do estado, tanto do ambiente quanto do usuário, permitiria que as aplicações ubíquas contidas

nesses ambientes se adaptassem de maneira eficiente para atender as expectativas do usuário, segundo saha e Mukherjee (2003). Devido a esses três desafios (heterogeneidade, disponibilização inconstante de recursos, e consciência do contexto), as aplicações da computação ubíqua são complexas e devem ser inteligentes para suportar as mudanças ocorridas no ambiente.

Portanto, é necessário levar em consideração as diversas dificuldades relacionadas no tratamento e gerenciamento de privacidade que se fazem necessárias, visto que os dados gerados em diferentes ambientes também estarão relacionados a dispositivos e usuários. Os dados gerados podem não disponibilizar informações necessárias para justificar o uso e consolidação desses ambientes ubíquos, tornando esses desafios ainda maiores, com aumento significativo de dificuldade. Com a evolução tecnológica no decorrer dos anos, surgiu o conceito denominado internet das coisas (*Internet of Things* – IoT), que engloba diversos dispositivos com diferentes funcionalidades que podem estar interligados em um mesmo ambiente ou mesmo em ambientes separados, conforme Kagal, Finin e Joshi (2001). Basicamente esse conceito se fundamenta na computação ubíqua relacionada à computação móvel e em demais dispositivos e equipamentos eletrônicos, adaptando-se ao ambiente e a seu projeto de infraestrutura, tornando ainda mais complexo o controle e gerenciamento de privacidade. Conforme as limitações do dispositivo e as necessidades de cada ambiente denominado ubíquo, alguns pesquisadores como David Culler (Universidade de Berkeley) desenvolvem projetos que estão sendo desenvolvidos na mesma linha de pesquisa. A Internet das coisas é um projeto inicialmente desenvolvido pelo MIT (*Auto-ID Laboratory*), recorrendo ao uso do RFID e Redes de Sensores sem Fios (RSSFs). O objetivo do projeto foi criar um sistema global de registro de bens usando dados de vários produtos chamados *Electronic Product Code*.

Com isso, a computação ubíqua também deve ser sensível ao contexto, apresentando soluções diferentes para situações diversas. Uma situação distinta pode ser caracterizada pela mudança de usuários, mudança nas condições de localização, disponibilidade de serviços climáticos ou de tempo, como datas e horários por exemplo. Ainda existem outras situações que que podem influenciar informações do mesmo contexto computacional do usuário ou ambiente em que se encontra, independente de novos dispositivos e conceitos que possam surgir. Portanto, se faz necessário desenvolver mecanismos para controle e o gerenciamento de privacidade, visto que, por um lado o usuário pode não precisar ou não querer ser localizado ou ainda não desejar ter compartilhamento de seus dados a todo momento. Por outro lado, tais informações podem ser melhor administradas por parte do ambiente ubíquo,

com isso, também se busca reduzir o processamento de dados desnecessários, aumentando níveis de segurança e consequentemente o gerenciamento dos serviços disponíveis. Considerando todos os mecanismos necessários para que sistemas ubíquos possam interagir, tais como o dinamismo e a autonomia para proporcionar a invisibilidade e a adaptabilidade de contexto, a privacidade além de se mostrar cientificamente relevante, para esses tipos de sistema, possui um grande valor intrínseco para sua aplicabilidade no mundo real. Neste contexto, o controle e gerenciamento de privacidade vem se mostrando fundamental, tendo em vista a grande troca de informações que existe em ambientes ubíquos. De acordo com Leithardt et al. (2013a), um ambiente ubíquo pode ser classificado como qualquer lugar onde o usuário possa estar presente, sendo possível controlar e gerenciar funcionalidades, aplicações e aplicativos de maneira onipresente, sendo atribuídas ainda, funcionalidades e características de acordo com o contexto individual dos dispositivos e/ou usuários a cada ambiente individualmente.

Na literatura pesquisada, vários trabalhos encontrados tratam sobre o controle de privacidade direcionado ao **usuário,** aos **dispositivos**, aos **serviços** ou a **comunicação** destacando os pesquisadores Hübner (2012), Langheinrich (2001), Langheinrich (2002), Yitao Duan and John Canny (2006). Como exemplo desses cenários, pode-se citar aplicações em igrejas, bibliotecas, cinemas, salas de aula, entre outros. Embora existam diversos trabalhos que abordem o tema controle de privacidade, estes não são relacionados ao controle e gerenciamento do **ambiente ubíquo.** Neste trabalho, o controle e gerenciamento é atribuído ao ambiente em que o usuário se encontra, com isso o ambiente atuará como dominante, ou seja, seguindo regras e critérios por ele determinados. Para tanto, serão necessários informações e detalhes específicos para o controle e gerenciamento de privacidade destes ambientes, conforme descreve Warren and Bradeis (1890) a privacidade é o direito de ser deixado só, portanto, ninguém quer ou necessita ser localizado ou encontrado o tempo todo. Considerando que esses ambientes são regidos por regras próprias e individuais, os critérios de usuários que os definem como autoridade delimitadora do ambiente, as comunicações e serviços, ficam subordinadas a eles perdendo total ou parcialmente a validade se seus interesses quando conflitam com o ambiente. Como exemplo, seria a situação de um usuário possuir definições como som alto em seu dispositivo móvel, porém, ao entrar em um determinado ambiente o mesmo passa a tocar em modo silencioso, caso a regra de critérios do ambiente se sobreponha às regras do dispositivo pessoal do usuário.

1.1 Objetivo Geral

Com base nas características necessárias para o controle e gerenciamento de privacidade citadas anteriormente, este trabalho possui como principal contribuição o desenvolvimento de um modelo de controle e gerenciamento da privacidade focado em ambientes ubíquos, tendo por premissa fundamental relacionar a computação ubíqua mais próxima e presente do mundo real. No entanto, com este enfoque, não se tem por objetivo abordar os problemas de segurança em computação ubíqua, como técnicas de evitar ataques ou criptografia de informação, este trabalho tão pouco aborda como serão realizados os controles restritivos de usuários e/ou dispositivos, bem como seus serviços e formas de comunicação.

1.2 Motivação

Trabalhos pesquisados na literatura contemplam o controle e gerenciamento de ambientes ubíquos sob a visão do dispositivo, usuário, tratamento de protocolos, entre outras formas de controle. No entanto, o que motiva o proponente são os controles e gerenciamentos desses ambientes com enfoque em suas definições, critérios e parâmetros que são atribuídos individualmente ao próprio ambiente ao invés dos objetos, usuários e comunicação que o mesmo utiliza, no entanto, sem deixar de considerá-los. O motivo principal consiste no controle e gerenciamento de privacidade do ambiente, considerando também, os dados provinientes dos dispositivos, objetos e usuários. Devido a grande quantidade de dados que precisam ser tratados, o controle e gerenciamento de privacidade não se mostra essencial na computação ubíqua.

1.3 Objetivos Específicos

Os objetivos específicos desta tese são abrangentes em relação às contribuições científicas que se apresentaram no desenvolvimento das pesquisas relacionadas ao controle e gerenciamento de privacidade. Com base em resultados preliminares obtidos, foram publicados relevantes artigos, comprovando assim a viabilidade deste trabalho. Além disso, novos desafios de pesquisa, que não estavam previstos inicialmente surgiram, com isso, os seguintes objetivos específicos tornaram – se necessários, são eles:

a) identificar e gerenciar informações taxonômicas;

b) controlar e gerenciar a privacidade de ambientes ubíquos;
c) controlar as características individuais de cada ambiente;
d) gerenciar os critérios e parâmetros definidos para o ambiente;
e) gerenciar a identificação e localização de usuários em ambientes heterogêneos;
f) gerenciar os diferentes tipos de perfis dos usuários, objetos e demais necessidades que compõem os ambientes;
g) definir outros mecanismos relacionados ao middleware responsável pelos controles e gerenciamento de regras e critérios relacionados ao histórico de uso do ambiente.

1.4 Problema de Pesquisa

Além dos diversos aspectos que abrangem o controle e gerenciamento da privacidade de ambientes ubíquos, há também a necessidade do tratamento das características relacionadas aos dispositivos presentes nestes ambientes e principalmente aos usuários que o frequentam. Diante disso, propõe-se inicialmente a divisão do espaço em que os ambientes se integram, dividindo os em três grandes regiões definidas como: pública, privada e restrita, onde cada região possui diferentes níveis de acesso. Por exemplo, regiões privadas possuem diferentes níveis com refininamentos de uso especificos, que são divididos em critérios e definições relacionadas e específicas a cada área e/ou região. A partir dessas definições foram elaborados experimentos e testes que se mostraram promissores. Com base em especificidades para o tratamento de diferentes perfis, inicialmente definidos como: administrativo, avançado, básico, convidado e bloqueado em um mesmo ambiente, surgindo alguns desafios.

Esses desafios foram desenvolvidos e tratados com a definição de uma taxonomia de privacidade, definição de um modelo de middleware de privacidade e a prototipação e testes em um ambiente real. Contudo, existem outras definições para o problema que também residem no tratamento dos dados de privacidade individuais, que podem variar de acordo com as definições de cada ambiente, visto que são muitas variáveis a serem tratadas que se alternam de acordo com as definições, regras e critérios definidos.

1.5 Organização do Texto

A organização deste trabalho está dividida em capítulos para melhor entendimento do leitor, o texto procura descrever as questões e problemas de pesquisas identificados. Além do

presente capítulo introdutório esta tese está estruturada da seguinte forma: O Capítulo 2 apresenta o aporte e a fundamentação teórica com a devida taxonomia e definições de privacidade em ambientes pervasivos e ubíquos. Igualmente, são avaliados os trabalhos relacionados, encontrados na literatura até o presente momento abordando as questões de pesquisa desta tese. No capítulo 3, formula-se a tese, a partir da visão do contexto e cenário atual. Os critérios para a criação do modelo de privacidade são descritos e definidos em uma arquitetura de middleware para ser aplicado na solução do problema de pesquisa relacionado a privacidade em ambientes pervasivos e ubíquos. Por fim, a descrição de um modelo genérico de privacidade, com estudo de caso e aplicação da arquitetura é aqui apresentada. No capítulo 4, detalha-se o protótipo, a arquitetura e seu funcionamento. Os resultados são avaliados e relacionam-se os trabalhos futuros necessários à continuidade das pesquisas. No capítulo 5 são apresentadas as contribuições científicas acadêmicas, os grupos e projetos de pesquisa. A disseminação do conhecimento científico é descrita pelas publicações de artigos, livros, trabalhos publicados em simpósios, congressos e conferências da área. Também são apresentadas outras contribuições como orientações de graduação e pós–graduação ocorridas ao longo do desenvolvimento deste trabalho. O planejamento e cronograma para a finalização desta tese e metas a serem alcançadas são descritas ao final deste capítulo. E por fim, no Capítulo 6 são descritas as referências bibliográficas utilizadas nesta tese, logo após são apresentados os anexos parcialmente relacionados aos testes e resultados obtidos.

2 APORTE TEÓRICO E FUNDAMENTAÇÃO

Este capítulo tem por objetivo apresentar a fundamentação teórica com pesquisas contendo os principais conceitos que envolvem o controle e gerenciamento de privacidade em ambientes ubíquos. Em seguida, apresentam-se os trabalhos relacionados, bem como os parâmetros e definições necessários a fim de posicionar a tese na literatura atual. Por fim, descreve-se uma análise crítica e comparativa dos trabalhos relacionados com as definições sobre a privacidade em ambientes ubíquos elencando os pontos em aberto e as contribuições científicas da tese.

2.1 Fundamentação Teórica

Esta seção apresenta os principais conceitos abordados na tese sobre controle e gerenciamento de privacidade em ambientes ubíquos. Apresenta, ainda, alguns trabalhos relevantes no âmbito em que essa tese se apresenta. Em ambientes pervasivos são diversos os desafios encontrados e problemas a serem tratados, destacando-se o controle e gerenciamento da privacidade. Podemos citar os conceitos definidos por Langheirinch (2002), que descreve que a privacidade está ligada intrinsecamente à percepção de cada indivíduo sobre o que representa, como por exemplo, uma ameaça na sua propriedade pessoal ou integridade física ou moral. Sendo assim, infere-se que a definição de privacidade é algo muito abstrata e subjetiva que toma forma nas mais diversas necessidades particulares de cada indivíduo Leithardt et al. (2013a). Tais necessidades não são homogêneas e podem ser dependentes tanto de aspectos culturais como religião, tradição, costumes, educação ou política, como de questões mais subjetivas ligadas a intimidade do usuário ou ao seu contexto corrente, tais como idade, estado de saúde, função no trabalho, humor, atividade, dentre outras.

Apesar da extensa a pesquisa na área conforme tabela 2.1, onde se apresenta um estudo comparativo, muitas questões continuam em aberto ou exigem um grande esforço para integração em uma única solução que trata do desafio relacionado à privacidade desses ambientes e conceitos. Não é difícil perceber que não se pode enumerar todos os aspectos relevantes em todas as situações, o que invalida as definições específicas de contexto conforme descreve Ioannis Krontiris, Tassos Dimitriou (2015). Os dados que caracterizam um contexto podem surgir tanto do mundo físico como do mundo virtual, e algumas vezes se misturam. Uma maneira de proteger a privacidade é fazer upload de dados sobre armazenamentos de dados pessoais, que são propriedade e controlados pelos usuários,

permitindo-lhes fiscalizar e limitar a divulgação de dados pessoais e exercer o controle de acesso aos seus dados.

As pessoas normalmente não percebem ambientes físicos (escritório, piso de uma loja, estádios) e ambientes virtuais (*desktop* de um computador, funcionalidades de um telefone celular) como partes separadas, já que objetos e processos podem ser representados nos dois universos. Assim, se faz necessário projetar estruturas capazes de representar elementos tanto do mundo real como virtual conforme descreve Rolim et al. (2015). Esses objetos podem ser elementos físicos ou apenas conceitos, como do mundo virtual, da maneira mais genérica possível, de tal forma que possibilite a criação de ambientes que suportem melhor as atividades físicas e virtuais relacionadas. No decorrer desta tese serão abordados outras contribuições e comparativos, descrita na seção trabalhos relacionados onde também serão apresentados o cenário atual relacionado ao tratamento de contexto e outras pesquisas e respectivos trabalhos que estão sendo desenvolvidos na área.

2.2 Trabalhos relacionados

Sensibilidade ao contexto, segundo Dey (2000), é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Entidade é uma pessoa, objeto ou local que é considerado relevante para a interação entre um usuário e a aplicação. Segundo Chen, Finin e Joshi (2003) foram definidos quatro tipos de contexto: Contexto da Computação (redes e recursos), Contexto do Usuário (pessoas, lugares e objetos), Contexto Físico (luz, odor, temperatura) e Contexto Temporal (hora, dia, mês). Um exemplo de contexto é a capacidade de um dispositivo detectar a temperatura em um determinado ambiente e atuar sobre equipamentos (condicionadores de ar) para fornecer o valor ideal de temperatura para os usuários.

Outra definição é a de Satyanarayanan (2001), que diz que o contexto de um usuário de uma aplicação sensível ao contexto consiste de atributos como localização física, estado fisiológico, estado emocional, histórico pessoal, padrões diários de comportamento, entre outros, que se fornecidos para um assistente humano, podem ser usados para tomada de decisões sem necessidade de interromper o usuário a todo o momento. Entretanto, há dois desafios principais relacionados ao desenvolvimento e uso de aplicações sensíveis ao contexto: a complexidade em desenvolver os serviços de provisão de contexto e a necessidade de manter a privacidade da informação de contexto (e.g., localização) do usuário. Segundo Yitao Duan and John Canny (2006), normalmente estas aplicações usam o contexto

computacional (e.g., nível de energia, largura de banda), pessoal (e.g., perfil, localização do usuário) ou físico (e.g., temperatura, umidade) para oferecer serviços customizados ou mais adequados ao usuário final.

Na Europa já existem diretivas e leis direcionadas para o controle e gerenciamento de privacidade dos dados pessoais dos usuários, por exemplo, em um serviço de *roaming* global para telefones móveis, em alguns países, já existe políticas legislativa com foco na privacidade pessoal. A *European Union Directive on Data Protection* DIGITAL AGENDA FOR EUROPE (2014), que compreende o conjunto de leis de privacidade mais completo na atualidade, no entanto são diretrizes e descrições que podem não aplicadas em todas as áreas, ambientes, usuários e principalmente países.

Com isso, vários trabalhos se fundamentam nesse conjunto de leis Europeias para definirem e organizarem suas normas e regras, porém, pesquisadores como Esquivel et al., (2015) e anteriores definiram politicas e regras com base em estudos e comparações. No trabalho descrito por Rodrigues (2006), são descritas algumas definições e políticas de controle e gerenciamento de privacidade, dentre as quais:

a) a privacidade deve ser obtida de forma correta e dentro das condições impostas pela lei;
b) deve ser usada somente para o propósito original especificado;
c) deve ser requisitada de forma adequada e relevante ao propósito original, ou seja, a precisão da informação requerida não deve ser mais específica do que o necessário para atender a necessidade do requisitante;
d) deve ser mantida de forma segura;
e) deve estar acessível ao dono da informação.

Entretanto, algumas diretivas foram adicionadas desde então, destacando a proteção pública, que estabelece um quadro legal obrigatório que garante o direito individual à privacidade DIGITAL AGENDA FOR EUROPE (2014). Tal direito é possível através da concretização de medidas que devem ser respeitadas por qualquer organização (incluindo governos e empresas) que lida com dados pessoais durante a concepção e utilização de processos de dados. Esse direito abrange o tratamento de dados pessoais e a proteção da privacidade, incluindo disposições relativas:

a) à segurança das redes e serviços;
b) à confidencialidade das comunicações;
c) ao acesso aos dados armazenados;
d) ao processamento de dados de tráfego e localização com identificação;

e) ao controle pessoal e público de listas de assinantes e comunicações comerciais não solicitadas.

Na literatura diversos trabalhos são focados em tratamento de contexto do usuário, dispositivos, nesta proposta o tratamento de contexto é direcionado ao ambiente, ou seja o ambiente é quem dita às regras de acordo com seu contexto vigente. No entanto, não serão discutidas técnicas ou demais formas de tratamento de contexto, o mesmo será apenas utilizado para fundamentar as necessidades de controle e gerenciamento de privacidade em ambientes ubíquos.

Dentre os trabalhos analisados, os mesmos focam nos dispositivos, bem como os seus meios de comunicação, ou focam na relação de controle e gerenciamento que é necessária para privacidade, entretanto, não tratam diretamente do controle e gerenciamento de privacidade no âmbito dos quesitos e relações do ambiente ubíquo em si. Portanto, definimos como principal contribuição deste trabalho o fato de se relacionar diretamente ao tratamento das características do ambiente, seus compartilhamentos e informações que serão disponibilizadas ao usuário. Este por sua vez terá de se adaptar ao contexto do ambiente e não aos usuários e dispositivos a sua volta, com isso pretende-se adaptar o contexto ubíquo no mundo real no qual vivemos e que muitas vezes não mudam, ou podem ser mudados por conta de controles de privacidade fundamentados no usuário ou seus dispositivos. Para tanto, propomos como contexto do ambiente ubíquo como quem define as regras, critérios e parâmetros de funcionamento, ou seja, dispositivos, usuários, comunicações etc. é que se adaptam as regras e definições do ambiente e não o contrário.

Com base nessas premissas, alguns trabalhos foram pesquisados, dentre os quais se destaca o de Beresford e Stajano (2003), que calcula a localização do usuário baseando-se no uso de GPS. A abordagem proposta calcula possíveis pontos dentro de um edifício, por exemplo, onde determinado usuário poderá estar, com isso insere regras de privacidade para o usuário dependendo de sua provável localização. O sistema proposto apesar de suas limitações à época, não trata, por exemplo, de serviços relacionados ao usuário, visto que não tem uma localização correta do mesmo, tão pouco trata do compartilhamento de dados relacionados a uma área de maior porte, uma vez que o GPS pode ter alguma discrepância mesmo de centímetros, que dependendo da aplicação pode parecer pouco, mas no caso da divisão de uma sala, uma simples parede fina poderá fazer uma diferença entre um ambiente e outro.

Segundo Zhu, Mutka e Ni (2005) o uso de protocolos de descoberta de serviços são projetados para minimizar a sobrecarga administrativa e aumentar a usabilidade. Eles também

podem prever e codificar todas as interações possíveis dos estados entre os dispositivos e programas em tempo de *design*. Ao adicionar uma camada de controle, protocolos de descoberta de serviços buscam simplificar o desempenho do sistema, o trabalho apresenta definições taxonômicas de comunicação e serviços. Entretanto, a solução proposta é focada no controle de protocolos e dados gerados pelo dispositivo relacionado ao usuário, não apresentando definições e descrições direcionadas ao controle do ambiente ubíquo. Em Ioannis Krontiris, Tassos Dimitriou (2015), é apresentado uma proposta sobre privacidade com uma solução que utiliza agentes móveis remotos para busca e tratamento de dados de usuários, com isso a proposta prove maior segurança e privacidade de acordo com a localização, no entanto, o foco do trabalho é confiabilidade dos dados, não trata especificamente dados do ambiente.

Para que melhor possam ser compreendidos os trabalhos pesquisados, apresentamos a seguir uma comparação entre o estado da arte e o modelo proposto conforme a Tabela 2.1.

2.3 Comparação entre Trabalhos Relacionados

A Tabela 2.1 apresenta, de maneira sumarizada, os requisitos identificados para elaboração dos critérios necessários para controle e gerenciamento de privacidade em ambientes ubíquos. Tais requisitos foram apresentados na taxonomia proposta que considerou a literatura pesquisada conforme Leithardt et al (2013b). Nesses trabalhos, os requisitos possuem exaustivos tratamentos, com definições de taxonomia individuais, conforme apresentado na Figura 3.1. Com base no estudo realizado e na comparação entre o estado da arte, definimos um modelo de privacidade para ambientes pervasivos/ubíquos denominado UbiPri e que será apresentado no próximo capítulo. Na tabela elaborada serão usadas as seguintes definições:

a) Aborda (Aborda): o trabalho trata do requisito abordado;
b) Não Aborda (Não Ab.): o trabalho não trata do requisito abordado;
c) Não Descrito (Não Desc.): não foi encontrada informação sobre o requisito abordado;
d) Em Desenvolvimento (Em Dev.): o requisito ainda está sendo desenvolvido, geralmente é apontado com frequência em testes, validações, resultados obtidos ou trabalhos futuros.

Tabela 2.1 - Comparativo estado arte - privacidade em ambientes ubíquos

Abordagem Solução	***Usuário***	***Dispositivo***	***Aplicação***	***Serviços***	***Comunicação***	***Ambiente***
Saha, Mukherjee (2003)	Aborda	Não Ab.	Aborda	Não Ab.	Não Ab.	Não Ab.
Zhu et al. (2005)	Aborda	Não Ab.	Aborda	Aborda	Em Dev.	Não Ab.
Rodrigues (2006)	Aborda	Não Ab.	Aborda	Aborda	Aborda	Não Ab.
Kagal et al. (2001)	Em Dev.	Aborda	Aborda	Aborda	Não Ab.	Não Desc.
Kalempa, Sobral (2009)	Em Dev.	Não Ab.	Aborda	Aborda	Aborda	Não Desc.
Mattes et al. (2012)	Aborda	Em Dev.	Aborda	Aborda	Aborda	Em Dev.
Gorlach et al. (2005)	Aborda	Aborda	Aborda	Não Ab.	Em Dev.	Não Ab.
Montserrat et al. (2011)	Aborda	Aborda	Em Dev.	Aborda	Aborda	Não Ab.
El Defrawy e Tsudik (2011)	Não Desc.	Não Ab.	Em Dev.	Em Dev.	Aborda	Não Ab.
Leithardt, hessel (2008)	Aborda	Aborda	Aborda	Aborda	Não Ab.	Em Dev.
Campbell et al (2003)	Não Ab.	Aborda	Aborda	Aborda	Não Ab.	Não Ab.
Henricksen et al (2005)	Aborda	Não Ab.	Aborda	Em Dev.	Aborda	Não Ab.
Krumm et al (2008)	Não Ab.	Não Desc.	Em Dev.	Em Dev.	Aborda	Não Ab.
Moschetta et al (2008)	Em Dev.	Não Ab.	Aborda	Aborda	Aborda	Não Ab.
Lehikoinen et al (2008)	Em Dev.	Não Ab.	Aborda	Aborda	Aborda	Não Ab.
Lupiana et al (2009)	Aborda	Aborda	Em Dev.	Aborda	Aborda	Não Desc.
Bardram et al (2003)	Aborda	Em Dev.	Aborda	Aborda	Aborda	Não Ab.
Iachello, Hong (2007)	Aborda	Aborda	Aborda	Não Ab.	Aborda	Não Ab.
Krontiris e Dimitriou (2015)	Aborda	Aborda	Em Dev	Não Ab.	Aborda	Em Dev.
UbiPri	Aborda	Aborda	Aborda	Aborda	Aborda	Aborda

Fonte: Adaptado de Leithardt, et al. (2013b).

Diante do cenário atual, não foi encontrado na literatura pesquisada um modelo abrangente para controlar e gerenciar a privacidade com foco em ambientes ubíquos, tendo como premissa os parâmetros e definições que esses necessitam tão pouco foi encontrado uma definição taxonômica que abranja as mesmas necessidades conforme define Marc Langheinrich (2001) que descreve a privacidade a partir de três ângulos: a sua história, o seu estatuto jurídico, e sua utilidade. Segundo Yitao Duan and John Canny (2006) Antes de projetar qualquer esquema de proteção, precisamos determinar quem deve ter o direito de acesso aos dados e em que condições, e isso vai ser drasticamente moldado por considerações de privacidade. Hübner (2012) apresenta três classificações para privacidade, são elas pessoal, territorial e da informação. No trabalho de Warren & Brandeis (1890) que é considerado como o primeiro trabalho publicado sobre o tema, define como privacidade o direito “de ser deixado só”.

Portanto, conforme a Tabela 2.1 os trabalhos pesquisados na literatura descrevem que determinada solução se aplica ao ambiente ubíquo, mas não relatam como proceder em detalhes para controlar e/ou gerenciar essas funcionalidades, apenas vislumbram definições de como seria. A descrição de informações detalhadas, testes, resultados ou simulações que comprovem que o ambiente ubíquo gerencie de forma onipresente não é trivial, nesse

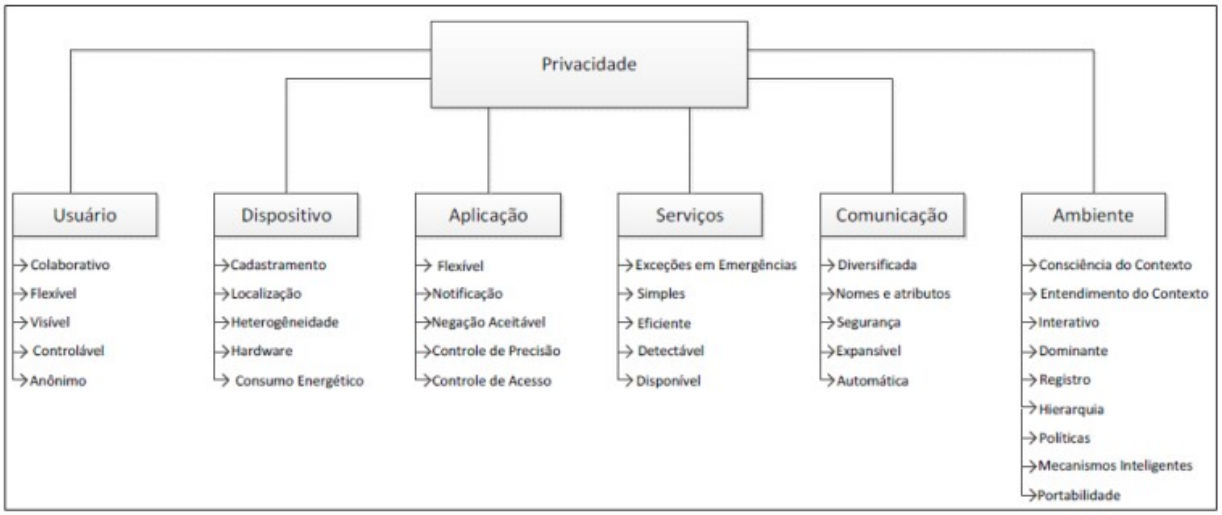
contexto essa tese contribui apresentando um protótipo conforme a literatura e definições ao longo do texto. No entanto, se faz necessário elaborar e utilizar definições taxonômicas que possibilitem direcionar os parâmetros e quesitos relacionados, tais conceitos serão apresentados na próxima seção.

Portanto, a continuidade das pesquisas nessa linha se mostra promissora e indica forte contribuição científica para o proponente e demais pesquisadores da área que poderão utilizar os resultados obtidos nesta tese para futuras pesquisas e comparações em privacidade.

3 TAXONOMIA DE PRIVACIDADE EM AMBIENTES UBÍQUOS

Entre os trabalhos de pesquisa que abordam a privacidade, foram consultadas referências que desenvolviam suas próprias taxonomias para tratamento de usuários, dispositivos, aplicações e comunicação, destacando-se o uso de protocolos, tratamento de serviços e ambientes ubíquos. Com isso, elencamos nesta seção as principais contribuições dos trabalhos pesquisados em Leithardt et al, (2013a) Leithardt et al. (2013b) e Leithardt et al. (2014) para descrever e definir uma taxonomia para controle e gerenciamento de privacidade a fim de propor parâmetros e itens necessários para utilização em ambiente ubíquo. A Figura 3.1 apresenta a taxonomia proposta, resultante da pesquisa realizada.

Figura 3.1 - Modelo taxonômico de privacidade em ambientes ubíquos



Fonte: Leithardt et al (2013b).

A taxonomia foi dividida em 6 grupos com os critérios para a composição definidos de acordo com a literatura pesquisada de acordo com a Tabela 2.1, e para utilização em ambientes ubíquos, são eles: Usuário, Dispositivo, Aplicação, Serviços, Comunicação e Ambiente. Cada grupo possui características específicas para que possa ser empregada conforme necessidade, como por exemplo: o Usuário precisa ser colaborativo para que haja interesse entre os demais, flexível para que se possa haver troca de informações, visível aos demais do mesmo ambiente, controlável para que haja possibilidade de sobrepor suas próprias preferências e anônimo em determinadas situações ou necessidades onde a privacidade assim determinar. O grupo de aplicação trata sobre questões relativas ao funcionamento, controle e gerenciamento da aplicação, diferente do grupo responsável pelos serviços disponibilizados.

3.1 Quesitos taxonômicos usuário

Alguns quesitos importantes foram apresentados por Moschetta et al. (2008), onde também consta que o usuário no ambiente ubíquo deve estar de acordo com os pontos elencados a seguir:

a) colaborativo: o usuário deve disponibilizar acessos de forma colaborativa para enriquecer de informações os demais usuários e a si próprio;
b) flexível: os usuários podem ajustar o grau de colaboração em função dos graus de segurança exigidos para determinado serviço ou consulta, com isso há flexibilidade entre os usuários e suas fontes de informação;
c) visível: o usuário disponibiliza seu perfil e sua identidade que podem ter classificações como descrito por Moschetta et al. (2008): a identidade pode ser fraca, com grau mínimo de confiabilidade, média com grau médio de confiança e forte com alto grau de confiabilidade;
d) controlável: em relação ao controle de compartilhamentos pelo próprio usuário, o mesmo poderá decidir mudar de opiniões, características e demais dados que possam ser controlados pelo mesmo, fundamentando-se no mundo real e em suas decisões;
e) anônimo: outras características também devem ser levadas em consideração, dentre as quais o anonimato, visto que por diversas situações o usuário não pode ou não deve ser localizado, por estar em ambientes que não possam ser divulgados, por sua condição de trabalho ou outras situações diversas.

No trabalho desenvolvido por Montserrat Ros et al. (2011) é realizado o controle de transferência de arquivos de música, por exemplo, com base na localização de pontos Wi-Fi. O trabalho expõe uma percepção diferente dos estudos apresentados, visto que os autores descrevem ser possível definir tipos e tamanhos a serem transferidos de acordo com o ponto e localização, que é um item do grupo abordado na taxonomia apresentada anteriormente. Porém, não se trata do ambiente em si e sim nos pontos de acesso ao mesmo. No entanto, Krumm (2009) descreve outra solução baseada em algoritmo que calcula a área de acordo com a proteção de dados exigida. Tal abordagem utiliza como cenário um hospital, e trata também da localização do usuário compartilhando ou não informações. Entretanto, não há interação com o ambiente ubíquo e sim com a localização do mesmo. O trabalho também aborda outras formas que poderiam restringir o acesso a determinadas informações como, por exemplo, a validação de documentos. Com relação às características desejáveis para

tratamento de privacidade em dispositivos, alguns trabalhos tratam da localização usando GPS para detecção dos usuários. Nesses trabalhos são relacionados pontos de acesso Wi-Fi, ou antenas de sinal de celular para obter coordenadas para localização e consequentemente prosseguimento aos demais serviços fundamentados na localização do usuário no ambiente ubíquo.

3.2 Quesitos taxonômicos do dispositivo

Na pesquisa elaborada por Görlach, Heinemann e Terpstra (2005), os pesquisadores definem como necessário para controlar e gerenciar dispositivos, os seguintes quesitos taxonômicos:

a) cadastramento: o cadastramento é necessário para realizar o simples controle e organização de possíveis locais de acesso com maior frequência de uso do dispositivo, a fim de diminuir o tráfego de dados desnecessários e realizar as identificações com maior rapidez, como por exemplo, nos locais onde o dispositivo já tenha sido cadastrado são armazenadas as características básicas como *login* e serviços;
b) localização: na localização é necessária uma base de dados no dispositivo onde deve constar todos os dados do usuário. Esta base de dados é acessível somente para atualização e validação de algumas informações, o tratamento das demais informações deve ser feito pelo ponto onde o usuário acessou, a fim de prover uma maior segurança e confiabilidade, como uma central de dados para manter atualizados os dados mais acessados e utilizados;
c) heterogeneidade: uma heterogeneidade tanto de protocolos de comunicação tratados pelo dispositivo, quanto da quantidade de serviços e tipos disponibilizados;
d) hardware: um dos quesitos de grande relevância que deve ser considerado no tratamento da privacidade relacionada aos dispositivos, no que se refere ao poder computacional do hardware utilizado conforme descreve Lupiana, O'Driscoll e Mtenzi (2009), é o poder de processamento e armazenamento. O hardware não deve ser tratado como algo depreciativo e sim como objetivo e meta a ser alcançada no tocante a igualdade de recursos;
e) consumo energético: é necessário considerar, de acordo com o hardware utilizado, o consumo energético para aplicação, serviços e comunicação. O sistema ubíquo

como um todo deve sempre levar em consideração a elaboração de aplicações que visam à redução de consumo energético dos dispositivos. Para tanto, deve-se buscar o processamento externo das informações, cabendo ao dispositivo apenas informá-las. No trabalho descrito por Beresford e Stajano (2003), é apresentada uma solução que localiza e controla a privacidade em regiões, dentro dessas regiões há microrregiões que podem ser capazes de realizar vários processamentos distribuídos, dividindo tarefas dos dispositivos e tarefas.

3.3 Quesitos taxonômicos de aplicações

As definições de taxonomia com as definições relacionadas aos grupos de aplicação são fundamentadas no trabalho descrito por Rodrigues (2006), conforme apresentado na Tabela 2.1, onde foram propostos os quesitos desejáveis para aplicação e serviços de privacidade. dentre os quais:

a) flexível - os usuários devem ser capazes de definir suas preferências de privacidade com diferentes níveis de detalhes para diferentes grupos de requisitantes. Em diferentes grupos de usuários ou locais em que o usuário se encontra poderão ter compartilhamentos diferentes baseados no público que o mesmo se relaciona no momento, como por exemplo, usuários de uma religião e esportes terão informações distintas a serem compartilhadas;
b) notificação - os usuários podem ser notificados sobre, ou estarem aptos a rastrear, qualquer tentativa de acesso às suas informações de contexto. Com isso, é possível que o usuário gerencie as notificações em diferentes situações. Por exemplo, em determinados momentos é necessário receber avisos constantes, ou ainda em outros momentos os avisos devem ser descartados automaticamente, tal como, em horário de sono que pode não ser o mesmo todos os dias;
c) negação aceitável - além das opções de controle de acesso *Grant*" e *Deny*", uma terceira opção ao - *Not Available*" - deve ser oferecida. A partir desta opção, os usuários podem negar o acesso sem que os requisitantes tenham conhecimento das suas ações. Este artifício também é conhecido como *plausible deniability*;
d) controle de precisão - os usuários podem ajustar a precisão temporal e espacial de suas informações de contexto. Aplica-se geralmente na mobilidade do usuário, sua disponibilidade de agenda e atribuições do dia a dia, onde as informações podem mudar constantemente;

e) controle de acesso - a qualquer momento, os usuários podem bloquear o acesso a qualquer (ou a todas) informação de contexto. Por questão básica de segurança, o próprio sistema poderá avisar que o usuário ou seu dispositivo está sobre possíveis ameaças e bloquear qualquer tipo de informação. Pode também ocorrer de o usuário se encontrar em locais públicos desconhecidos onde não seja recomendável a disponibilização de informações.

3.4 Quesitos taxonômicos de serviços

Segundo Kagal, Finin e Joshi (2001), tradicionalmente os computadores autônomos e pequenas redes dependem de autenticação do usuário e controle de acesso para garantir a segurança. Estes métodos físicos usam o sistema com base em controles para verificar a identidade de uma pessoa ou processo, com isso, podem permitir ou restringir a capacidade de utilizar, alterar ou visualizar um serviço ou recurso de computador, com isso se faz necessário definir os quesitos mínimos necessários para o modelo taxonômico, dentre os quais:

a) exceções em emergências - os usuários podem definir políticas de exceções que tenham precedência maior do que qualquer outra política de privacidade. Como no mundo real existem fatores que fogem totalmente do nosso controle, para esses casos é necessário definir uma política que controle essas situações, como a ligação urgente de um familiar em momento inoportuno ou em local definido como restrito a ligações;
b) simples - os usuários não devem ser sobrecarregados com a configuração das suas preferências de privacidade. Por questão básica de usabilidade, ninguém pretende ter que ficar abrindo diversos controles para configurar determinada funcionalidade. Para isso é necessário que o próprio sistema armazene, por exemplo, quais as funções mais acessadas, mais úteis e assim sucessivamente;
c) eficiente - o tratamento das questões de privacidade não deve causar um atraso significativo na comunicação ou uma carga de processamento excessiva para os serviços de provisão de contexto, com isso o gerenciamento de regras e definições de critérios se torna ainda mais complexo.

Os usuários móveis pretendem acessar seus recursos hospedados localmente e serviços a qualquer hora e em qualquer lugar, levando a sérios riscos de segurança e problemas de controle de acesso. Com base nessas características, Kagal, Finin e Joshi (2001) propõem uma solução baseada em gerenciamento de confiança que envolve o desenvolvimento de uma

política de segurança, bem como a atribuição de credenciais a entidades, verificando que as credenciais possam cumprir a política, delegando a terceiros a confiança. A solução proposta é válida, porém, não trata de todos os quesitos necessários abordados na taxonomia apresentada na Figura 3.1, tão pouco foca no ambiente ubíquo. No trabalho descrito por Zhu, Mutka e Ni (2005), são apontadas outras características que são utilizadas na taxonomia de privacidade relacionada aos serviços e as aplicações, são elas:

a) detectável: a aplicação deve implementar requisitos e parâmetros para descoberta de serviços disponíveis e disponibilizá-los aos usuários para que os mesmos utilizem. Essa utilização e disponibilização devem ser de forma onipresente e automática, de tal forma que não seja necessário reconfigurar a cada nova situação;
b) disponível: a aplicação deve ter controle sobre a utilização dos serviços disponíveis, assim, outros usuários, dispositivos, comunicações, serviços e ambiente ubíquo como um todo poderão usufruir de uma quantidade maior de informações;

No trabalho desenvolvido por Zhu, Mutka e Ni (2005) também é apresentada uma taxonomia de comunicação fundamentada em protocolos de comunicação e serviços para ambientes ubíquos. O trabalho apresenta ainda diversos problemas encontrados para realização do tratamento de segurança voltado para aplicações e serviços ubíquos, porém o que mais se destaca são as definições de protocolos e serviços. Para elaboração das características relacionadas a esse grupo da taxonomia de privacidade foram elencados as seguintes especifidades:

a) diversificada: a comunicação deve ser adaptável ao maior número possível de dispositivos e ambientes e estes devem estar preparados para troca de dados utilizando diferentes formas de comunicação sem intervenção do usuário;
b) nomes e atributos: com base em nomes e atributos definidos para usuários e ambientes são relacionados os critérios de utilização de serviços e atribuições de acordo com ao perfil definido;
c) segurança: a infraestrutura deve ser capaz de tratar de funções de performance, controle de certificações, *login*s e outros gerenciamentos necessários para prover comunicação ubíqua de forma segura, tendo maior confiabilidade, conforme descreve El Defrawy e Tsudik (2011);
d) expansível: a comunicação deve ter capacidade de gerenciar um mesmo protocolo que pode e deve ser usado por mais de um usuário, igualmente o mesmo protocolo

deve prover serviços em vários ambientes com qualidade satisfatória, atendendo a maior quantidade de usuários e dispositivos simultaneamente;

e) automática: A comunicação deve possuir mecanismos para tratamento de mobilidade e adaptabilidade utilizando canais de comunicação *unicast*, *broadcast* ou *multicast* sem a intervenção do usuário.

Em relação às **descrições taxonômicas de ambientes ubíquos**, as definições taxonômicas foram elaboradas e fundamentadas conforme a literatura pesquisada apresentada na Tabela 2.1 e relacionada à consciência, tratamento e gerenciamento de contexto, no entanto, a contribuição cientifica da tese não possui como objetivo principal o tratamento de perfil e contexto de usuário, mas utiliza-se destas áreas para fundamentar a necessidade de controle e gerenciamento de privacidade em ambientes ubíquos. A tese também não trata de controle de contexto, no entanto, utiliza as informações contextuais e definições de parâmetros para realizar o controle e gerenciamento de privacidade em ambientes ubíquos. Com isso, verificou-se que no trabalho desenvolvido por Costa, Yamin e Geyer (2008), apresenta uma estrutura de *Framework* e *Middleware* para computação ubíqua, ainda, segundo os autores existem duas exigências fundamentais na computação ubíqua relacionada à integração física e espontânea desses ambientes que é a segurança e privacidade dos dados.

No trabalho descrito por Campbell et al. (2003), são apresentadas formas de comunicação e infraestrutura que compreendem vários quesitos e características necessárias para esses ambientes, tais como: escalabilidade, heterogeneidade, integração, invisibilidade, consciência e gerenciamento de contexto como o principal desafio a ser tratado na computação ubíqua. No entanto, os trabalhos preocupam-se em focar nos quesitos de infraestrutura de *software* da computação ubíqua no âmbito geral e não especificamente no ambiente computacional ubíquo.

O trabalho descrito por Kalempa e Sobral (2009) trata sobre uma modelagem de sistemas na área de computação ubíqua. Mais precisamente, contém um estudo sobre a questão da privacidade, como forma de estender outro trabalho de pesquisa, o qual trata da formalização de um metamodelo a ser utilizado como base para a construção de sistemas ubíquos. A extensão proposta também aborda a questão da privacidade em nível de usuário de um sistema ubíquo e tem como objetivo a especificação da privacidade para estes ambientes. Apesar das várias contribuições com relação às características pesquisadas e os tópicos abordados no estado da arte, o trabalho não trata diretamente da privacidade do ambiente ubíquo e sim uma abordagem focada no usuário do ambiente. Fundamentado nessas

abordagens podemos elencar alguns quesitos necessários para ambientes ubíquos, destacando-se a consciência do contexto que é um dos parâmetros mais importantes em ambientes inteligentes, que torna um sistema de computação ubíquo invasivo o mínimo possível. Ou seja, o sistema e o ambiente devem ser capazes de conhecer o estado do usuário e do seu ambiente, e alterar seu comportamento baseado nesta informação, conforme descreve Satyanarayanan (2001). Por exemplo, um usuário que ao adentrar em uma sala é automaticamente reconhecido pelo ambiente e este disponibiliza seus serviços e demais configurações que são atribuídas ao usuário. Esquivel et al. (2015) descreve uma solução metafórica para gerenciamento de informações privadas no usuário com foco na venda de produtos e serviços de comércios, no entanto o trabalho foca nas informações do usuário e não diretamente no que o ambiente tem a oferecer e com isso buscar ou conquistar usuários e/ou clientes com base no que há disponível para oferecer.

No trabalho desenvolvido por Pereira et al. (2011), é apresentado um modelo taxonômico com objetivo de estabelecer um modelo de referência, para permitir a classificação de RSSFs em geral, incluindo aquelas com Qualidade de Serviço (QoS – *Quality of Service*), e auxiliar no desenvolvimento de um novo conjunto de métricas de QoS que caracterizam completamente este novo tipo de rede, porém, não apresenta definições relacionadas com a privacidade do ambiente, sendo o seu foco o controle de métricas de QoS, com base nessas métricas o autor procura atender também quesitos de privacidade individual.

No trabalho desenvolvido por Li et al. (2009) é apresentado um levantamento do estado da arte sobre privacidade relacionado a técnicas para RSSFs com uma taxonomia de controle de privacidade e contexto de dados utilizados nas redes de sensores sem fios. São analisadas duas categorias principais de privacidade de preservação e técnicas para proteger dois tipos de informação privada, orientada a dados e orientada aos contextos de privacidade, respectivamente. O trabalho é interessante visto que é apresentado uma solução para os dados trafegados nas RSSFs, mas não apresentam soluções relacionadas aos ambientes em que essas se encontram, tão pouco foca na privacidade genérica, objetivando sempre as RSSFs. Segundo descrevem Kalempa e Sobral (2009), a troca de informações e o contexto que são identificados por RSSFs, RFID, entre outros, é de natureza das atividades humanas é um ramo de pesquisa muito importante para muitas áreas, tais como, psicologia, sociologia, e ergonomia. Essa variedade de perspectivas gera vários problemas, visto que, cada área pode explorar táticas diferentes para entender as ações humanas. O entendimento destas diferentes áreas é utilizado também na computação ubíqua com a finalidade de representar atividades humanas, e pode resultar em um desafio de multidisciplinaridade significante. Com isso faz-

se necessário o acoplamento de vários outros sistemas para suprir de informações ao ambiente ubíquo conforme apresentado na Figura 3.1, e consequentemente como o mesmo deve proceder para realizar tarefas, funcionalidades onipresentes, para tanto alguns quesitos foram elencados para melhor entendimento do ambiente, são eles:

a) consciência do contexto: descreve que o ambiente deve permitir que o usuário pudesse compartilhar seus próprios dados e também utilizar informações e serviços disponibilizados pelo ambiente e os demais usuários que nele se encontram. Dessa forma, espera-se que o ambiente suporte a representação de serviços e informações, independente de domínio (Considera-se domínio, o controle e gerenciamento que o usuário exerce sobre o ambiente em que se encontra). Com o auxílio do ambiente, o usuário poderá escolher os serviços que mais se identificam com seus objetivos e se adaptam ao seu contexto. Os serviços devem estar acessíveis de qualquer lugar, disponíveis em vários formatos, considerando a heterogeneidade de dispositivos e demais definições.
b) entendimento do contexto: seres humanos se adaptam a mudanças no contexto de um dispositivo, pessoas, ambientes entre outras características do dia a dia, embora algumas pessoas precisem mais informações outras conseguem captar mais rapidamente mudanças. Para os dispositivos de detecção como RFID, RSSFs, entre outros dispositivos a identificação dessas mudanças também precisam de dados, e quanto maior a quantidade maior certeza do entendimento das ações haverá.
c) interativo: os usuários precisam interagir com o ambiente, a fim de obter informações sobre os mesmos. Tal interação deve ser útil e agradável, deve ser ajustada ao contexto do ambiente sem ser desagradável. Ninguém quer ficar recebendo informações toda vez que passar em frente a uma loja, por exemplo. Uma forma de evitar esse problema é avisar somente os usuários cadastrados, com opção de escolhas de avisos.

No trabalho desenvolvido por Kagal, Finin e Joshi (2001) é apresentada uma solução distribuída de segurança em ambientes ubíquos que requer um tipo de autenticação para controle de acesso, após autenticação são disponibilizados serviços essenciais para tratamento e interatividade ubíqua.

a) registro: o controle da privacidade do ambiente ubíquo deve permitir que a tecnologia estivesse muito próxima dos indivíduos, residindo-nos mais diversos cenários reais que possam ser considerados. Para o controle e registro individual no ambiente é necessário que haja informações que descrevam tudo que pertence

ao ambiente, bem como a disponibilidade de acesso, serviços, compartilhamentos, pessoas autorizadas, dispositivos, comunicação e aplicações que poderão interagir. Porém, a regra será sempre com base nas definições do ambiente, sejam regras de serviço, comunicação, disponibilidade e outras que poderão surgir. Portanto é necessário o registro detalhado de cada característica individual do usuário em momentos diferentes para cada ambiente identificado, necessitando assim, o registro de todas as informações que ocorrem no ambiente para controle e gerenciamento;

b) dominante: seguindo este paradigma, a computação deve ser invisível, ou seja, causar o mínimo possível de distração ao usuário, segundo descrevem pesquisas, o ambiente não deve ficar se auto reconfigurando a cada nova solicitação de *login*, ou mudanças de usuários do ambiente. Ou seja, o ambiente vai ser, e deve estar configurado com características únicas inerentes aos seus critérios e definições, como por exemplo: ir ao jogo de futebol e estar entre a torcida adversária, isso seria uma característica de segurança fundamental, visto que a localização do usuário poderá causar sérios problemas ao usuário e ao ambiente onde se encontra. Com base nisso é necessário definir a hierarquia e definições de utilização do mesmo.

c) hierarquia: no ambiente ubíquo, o ambiente virtual deve ter regras definidas tal qual o mundo real, onde independente do ambiente em que se encontre, sempre haverá um proprietário, uma norma definida pelo alto escalão e que deve ser seguida. Ou seja, uma empresa, uma igreja, uma escola, ou ainda nossas casas, possuem uma árvore hierárquica, com base nisso a taxonomia da hierarquia no ambiente ubíquo deve seguir a mesma linha de raciocínio, sempre fundamentada nas regras definidas pelo ambiente e no mundo real;

No trabalho desenvolvido por Henricksen et al. (2005) é relacionado o controle de hierarquia baseado nos fatos e preferências do usuário. É apresentado um modelo de contexto da aplicação que controla os fatos e ocorrências individuais, buscando informações em diversas fontes ubíquas, porém, não trata do controle de privacidade no ambiente ubíquo.

d) políticas: em relação às políticas do ambiente é necessária à habilitação do controle de entrada/saída do usuário em apenas um ambiente por vez, com isso controla-se o compartilhamento de informações e serviços do ambiente em que o usuário se encontra. Por exemplo, uma sala de aula poderá ser configurada para receber o professor como um tipo de usuário avançado e os alunos receberão *status* de

convidados, porém em outro ambiente o mesmo professor poderá não ser o professor e sim aluno, com isso se adquire os mesmos estados dos demais alunos e consequentemente as regras do ambiente, os detalhes de funcionamento serão apresentados no capítulo do modelo proposto;

Na pesquisa desenvolvida por Mattes et al. (2004) é apresentada uma linguagem lógica para expressar políticas de segurança, chamada LEPS, onde se define um modelo de políticas de segurança para os serviços de controle de acesso, autenticação, integridade, privacidade, auditoria e irretratabilidade. A proposta é interessante, porém é focada nos usuários e grupos relacionados. Entretanto, é possível utilizar parte da ideia do modelo hierárquico proposto ao ambiente ubíquo.

a) mecanismos inteligentes: Tais mecanismos devem possuir definições e regras a fim de fazer com que o próprio ambiente tome as decisões de forma inteligente, sem a intervenção humana.

Uma das formas mais utilizadas é o uso de controles de inteligência artificial como o trabalho apresentado por Rolim et al. (2012b), que descreve um sistema de inferência de informações denominado MANFIS, onde é possível ter várias entradas de dados e apenas uma saída como resultado. Na pesquisa desenvolvida por Lupiana, O'Driscoll e Mtenzi (2009) foi realizada uma classificação da taxonomia para diferentes ambientes ubíquos como sendo de duas categorias principais, ambientes interativos e ambientes inteligentes. A taxonomia classifica todos os ambientes ubíquos que facilitam a interação com o usuário em operações diárias e inteligentes. A classificação se fundamenta em aspectos rotineiros do usuário, assim o ambiente possui as informações habituais. A pesquisa descreve que independentemente do ambiente é necessário ter mecanismos de controle capazes de tomar decisões.

Entretanto, não é realizada uma definição concreta da taxonomia exigida nos ambientes ubíquos, utilizando trabalhos e técnicas desenvolvidas por outros pesquisadores para apresentar uma ideia do que seria o ideal. Tão pouco o referido trabalho trata das questões diversas do ambiente ubíquo, apenas concluem que o ambiente pervasivo deve ser interativo para atender aos requisitos ubíquos.

a) portabilidade: no ambiente ubíquo é uma definição muito importante, que não foi tratada em nenhum dos trabalhos pesquisados. Restringir uma aplicação ou serviço a uma única linguagem de programação, sistema operacional ou quais outras formas de utilização no ambiente ubíquo, também pode ser considerado um tipo de não compartilhamento ou imposição.

Os ambientes ubíquos precisam ser controlados para que aceitem ou imponham o uso de determinadas tecnologias, dispositivos ou *software*s de acordo com critérios e definições, portanto é necessários a elaboração de uma taxonomia de acordo com a literatura pesquisada e as necessidades para elaboração e definição do modelo de privacidade. É certo que existem situações em que são necessárias determinadas regras em se tratando dos tópicos abordados anteriormente, mas para esses casos são necessárias definições de uso de soluções genéricas onde o usuário possa utilizar o mínimo possível dos recursos do ambiente.

Lehikoinen e Huuskonen (2008) apresenta um modelo que procura aproximar a privacidade humana do mundo ubíquo fundamentado em pesquisas teóricas das inter-relações pessoais. O trabalho descreve um estado da arte, rico em informações que estão relacionadas à aproximação das pessoas, com isso obtêm-se suas preferências que podem ser com informações obtidas de determinados grupos de usuários, ou informações que elas colaboram entre si. O modelo é baseado em registro de usuários e controle, com base em informações armazenadas, o ambiente corresponde de forma onipresente. A ideia da tese é bem fundamentada, possui vários fatores que contribuem, porém o modelo apresentado não trata do ambiente ubíquo e sim das pessoas que estão a sua volta, com isso o mesmo molda-se de acordo com as preferências dos usuários.

No trabalho descrito por Iachello e Hong (2007) é realizada uma pesquisa sobre várias questões de privacidade abordadas no contexto de interação homem computador (IHC). O trabalho também traz uma perspectiva de vários pontos que devem ser tratados como tendências na área baseando-se em pesquisas realizadas. Como principal contribuição o trabalho aborda a questão da utilização de vários quesitos, dentre os quais a proteção e segurança do ambiente ubíquo. No entanto, o trabalho apenas apresentou superficialmente a perspectiva do controle de privacidade do ambiente, uma vez que o foco do trabalho descrito por Iachello e Hong (2007) não é a privacidade de ambientes ubíquos. Em Bardram, Kjaer e Pedersen (2003) é apresentada uma solução baseada em autenticação que usa vários exemplos como RFID, dentre outros, propondo um mecanismo único para gerenciar diferentes protocolos de autenticação em ambientes ubíquos. Porém tais mecanismos apresentados apenas preocupam-se em realizar interações de autenticação com o sistema ubíquo, não há uma perspectiva de mudança de regras e definições no ambiente.

Levando-se em consideração o que foi observado nos trabalhos mencionados, percebemos que é necessário agregar outras técnicas, modelos e algoritmos para que o ambiente ubíquo possa compartilhar informações. Há também outros trabalhos pesquisados que permeiam a tese com relação ao controle de privacidade e consequentemente

compartilhamento de dados, podendo descrever a privacidade das informações capturadas e a classificação dos fluxos capturados como público ou privado. Alguns trabalhos antigos já descreviam esse problema e essa preocupação como descrito Satyanarayanan (2001), mas observa-se que ela é recorrente. Com isso, se fez necessária a elaboração de uma definição taxonômica de privacidade em ambientes ubíquos, essa definição serviu para nortear itens como critérios, parâmetros e definições para o controle e gerenciamento de privacidade. Para tanto, também foi desenvolvido um protótipo com as definições e parâmetros elencados, o mesmo foi testado e validado conforme segue nas descrições do próximo capítulo 4, tese.

4 TESE

Neste capítulo será apresentada a tese com os objetivos e definições necessárias para a elaboração do modelo desenvolvido. Inicialmente esse capítulo apresenta uma visão geral da proposta aplicando o modelo proposto ao cenário atual da computação ubíqua. Em seguida são descritos os critérios e definições necessários para o modelo de privacidade desenvolvido relacionado a uma arquitetura de *middleware*. Com base nessas características foi desenvolvido um modelo genérico de controle e gerenciamento de privacidade implementado com base em um cenário de aplicação e um estudo de caso, por fim é apresentado a arquitetura da aplicação.

4.1 Visão Geral

A tese tem como objetivo o desenvolvimento de um modelo de controle e gerenciamento de privacidade para ambientes ubíquos, fundamentado no aporte teórico apresentado no capitulo 2 e na definição taxonômica descrita no capítulo 3, que tiveram seus resultados publicados em Leithardt et al., (2013b). Os objetivos visam também validar os parâmetros e critérios necessários para o controle e gerenciamento de privacidade em ambientes ubíquos com fundamento na teoria e pesquisa realizada. Para tanto, o modelo de privacidade deve contemplar todos os aspectos e requisitos pesquisados na literatura, necessários para controlar os diferentes critérios de acordo com o ambiente individual e suas características. Com isso, serão apresentados neste capítulo o cenário atual sobre privacidade de dados, as definições e os critérios utilizados para o modelo de privacidade proposto, além do detalhamento do middleware e suas definições necessárias para controle e gerenciamento de privacidade em ambientes ubíquos. Na sequência apresentamos o modelo genérico de privacidade e uma descrição de um possível cenário de aplicação e, por fim um estudo de caso é detalhado.

Atualmente a proposta já contempla a especificação do modelo necessário para o controle e gerenciamento de privacidade, denominado Modelo Genérico de Privacidade apresentado por Leithardt et al. (2014), bem como alguns parâmetros, critérios e métricas para identificação do usuário, dispositivo, localização, comunicação para o modelo, além da arquitetura de *middleware* necessária. Com base nesse modelo genérico de privacidade são apresentadas as definições e critérios necessários para o controle e gerenciamento de privacidade em ambientes ubíquos conforme apresenta a próxima seção.

4.2 Definições de Critérios para o Modelo de Privacidade

O critério essencial que permite que os dados sejam armazenados e processados por uma organização é o consentimento e autorização do indivíduo em fornecer os seus dados. A diretiva abrange todos os dados que trafegam através de redes de sinal e acesso a internet aberto denominadas de acesso público na Europa e, portanto, também abrange os dados ou serviços que se originam fora da Europa.

Com base nessas definições, o presente trabalho apresenta um modelo de controle de privacidade no ambiente ubíquo composto de definições de perfis individuais, onde o perfil do contexto de privacidade do ambiente ubíquo deverá ser ajustado para controle de privacidade dos usuários nas características do nosso cotidiano e algumas regras definidas por (DIGITAL AGENDA FOR EUROPE, 2014). Para tanto, foram definidas conforme a literatura pesquisada e as necessidades do modelo proposto as seguintes regras necessárias para realizar o controle e gerenciamento de informações em ambientes ubíquos, conforme segue:

a) Bloqueado: acesso bloqueado aos usuários
   - O acesso ao ambiente poderá ser bloqueado por tempo determinado ou indeterminado;
b) Convidado: acesso controlado e limitado
   - Acesso poderá ser realizado por tempo determinado ou indeterminado;
   - Restrições de serviços e compartilhamentos;
   - Disponibilidade de privacidade controlada.
c) Básico:
   - Compartilhamento de acesso controlado;
   - Compartilhamento de recursos e serviços limitados no ambiente;
   - O fator limitação de uso de aplicação e critérios são definidos por escala podendo variar de 1 a N, dependendo do ambiente e os recursos que o mesmo dispõe. Para tanto será necessário o acesso à base de dados Ubíqua controladora desse ambiente onde a mesma terá cadastrado todos os recursos disponíveis em todos os ambientes pervasivos;
   - Compartilhamento de localização entre outros usuários do ambiente pervasivo.

d) Avançado
   - Acesso a todos os níveis anteriores;
   - Compartilhamento total de acesso;
   - Compartilhamento de recursos e serviços no ambiente;
   - Compartilhamento de localização aos usuários no ambiente ubíquo.

e) Administrativo
   - Acesso a todos os níveis anteriores;
   - Acesso a todos os recursos do ambiente pervasivo;
   - Controle e gerenciamento do ambiente pervasivo.

Esses critérios / regras serão atribuídos pelo ambiente ubíquo ao usuário que tiver sido identificado e localizado pela autenticação do sistema. No entanto, quando se tratar de ambiente pervasivo externo, como parques, praças, ou outros locais públicos, o usuário recebe o critério de privacidade Administrativo, uma vez que não há como o ambiente ubíquo ter domínio sobre um ambiente pervasivo público. O tratamento do contexto do usuário, perfil do usuário, localização, velocidade de movimentação, proximidade entre as pessoas, situação social, etc., será de acordo com os critérios que o mesmo possui no sistema, com isso funções como: luminosidade, nível de ruído, temperatura, umidade, poderão não estar habilitadas para um mesmo usuário em ambientes diferentes.

Há trabalhos como o desenvolvido por Moschetta et al. (2008), que incluiu nessa relação o contexto temporal caracterizado pela hora do dia, informações de calendário semanal, estação do ano, dentre outros, deixando o controle de privacidade do ambiente a cargo do usuário e não do ambiente ubíquo de acordo com os critérios e recursos definidos para cada ambiente. Com base em definições na literatura na Tabela 4.1 são apresentados exemplos de recursos que serão ativados ao usuário pelo ambiente pervasivo de acordo com os critérios definidos ao usuário e que serão utilizados no cenário da aplicação posteriormente. A Tabela 4.1 apresenta na coluna de critérios todos os critérios definidos, com exemplos de recursos que poderão estar disponibilizados ou não para o cenário de aplicação proposto. Os critérios e recursos podem variar de acordo com as definições e especificações de cada ambiente, sendo atribuídos de acordo com as definições minimas de taxonomia necessária.

Dependendo do critério definido ao usuário em determinado ambiente pervasivo, poderá ter recursos disponíveis nos ambientes controlados, limitados ou completos de acordo

com cada caso de usuários e critérios a ele atribuído. Considerando-se ainda que em um local público, naturalmente que o usuário tenha total controle sobre si e seus dispositivos, uma vez que o mesmo se encontra em um local público. Com base nesses critérios se faz necessária à adoção de um *middleware* para o controle e gerenciamento desses diferentes ambientes e configurações, para tanto a próxima subseção descreve a arquitetura de *middleware* que inicialmente foi utilizada como base para a tese.

Tabela 4.1 - Definição de critérios

Critérios / Recursos	Localização	Serviços	Comunicação	Acesso à informação extra	Troca de dados	Alteração de Controle
Bloqueado	Sim	Não	Não	Não	Não	Não
Convidado	Sim	Restrito	Restrito	Restrito	Restrito	Restrito
Básico	Sim	Limitado	Limitado	Limitado	Limitado	Limitado
Avançado	Sim	Sim	Sim	Sim	Sim	Sim
Administrativo	Sim	Sim	Sim	Sim	Sim	Sim

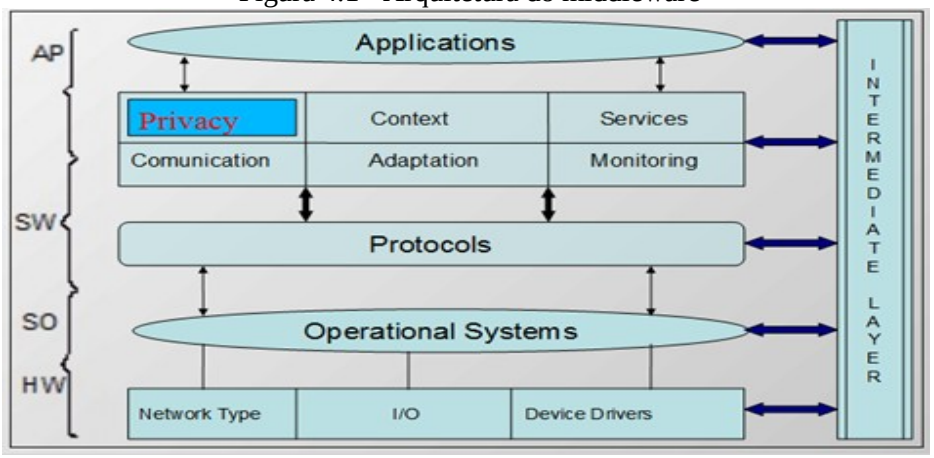
Fonte: Próprio autor.

4.3 Arquitetura Necessária do Middleware

A arquitetura completa do *middleware* (MW) para suportar todos os níveis necessários de controle da aplicação, *software* e hardware foi fundamentada no modelo inicial proposto por Leithardt (2008), onde é apresentado um *middleware* dividido em quatro camadas, sendo camada de hardware, *software*, *middleware* e aplicação voltada para sistemas pervasivos. Na arquitetura desenvolvida por Leithardt (2008) são validados os módulo necessários para testes iniciais de um ambiente pervasivo utilizando uma plataforma de aplicações de controle multimídia aberta (OMAP). Nos testes realizados foram implementados alguns módulos, descantando-se o módulo responsável pelo gerenciamento do contexto. Como cenário de aplicação foi utilizado um sistema de agendas pervasivo com usuários acessando a agenda de forma inteligente. Porém, nessa arquitetura não era previsto o uso de aplicações que envolvessem redes de sensores sem fios (RSSFs) e identificadores por rádio frequência (RFID). Essa observação pode ser confirmada no momento que em foram implementados protocolos para ambientes pervasivos com uso de RSSFs e RFID desenvolvidos no trabalho apresentado por Silva et al. (2009). Nessa aplicação foi possível reduzir o consumo energético nas redes de sensores sem fios com o uso disseminado de *tags* RFID em alguns locais do

ambiente ubíquo. Com isso, verificou-se a necessidade de alterações no *middleware* de forma a dar suporte a aplicações pervasivas/ubíquas com gerencia de protocolos e dispositivos de RSSFs e RFID. Assim, Leithardt, Geyer e Silva (2011) apresentaram uma nova proposta para plataforma de MW capaz de suportar as tecnologias necessárias para ambiente ubíquo com foco no gerenciamento de protocolos e redução de consumo energético, conforme modelo apresentado na Figura 4.1.

Figura 4.1 - Arquitetura do middleware



Fonte: Leithardt et al. (2013b).

O *middleware* proposto por Leithardt, Geyer e Silva (2011) é composto de 4 camadas interligadas englobando as características e requisitos necessários para controle de redes de sensores sem fios e RFID em ambientes ubíquos. Possui como fator chave a redução do consumo energético ocasionado pela redução de trocas de mensagens entre nós e estação base. Desta forma poderá ser utilizado para tratar de soluções híbridas que envolvem ambientes pervasivos/ubíquos, no entanto destaca se o módulo de controle e gerenciamento de privacidade, denominado (*Privacy*). A ideia principal do middleware é permitir que um determinado dispositivo fosse voltado para suprir as necessidades de um usuário ou do ambiente como um todo, adaptando-se ao ambiente e a seu projeto de infraestrutura, conforme as limitações do dispositivo e as necessidades de cada ambiente. Segue a descrição de cada camada, bem como suas atribuições que iniciam na camada de *Hardware* (HW) onde estão dispostos 3 módulos necessários para tratamento de requisitos físicos necessários visando o tratamento dos dispositivos físicos sendo implementado em HW;

a) *Network Type*: utilizada para conectar a plataforma na rede pervasiva, também tem a funcionalidade específica que será utilizada para auxiliar e controlar a funcionalidade eletroeletrônica do dispositivo;

- *I/O*: utilizada para a comunicação e interface com os usuários, ambientes e dispositivos;
- *Device Drivers*: unidade de pré-processamento responsável pelo gerenciamento, armazenamento e execução dos requisitos mínimos para funcionamento dos dispositivos. Funciona como gatilho de conexão dos dispositivos físicos cadastrados no sistema, como por exemplo, especificações de *MAC address*, IMEI "*International Mobile Equipment Identity*", Bluetooth, dentre outras.

b) Layer Operating Systems (SO): tem como objetivo tratar as funcionalidades dos sistemas operacionais para o sistema embarcado. É auxiliada pelo módulo *Device Drivers* que gerencia os componentes da camada física e envia as informações para a subcamada do Sistema Operacional que é responsável por gerenciar as tarefas da aplicação que executa no dispositivo e determina os serviços disponibilizados pelo mesmo, além de tratar as limitações do sistema operacional.

c) Layer Software (SW) constitui um conjunto de componentes necessários para auxiliar a integração e tratamento do dispositivo com a rede ubíqua e disponibilizar os serviços e demais funcionalidades necessárias que compõem o *middleware* na arquitetura. Essa camada é formada por sete componentes básicos, são eles:

- *Module Protocols*: este módulo realiza o tratamento de protocolos necessários para a integração das funcionalidades do middleware, alguns protocolos como zigbee que estão diretamente relacionados às redes de sensores podem ter acesso direto com a aplicação. Para tanto, se utiliza a camada intermediária para fazer a integração sem a necessidade de utilizar os demais módulos e funcionalidades do middleware;
- *Module Communication*: que integra o dispositivo na rede e gerencia a comunicação do dispositivo com outros dispositivos;
- *Module services*: que fornece gerenciamento dos serviços e recursos de ambientes e dispositivos para a aplicação ubíqua. Também possui o atributo de fornecer e controlar a adaptação de novos componentes de SW;
- *Module Adaptation*: responsável pela adaptação e gerenciamento de usuários, serviços, dispositivos, aplicação, comunicação e ambiente pervasivo/ubíquo;
- *Module Privacy*: este módulo é responsável pelo tratamento de privacidade do ambiente, fornecendo serviços de controle e autenticação. É nesse módulo que o

modelo proposto se encaixa com a principal finalidade de gerenciar e controlar os diversos tipos de compartilhamento relacionando-os a privacidade do ambiente;

- *Module Context*: auxilia na detecção de contexto do ambiente e usuário para a arquitetura;
- *Module Monitoring*: fornece o monitoramento do ambiente e de dispositivos para a aplicação, informando status, erros e problemas.

Camada *Applications* (AP) é um módulo que possui fragmentos das aplicações que executam no ambiente, nessa camada serão realizadas todas as configurações necessárias para que qualquer aplicação funcione no ambiente ubíquo. Um exemplo de aplicação seria disponibilizar os serviços de uma cafeteira para o ambiente ubíquo onde determinado usuário se encontre.

Camada *Intermediate*: essa camada tem a finalidade de interligar e interagir com todas as camadas simultaneamente, tendo como seu principal objetivo estabelecer conexão entre uma camada e outra sem necessariamente conectar todas as demais. Por exemplo, um determinado hardware tipo um sensor, poderá ter como única finalidade avisar a aplicação na ocorrência de um evento. Para tal situação o próprio sensor poderá ter sistema operacional para se auto gerenciar, não havendo necessidade de conectar-se com a camada de sistema operacional e as demais, economizando recursos da plataforma.

O tratamento da privacidade do ambiente será inserido no módulo de contexto do *middleware*, uma vez que o contexto pode ser tratado no nível usuário, dispositivos, comunicação, serviços e ambientes pervasivos/ubíquos. Com isso, não será necessário reestruturar a arquitetura existente, podendo ser reaproveitado o *middleware* existente proposto por Leithardt, Geyer e Silva (2011). Basicamente haverá uma especificação dentro do módulo que são chamadas de acordo com as especificações, definições, necessidades e parâmetros, nessas chamadas serão inseridas opções de utilização e acionamento do módulo contexto de privacidade com os seguintes tipos de opções: 1- Ambiente; 2- Usuário; 3- Dispositivos; 4- Comunicação; 5- Serviços; 6- Aplicação; 7- Outros. Com isso, é possível habilitar a utilização da arquitetura de *middleware* no modelo genérico de privacidade que será descrito na próxima seção.

4.4 Modelo Genérico de Privacidade

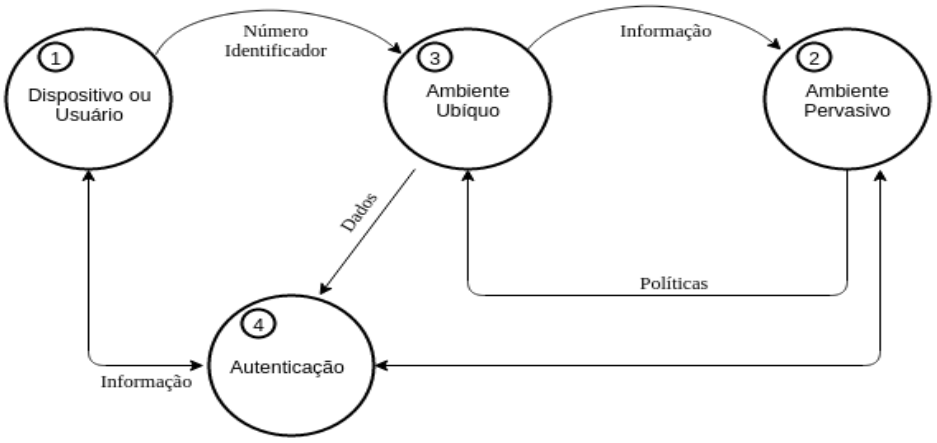
No modelo de privacidade desenvolvido consideramos que o usuário ao cadastrar-se em um ambiente ubíquo, forneça todas as informações necessárias para que o mesmo possa

utilizar os recursos e disponibilidades do ambiente com base em seus critérios recebidos. A opção da classe controlada para que o usuário se encaixe caberá ao ambiente disponibilizar, bem como as regras definidas para cada ambiente decidir com base no perfil de cada usuário que for identificado. Somente o perfil administrador poderá realizar alterações, uma vez que o usuário é relacionado ao ambiente e não ao contrário. Conforme descrito na introdução desta tese, o modelo genérico de privacidade apresenta um ambiente ubíquo ao qual o mesmo define as regras, devido ao fato que para controlar e gerenciar a privacidade é necessário definir critérios e estabelecer normas nos ambientes conforme apresentado na tabela e descrições da seção 3.2.

Por exemplo, em um local privado ou restrito, entende-se como um local onde haverá mais regras e definições que não importam o tipo de usuário e sim as definições do ambiente. E locais públicos e/ou sem atribuições de critérios como ambientes em que o usuário poderá utilizar suas definições de uso de dispositivo com seus gostos e preferências, como por exemplo som alto ao tocar o celular independente de horário. Porém, mesmo dentro desses ambientes poderá haver variações, se por exemplo, dentro de um determinado horário não ser permitido som alto, por que esse ambiente deixou de ser público. Isso faz surgir várias questões que serão discutidas na sessão de cenário da aplicação, no entanto, é necessária a elaboração e descrição detalhada de um modelo genérico de privacidade, conforme apresentado na Figura 4.2.

O modelo de processos de privacidade para ambientes ubíquos apresentado na Figura 4.2 é composto de 4 processos, sendo que cada processo possui características individuais conforme definições descritas na seção 3.1, onde foram definidas a taxonomia de cada usuário, dispositivos e funcionalidades de objetos controlados pela privacidade. Com base nessas definições desenvolvemos o Modelo Genérico de Privacidade com os processos interagindo entre si com as seguintes descrições:

Figura 4.2 - Modelo de Privacidade de Ambientes



Fonte: Próprio autor.

1) Usuário/Dispositivo: esse processo é responsável pela identificação do usuário ou dispositivo no ambiente pervasivo. A identificação equivale ao seu *login* ao sistema, gerando um fluxo de dados com sua identificação direcionado ao processo 2, onde o mesmo pretende ser detectado.
2) Ambiente pervasivo: nesse processo são processadas as informações recebidas do usuário, bem como o dispositivo utilizado para efetuar a tentativa de *login* ao sistema. Cada ambiente pervasivo possui suas próprias definições e regras, que por sua vez estão cadastradas no próprio ambiente e são copiadas no processo 3. Essas informações também são enviadas ao processo 3 juntamente com as informações de solicitação do usuário, gerando outro fluxo de informações para o processo 3.
3) Ambiente ubíquo: nesse processo estão armazenadas todas as informações dos usuários cadastrados no ambiente ubíquo, inclusive suas preferências e definições.
4) Políticas: Os dados recebidos do ambiente pervasivo são processados e validados de acordo com as definições do local (ambiente pervasivo), que o usuário pretende efetuar *login*;. Optou se em dividir os ambientes em pervasivo e ubíquo para melhor exemplificar o funcionamento, tais definições são fundamentadas na relação entre computação pervasiva, móvel e ubíqua apresentadas na introdução da tese.
5) Autenticação: nesse processo são recebidos os dados enviados pelo processo 3, contendo as regras e definições de permissões ao usuário ou dispositivo que solicitou o *login* ao ambiente pervasivo, essas regras estão definidas na base de dados e variam de acordo com cada ambiente individualmente. Posteriormente ao processamento das informações recebidas, estas são repassadas ao usuário,

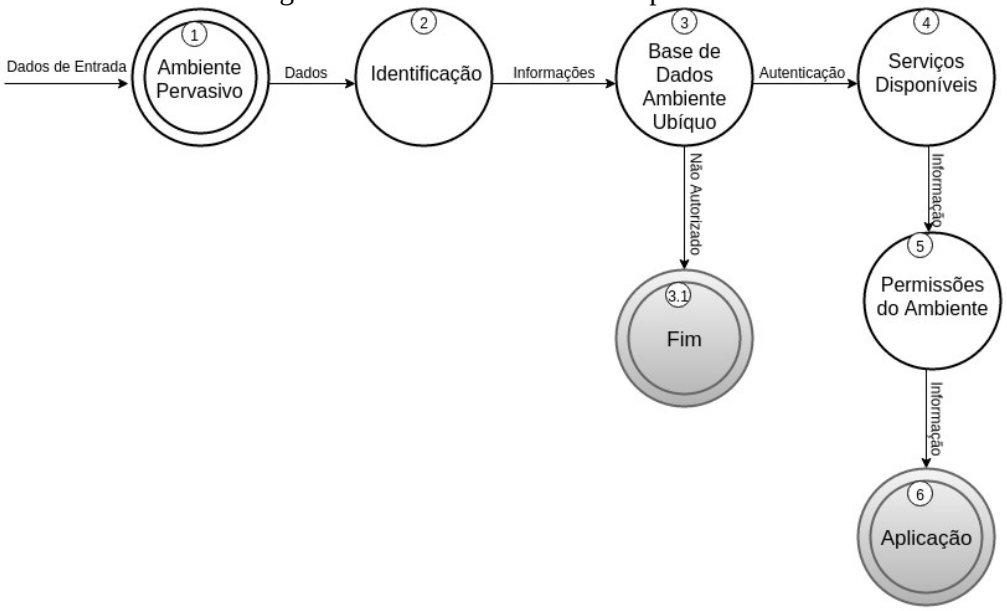
validando seu *login* de acordo com o tipo definido, conforme os critérios que foram determinados na Seção 3.3.

A partir dessas definições básicas do modelo genérico, se faz necessário explorar individualmente cada processo do modelo, de forma a descrever quais são suas atribuições e controles individuais, iniciando pelo processo 2, conforme mostrado na Figura 4.3.

No modelo genérico de privacidade consideramos ambientes privados ou restritos como ambientes pervasivos e ambientes ubíquos como ambientes públicos e/ou com funções no qual o usuário possa alterar. Com base no modelo do ambiente pervasivo apresentado na Figura 4.3, a transição de dados e informações ocorre no modelo da seguinte forma:

1) ambiente pervasivo: inicialmente recebe os dados de entrada que são processados no ambiente e registrados o local, a data e a hora da solicitação;
2) identificação: os dados vindos do ambiente pervasivo são identificados e as informações são repassadas ao ambiente ubíquo que possui uma base de dados com todos os cadastros e registros, tanto de usuário quanto de ambientes e serviços;
3) base de dados ambiente ubíquo: possui os dados que são executados para validação das informações recebidas, caso não haja cadastro, não há validação, consequentemente a solicitação é finalizada. Havendo cadastro dos dados recebidos, estes são enviados ao módulo de serviços disponíveis onde são armazenados todos os serviços disponíveis de todos os ambientes pervasivos. Este por sua vez encaminha as informações de autenticação do usuário e os serviços disponíveis ao ambiente pervasivo para que repasse à aplicação para iniciar a utilização;
4) serviços disponíveis: armazena todos os serviços disponíveis do ambiente pervasivo e que estão em uso;
5) ambiente pervasivo: ambiente onde o usuário está autenticado com dados atualizados de serviços em uso e disponíveis;
6) aplicação: controla no ambiente pervasivo os acessos de usuários e os serviços disponíveis e que estão sendo utilizados.

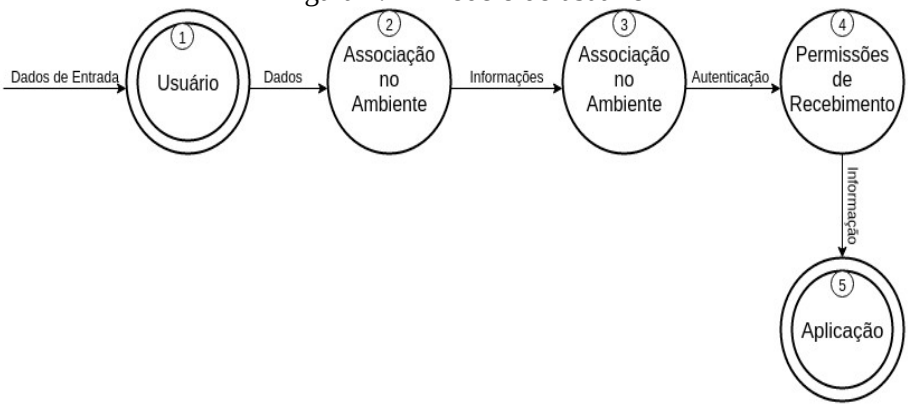
Figura 4.3 - Modelo do ambiente pervasivo.



Fonte: Próprio autor.

Na Figura 4.4 é apresentado o modelo do usuário com base no modelo genérico de privacidade definido anteriormente no processo 2 da Figura 4.3.

Figura 4.4 - Modelo do usuário



Fonte: Próprio autor.

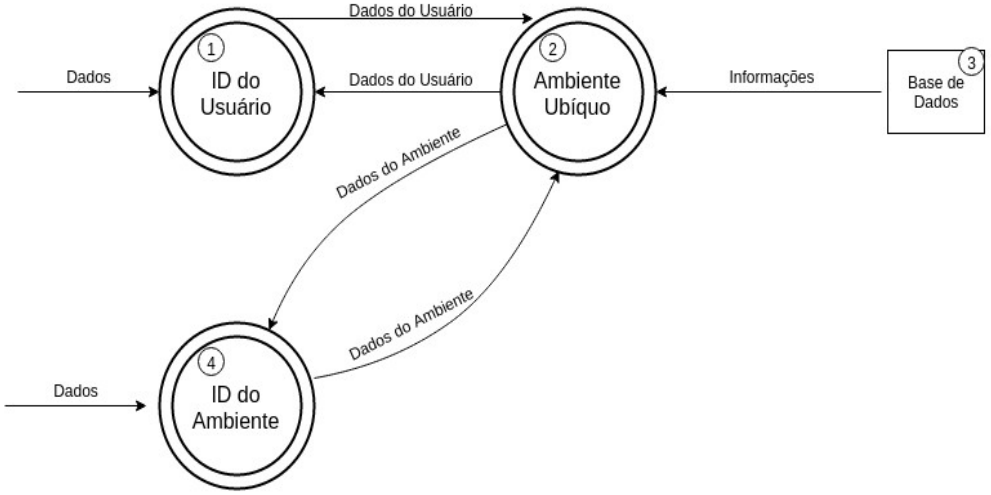
Com base no modelo genérico do usuário apresentado na Figura 4.4, a transição de dados e informações ocorre no modelo da seguinte forma:

1) usuário: inicialmente é identificado por meio de um dispositivo eletrônico (Smartphone, cartão com RFID, entre outros), sendo esses dados enviados para associação ao ambiente pervasivo onde pretende se registrar;
2) associação no ambiente: os dados vindos do usuário são identificados e registrados, as informações são repassadas ao módulo de transição seguinte;

3) aguardando aprovação: neste processo onde os dados são mantidos temporariamente até que sejam recebidas as permissões de acesso e utilização com os devidos critérios definidos;
4) permissões de recebimento: processo que registra as permissões do usuário, dispositivo, comunicação, serviços e demais dados permitidos ao usuário. Poderá também ter uma espécie de cache para tratar de situações de *login/logout* do usuário em curto espaço de tempo;
5) aplicação: controla no ambiente pervasivo os acessos de usuários e serviços sendo utilizados e disponíveis.

Na Figura 4.5, é apresentado o modelo genérico do ambiente ubíquo com base no modelo de privacidade definido anteriormente no processo 3 da Figura 4.2.

Figura 4.5 - Modelo do ambiente ubíquo



Fonte: Próprio autor.

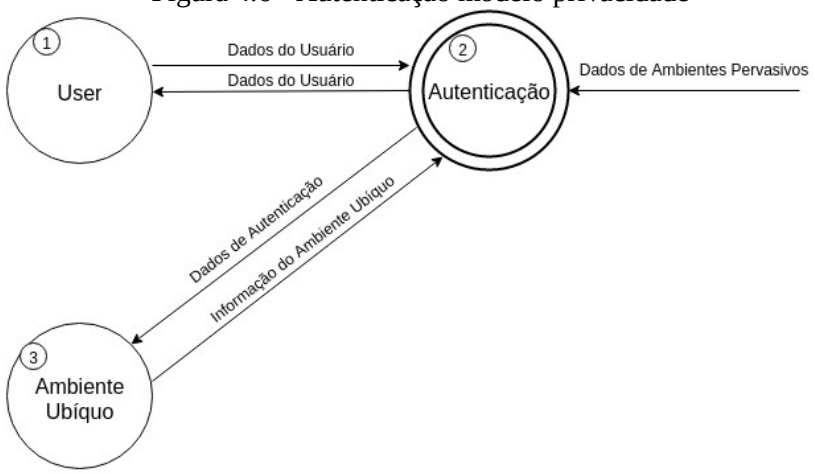
Com base no modelo genérico do ambiente ubíquo apresentado na Figura 4.5, a transição de dados e informações ocorre no modelo da seguinte forma:

1) usuário: inicialmente é identificado por meio de um dispositivo eletrônico que pertence ao mesmo, os dados de login são enviados para o ambiente ubíquo onde pretende se registrar. Neste caso especificamente o usuário pretende efetuar *login* e registro diretamente no ambiente ubíquo, para situações onde, por exemplo, o mesmo se encontre em ambientes públicos e/ou privados. Com isso o mesmo receberá como retorno dados de outros usuários que se encontrem na mesma situação;

2) ambiente ubíquo: tem a função de receber, controlar e registrar as informações recebidas de usuários relacionados aos ambientes pervasivos, portanto, está sendo considerado essas definições de ambientes com base nas descrições apresentadas na introdução de tese. Essas informações são repassadas a base de dados e consequentemente são registradas e atualizadas tanto de solicitações recebidas de usuários quanto de ambientes pervasivos;
3) base de dados: neste processo os dados do usuário, ambiente pervasivo e ubíquo são cadastrados, mantidos, controlados e atualizados;
4) identificação do ambiente: neste processo possui as regras, definições e critérios para utilização de acordo com variáveis que mudam conforme cada ambiente individual.

Na Figura 4.6, é apresentado o modelo de autenticação com base no modelo genérico de privacidade definido anteriormente no processo 4.

Figura 4.6 - Autenticação modelo privacidade



Fonte: Próprio autor.

Com base no modelo de autenticação apresentado na Figura 4.6, a transição de dados e informações ocorrem no modelo da seguinte forma:

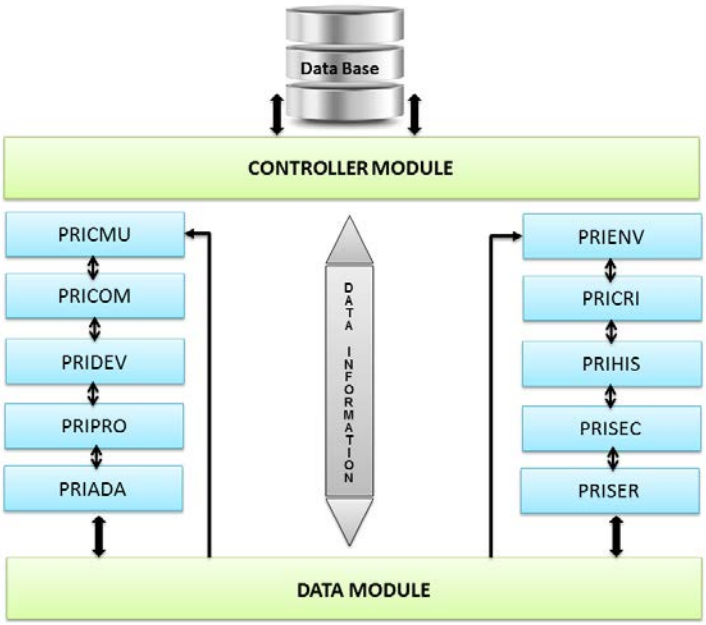
1) usuário: o usuário inicialmente é identificado utilizando um dispositivo eletrônico, ambos os dados do usuário e do dispositivo, são enviados para autenticação. A relação do dispositivo com o usuário é considerada para a autenticação no ambiente ubíquo.

2) Autenticação: nesse processo são recebidos os dados enviados pelo processo usuário, contendo as regras e definições que são gerenciadas e enviadas pela aplicação contendo a lista de permissões com base nos critérios definidos pelo ambiente ao usuário ou dispositivo que solicitou o *login* ao ambiente pervasivo. Posteriormente ao processamento das informações recebidas, estas são repassadas ao usuário, validando seu *login* com tipo definido, conforme os critérios definidos na seção 3.1. Poderão também ocorrer situações onde o usuário solicita autenticação diretamente no ambiente ubíquo, por exemplo, quando o usuário está em ambientes públicos. Com isso, o usuário receberá como retorno dados de outros usuários que se encontrem na mesma situação;
3) ambiente ubíquo: tem a função de receber, controlar e registrar as informações recebidas de usuários e ambientes pervasivos. Essas informações são repassadas a base de dados e consequentemente são registradas e atualizadas tanto as solicitações recebidas de usuários quanto de ambientes pervasivos;
4) base de dados: neste processo os dados do usuário, ambiente pervasivo e ubíquo que são identificados e diferenciados através dos critérios e definições atribuídas a cada ambiente individualmente sendo cadastrados, mantidos, controlados e atualizados conforme apresentado na Figura 4.7.

4.5 Modelo Gerenciador de Privacidade (UbiPri)

Com base na descrição apresentada no módulo de privacidade do middleware de arquitetura apresentado na Figura 4.1, foi desenvolvido o modelo de controle e gerenciamento de privacidade denominado UbiPri (Leithardt et al., 2013b), conforme Figura 4.7, para ambientes pervasivos/ubíquos que visa atender essas características e necessidades. O modelo gerenciador de privacidade UbiPri possui ligações que relacionam todos os módulos responsáveis pelo controle e gerenciamento de privacidade de ambientes, onde cada módulo possui uma relação e descrição de funções conforme segue:

Figura 4.7 - Modelo Gerenciador de Privacidade



Fonte: Leithardt et al. (2013b).

1) *Data Base*: módulo responsável pelo armazenamento de informações e definições de regras sobre usuários, dispositivos e comunicações no ambiente pervasivo/ubíquo. Essa base de dados atua como uma espécie de cadastro único contendo todas as informações necessárias para o controle e gerenciamento do mecanismo de privacidade em ambientes pervasivos/ubíquos. A base de dados está diretamente direcionada ao módulo de dados (*Data Module*) representado por setas bidirecionais, onde as informações serão trocadas com o módulo de dados.
2) *Controller Module*: Esse módulo tem a finalidade de receber solicitações de acesso e realizar o controle da base de dados diretamente nas tabelas, com informações necessárias de acordo com as solicitações e definições de acesso e controle do ambiente pervasivo/ubíquo. Esse módulo também realiza solicitações de validações e atualizações na base de dados após as informações terem sido retornadas ao módulo sendo calculadas e tratadas anteriormente pelo módulo de controle. Tais cálculos de controle serão fundamentados nas características e definições estabelecidas em cada módulo de gerenciamento de privacidade do modelo gerenciador. Com os dados atualizados e definidos são retornados ao solicitante suas permissões de acordo com as definições de critérios e permissões recebidas.

3) *Data Module*: neste módulo serão realizados os processamentos de todas as variáveis e parâmetros recebidos dos demais módulos. Também tem a função de receber diversos dados e tratá-los gerando uma única saída de informações a cada processamento executado. Para tanto, é utilizado algoritmos com funcionalidades e cálculos matemáticos relacionados à necessidade e particularidade de cada ambiente, com isso foram necessárias a execução de variáveis e parâmetros individuais do ambiente como que usuário, dispositivos, serviços, perfis, etc. Para tanto, desenvolveu – se uma expressão matemática para atender esses requisitos e funcionalidades necessárias que é definida da seguinte forma:

$$S(r,c)=A_{r,c}$$

Onde S corresponde ao serviço disponibilizado como por exemplo funcionalidades não oferecidas pelo próprio dispositivo do usuário e que podem estar disponível no ambiente em que o mesmo se encontra, tais serviços estão relacionados diretamente ao perfil em que o usuário tem definido e os critérios a ele atribuídos. A letra R representam os recursos disponíveis no ambiente e que poderão ser ativados ao usuário como por exemplo a ativação de um ar condicionado, o acionamento automático de luzes no ambiente, etc. A variável C corresponde aos critérios atribuídos ao usuário de acordo com as definições e regras da localização e o ambiente. E por fim a variável A é atribuída à tabela de regras e critérios definidos para cada ambiente, uma vez que cada ambiente possui regras, critérios e definições diferentes para dispositivos, usuários diferentes, por tanto, uma tabela com definições básicas é necessária para o controle e gerenciamento de privacidade.

a) *PRICMU*: módulo de gerenciamento e controle de privacidade das informações do usuário. Neste módulo serão tratadas as definições de características relacionadas às preferências individuais do usuário como, por exemplo: temperatura, luminosidade, compartilhamentos autorizados (informações que o usuário deseja compartilhar com outros usuários e com o próprio ambiente), tais como localização, dados e outras preferências e serviços que podem ser disponibilizados aos usuários. A interligação é realizada com o módulo de gerenciamento de histórico PRIHIS que possui a finalidade de realizar o tratamento refinado sobre dados históricos. Em Cambruzzi et al. (2012) foi desenvolvido um protótipo onde

múltiplas aplicações podem compartilhar dados com finalidades específicas que realizam o mapeamento e registro de contextos, com base nesse mapeamento é possível detalhar dimensões temporais entre outras funcionalidades que poderão se adaptar ao módulo PRIHIS.

b) *PRICOM*: módulo responsável pelo gerenciamento e controle de privacidade de comunicações, onde serão tratadas as formas de comunicação e como estas serão utilizadas dentro do ambiente pervasivo/ubíquo, como por exemplo, restrições de sinal ou tipo de adaptador (utilizado com a finalidade de servir também como um controlador de acessos conforme é o ambiente no mundo real, onde determinados ambientes apenas possuem um tipo de comunicação).

c) *PRIDEV*: módulo de gerenciamento e controle de privacidade de dispositivos. Neste serão tratados os dados sobre os dispositivos que podem ser do próprio ambiente ou aqueles passam a interagir com ele. O gerenciamento e controle se referem às características relacionadas ao *software* e hardware vinculados a cada dispositivo individualmente, tais como tamanho, peso, resolução de tela, sistema operacional, meio de comunicação, etc.

d) *PRIPRO*: neste módulo serão realizadas as transações de controles relativos ao gerenciamento de perfil do usuário, as definições de funcionamento do perfil estão detalhadas na Figura 4.4. Tem por função controlar as informações previamente definidas por uma ferramenta de busca. Assim, terá apenas a finalidade de distribuir e direcionar a informação aos módulos seguintes de maneira sintetizada a fim de adaptá-la de forma mais adequada com relação à privacidade individual e com base no perfil individual.

e) *PRIADA*: módulo de gerenciamento e controle de adaptação, responsável pelo tratamento das informações relacionadas à adaptação de *software* e hardware no ambiente pervasivo/ubíquo. Podemos citar como exemplo os conteúdos e mídias que podem funcionar em determinados dispositivos e em outros dispositivos com características diferentes, podem não ter o mesmo desempenho ou funcionalidade por várias questões, como tamanho, comunicação, configuração, dentre outras características necessárias para o funcionamento, necessitando assim um mecanismo de controle e gerenciamento adaptativo.

f) *PRIENV*: neste módulo são cadastrados os atributos relacionados ao ambiente, com essas informações será possível verificar e gerenciar o que compõe o ambiente,

suas disponibilidades e capacidades a fim de compartilhar os recursos e serviços aos usuários que necessitarem.

g) *PRICRI*: esse módulo possui as regras e definições de critérios e definições do ambiente como acesso, utilização, compartilhamento, localização e outras variáveis que poderão ser incluídas, alteradas, modificadas e/ou substituídas de acordo com as próprias definições do ambiente com relação aos critérios e normas estabelecidos. Parte – se do pressuposto de que cada ambiente possui características individuais como, por exemplo, uma sala de aula pode possuir horários diferenciados para diferentes usuários e dispositivos durante um mesmo dia, necessitando controles e configurações de perfis e critérios individualizados. Essas definições são tratadas individualmente pelos demais módulos que possuem características e controles específicos. As definições de funcionamento são pré-estabelecidas para cada ambiente podendo ter variações, como por exemplo, um usuário acessar um determinado ambiente no mesmo dia tendo diferentes critérios definidos de acordo com o horário de acesso.

h) *PRIHIS*: Neste módulo serão armazenadas e tratadas às informações relativas ao histórico do usuário, ambiente, dispositivos e outras variáveis que poderão ser adicionadas posteriormente com objetivo de obter informações contextuais. A característica de funcionamento é a utilização de informações que são capturadas ao longo de um determinado período tendo como base outras fontes de informações como, por exemplo, múltiplas trilhas, contexto, etc. Este módulo funciona também como uma espécie de cache pré-definida de possíveis situações a serem utilizadas como, por exemplo: histórico de determinada sala que durante 10 anos vem sendo utilizada por turmas relacionadas a uma determinada disciplina como informática. Poderia também ter um histórico de realizar determinadas manutenções em alguns computadores em dado tempo pelo histórico de manutenções efetivadas anteriormente.

i) *PRISEC*: neste módulo serão realizados os controles e gerenciamentos relacionados à segurança, tanto do usuário quanto do ambiente. Tem a função de receber os parâmetros e definições relacionados à criptografia de dados ou outros tratamentos relacionados à segurança e encaminhá-los ao solicitante de acordo com a necessidade para cada situação. Por exemplo, ao adentrar em determinado ambiente o usuário poderá ser bloqueado por situações que fogem aos critérios definidos para ele dentro do ambiente tais como, data e hora não permitidas.

j) *PRISER*: módulo de gerenciamento de serviços do ambiente, cuja função é tratar informações sobre a disponibilidade de serviços que serão utilizados individualmente em cada ambiente como, por exemplo, informações que são compartilhadas com outros ambientes, tais como dispositivos, comunicações, localização de usuários, disponibilidades do ambiente e seus componentes que interagem com os usuários. As definições e regras para utilização e disponibilização desses serviços são inseridas no módulo de critérios do ambiente a fim de controlar o acesso e gerenciamento.

Todos os módulos atuam de forma independente, com características e funcionalidades próprias que podem variar de acordo com as regras previamente estabelecidas, cadastradas e gerenciadas. Portanto, um módulo no modelo definido não depende diretamente de outro, bastando apenas acesso aos dados contidos na base dados conforme a Figura 4.7. Uma vez definidas essas regras, cada módulo estabelece os parâmetros com base nas definições do módulo anterior. Sendo assim, para um mesmo ambiente ubíquo é possível ter vários ambientes com regras e definições diferentes. E um mesmo usuário poderá utilizar um ou vários ambientes diferentes, cada um tendo um critério definido que pode mudar com base no dispositivo utilizado, na comunicação, entre outros quesitos que serão processados no módulo de dados. Para melhor exemplificar as situações definidas, é apresentado estudos de casos com alguns cenários de aplicação com utilização do modelo proposto de acordo com os quesitos taxonômicos e critérios definidos.

4.6 Estudos de Caso

Nesta seção serão apresentados alguns cenários de estudo de caso para o modelo proposto de privacidade, com isso é possível contribuir e validar a aplicabilidade do modelo de privacidade desenvolvido. O mesmo possui características que podem ser adaptáveis a vários ambientes, dispositivos e usuários com critérios e definições heterogêneas. Para tanto, foram descritos os seguintes cenários.

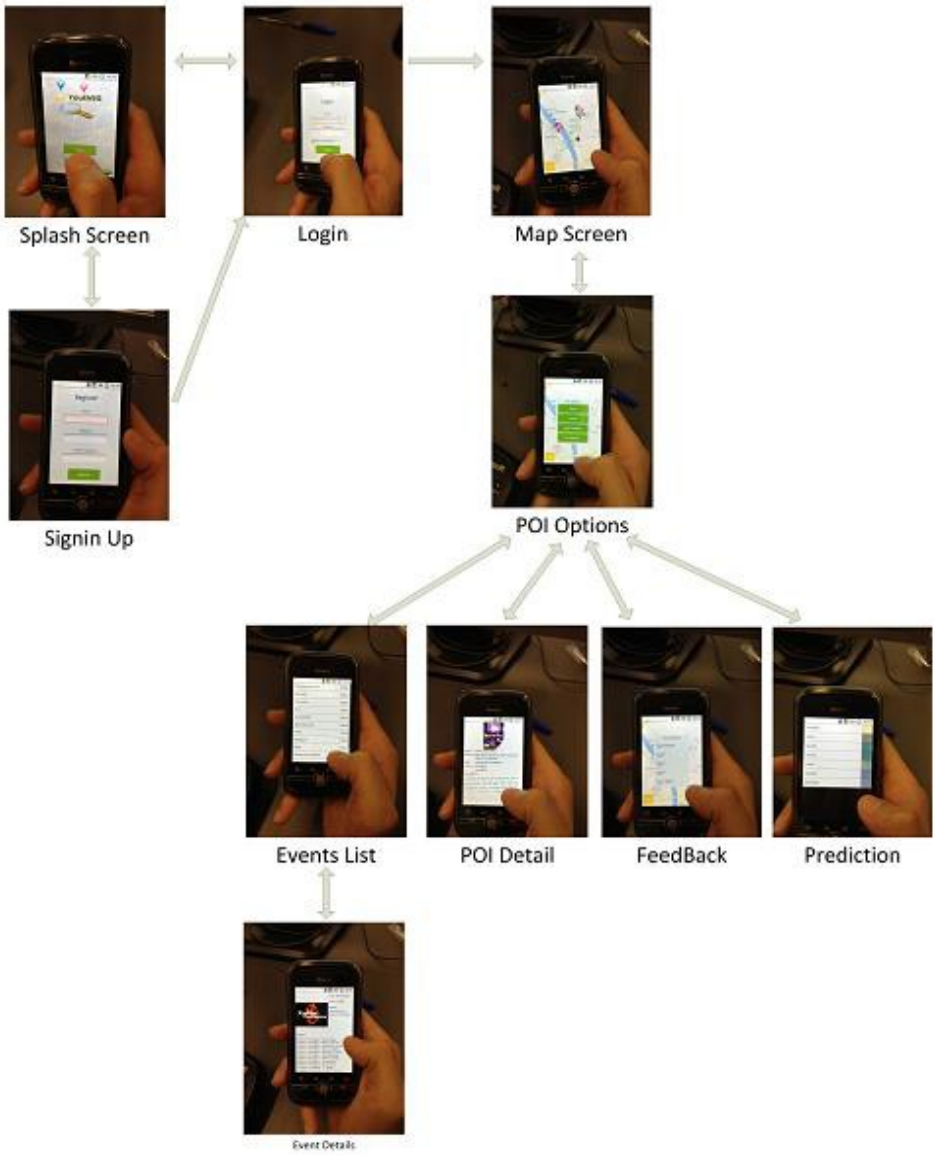
4.6.1 Cenário de aplicação 1

Em uma sexta-feira à noite, “João” decide sair com o seu habitual grupo de amigos. O seu dispositivo móvel, utilizando *Android*, tem acesso ao sistema denominado YOUTH5G

(Sistema de identificação de movimento em ambientes utilizando sensores sem fios), o qual é capaz de usar o sensor GPS para determinar a sua posição atual. Ao fazer *login* no sistema YOUTH5G, é atribuído ao João um perfil de privacidade conforme o tipo de ambiente, o qual é dependente do ambiente onde ele se encontra. No seu bar habitual, onde o João é cliente frequente, ele tem acesso a vários serviços de utilizador "avançado", tendo direito a uma bebida grátis. No entanto, João e seus amigos decidem experimentar um bar diferente, desta vez. O João acessa a aplicação YOUTH5G, a qual usa os serviços SAPO (SAPOSERVICES, 2014) para retornar os pontos de interesse referentes aos bares mais próximos da sua localização atual. Ao clicar num dos bares, João tem acesso à informação referente à densidade populacional de cada bar, bem como ao nível de "agito, movimentação" nesse bar; os bares onde o ambiente é mais "agitado" (os clientes dançam e movem-se mais) terão um nível de excitação maior que os bares com um ambiente mais calmo (os clientes estão sentados a conversar). Usando a informação disponibilizada pelo YOUTH5G, João e seus amigos decidem ir conhecer um bar novo, mais populoso e agitado do que o habitual. Ao entrar no bar, é atribuído ao João um novo perfil de privacidade "*guest*", que não lhe dá direito à bebida grátis, mas de qualquer forma, João e seus amigos passam uma excelente noite de sexta-feira e divertem-se muito.

Para validar o trabalho de acordo com o modelo genérico de privacidade e middleware apresentado na Figura 4.1, com os requisitos necessários, utilizamos o cenário da aplicação descrito anteriormente. Para tal, foi elaborado um protótipo inicial tendo como base a utilização de dispositivos móveis, inicialmente com usuário entrando em um ambiente pervasivo conforme as descrições anteriores. A Figura 4.8 apresenta as diferentes etapas da aplicação no cenário descrito e modelado no protótipo.

Figura 4.8 - Exemplificação de funcionamento



Fonte: Próprio autor.

Atualmente existem vários sistemas semelhantes que combinam informações compartilhadas por utilizadores para vários fins sociais, como por exemplo, o Waze (WAZE, 2015), o Foursquare (FOURSQUARE, 2015) e o GetGlue (GETGLUE, 2015). No entanto, todas estas aplicações são estáticas, ou seja, a informação adquirida é dependente da colaboração voluntária dos utilizadores, e das informações inseridas por cada elemento da rede. A solução proposta neste cenário de aplicação traz um diferencial, uma vez que as informações são recebidas e tratadas em tempo real, utilizando os sensores de movimento e localização do dispositivo móvel. Isto torna a aplicação mais atrativa, mais flexível e mais útil, permitindo também uma maior expansão em termos de funcionalidades. Em vez de depender unicamente das informações inseridas pelos usuários, o sistema usa os dispositivos

ubíquos do ambiente e do utilizador para providenciar informação mais confiável e flexível. Com isso, é possível habilitar um ou dois compartilhamentos para validação inicial, como , por exemplo, local onde esteja tocando música ao vivo e/ou determinado artista se apresentando, nesse caso haveria desativação de sinal Wi-Fi e 3G local, a fim de não permitir o envio de informações do local, como filmagem do artista se apresentando ao vivo.

Também é possível a localização de objetos dentro dos ambientes. Para este fim, poderemos utilizar o modelo proposto por Kubicki, Lepreux e Kolski (2011), onde se empregam uma espécie de tabelas interativas TangiSense, que está equipada com tecnologia RFID. A interação acontece com a manipulação de objetos concretos colocados sobre a mesa. Esta tecnologia torna possível identificar os objetos, que podem ser acoplados com os utilizadores. Partindo deste princípio, a aplicação propõe o uso de etiquetas RFID para coletar os elementos de sensibilidade ao contexto, a fim de adaptar os espaços de trabalho para as diversas situações possíveis em torno de uma mesa ou objetos (trabalhar sozinho ou com vários usuários, em um espaço comum ou individual). Com isso, é possível, por exemplo, ampliar a ideia para uma sala maior, podendo captar informações dos objetos presentes e realizar tratamento do contexto, consequentemente o compartilhamento dos mesmos controlando a privacidade no ambiente.

Inicialmente foi elaborado apenas um protótipo para validar os requisitos da taxonomia descrita conforme apresentado na seção 3. Foi utilizado o modelo genérico de privacidade para este cenário para validar o funcionamento da aplicação e localização em diferentes locais, os resultados obtidos foram apresentados em Leithardt et al. (2013c). Várias questões não foram abordadas como segurança, desempenho, escalabilidade e arquitetura computacional utilizada. No entanto, cabe ressaltar que estes e outros tópicos necessitam ser abordados em trabalhos futuros a fim de melhorar a qualidade e comprovar a validade do modelo taxonômico e genérico apresentado. Na próxima seção serão apresentadas outras características utilizadas com base em estudo caso proposto conforme segue.

4.6.2 Cenário de aplicação 2

Até pouco tempo a maioria das comunicações entre seres humanos era feita na forma de papel: um mecanismo de registro de dado muito antigo e respeitado. Já no último século, a humanidade experimentou uma revolução tecnológica sem precedentes: a invenção do transístor é um dos sistemas eletrônicos que possibilitaram mudar completamente a maneira como guardamos informação e como comunicamos uns com os outros.

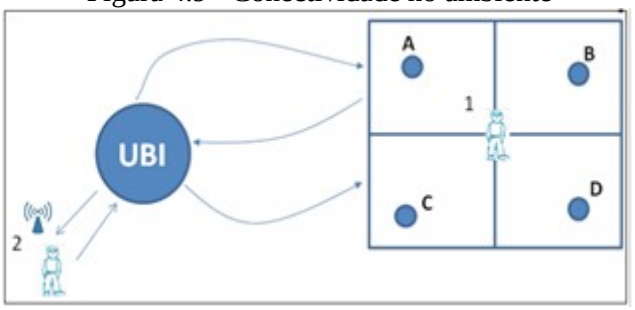
Na última década tem-se também verificado uma grande evolução em tecnologias de comunicação móveis; desde o seu aparecimento, os dispositivos móveis têm evoluído incrivelmente rápido, sendo agora uma poderosa fonte computacional, com grande capacidade de processamento, memória e com acesso a muitos tipos de sensores. Os dispositivos móveis evoluíram também em termos de aspecto e funcionalidades, assumindo agora a forma de "smartphones" e "*tablets*", tornando-se efetivamente computadores "de bolso". Seguindo as mesmas tendências que a maioria da tecnologia baseada em silício, os dispositivos móveis estão também cada vez mais baratos e disseminados, sendo perfeitamente comum encontrar dispositivos móveis com ligação à Internet mesmo em países em desenvolvimento.

Nunca, em toda a sua história, teve a humanidade uma coleção de tecnologias capazes de ligar e recolher informação sobre um número tão grande de usuários, independentemente do seu lugar físico. Em Ploderer, Howard e Thomas (2010) é apresentado um estudo que descreve um tipo de restrição de privacidade para facilitar as interações entre estranhos, tendo como base uma aplicação voltada ao público esportista. Esta pesquisa apresenta no estado da arte uma solução chamada *BodySpace*, que não oferece opções de controle de privacidade que restringem a visibilidade dos perfis das pessoas, isto porque as pessoas formam impressões dos outros usuários com base em fotos, descrições pessoais e links para amigos no perfil, antes de estabelecer conexões com estranhos on-line. A combinação destes fatores abre as portas à criação de muitos tipos novos de aplicações, que combinem o uso de informação sensorial aplicada ao nível individual, com a capacidade de troca de informações das redes sociais, bem como com a liberdade móvel dos dispositivos do tipo "*tablet*" e "smartphone". É neste contexto que funciona a aplicação denominada YOUTH5G, cujo cenário de aplicação é o de o ambiente noturno utilizando as regras e definições do modelo genérico de privacidade. O YOUTH5G é um sistema que combina "social networking" com tecnologias de sensoriamento móveis para permitir ao utilizador adquirir informações sobre eventos e estabelecimentos, em tempo real, através dos seus dispositivos móveis utilizando os serviços disponibilizados pela operadora de serviços e telecomunicações de Portugal SAPO (SAPOSERVICES, 2014). O sistema será usado no contexto de diversão noturna, ajudando os seus utilizadores a mais facilmente escolher os melhores locais de entretenimento fundamentado nos requisitos, regras, critérios e modelos de privacidade para ambientes pervasivos.

Caso o usuário se encontre no meio de vários ambientes, este poderá escolher a qual ambiente pertence, ficando a cargo do ambiente pervasivo escolhido solicitar as informações de *login* e de registo ao ambiente ubíquo disponível para aquele ambiente.

Caso o usuário se encontre num ambiente “público” (e.g. rua, campo deserto), o usuário faz autenticação diretamente com o ambiente ubíquo (utilizando um “Access Point” público ou GPRS) e assume o perfil de “*advanced*”, ou seja, pode executar as tarefas que quiser e configurar o seu dispositivo como bem entender, podendo também requisitar informações gerais sobre os ambientes que lhe são mais próximos, conforme apresentado na Figura 4.9 onde os ambientes são divididos em regiões (A, B, C e D), cada uma com critérios e definições diferentes para o mesmo usuário. Esses critérios e definições podem variar de acordo com horário, cargo ocupado, dias da semana, entre outras variáveis:

Figura 4.9 - Conectividade no ambiente



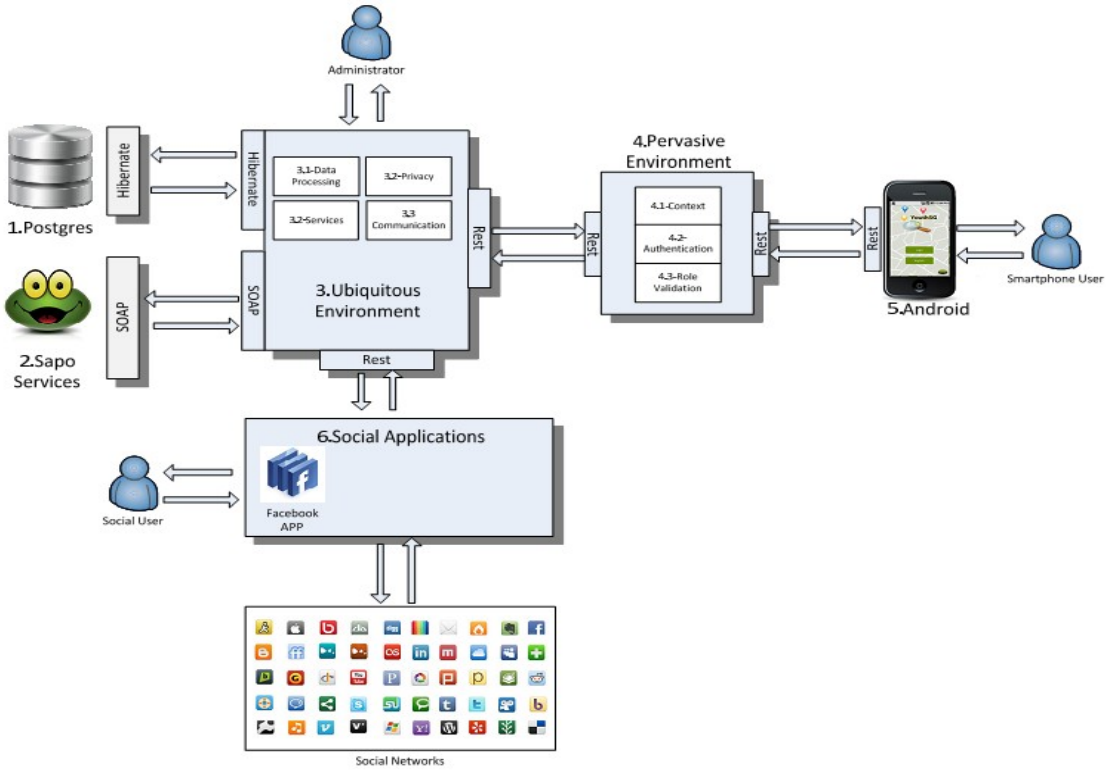
Fonte: Próprio autor.

O usuário poderá não estar dentro de um determinado ambiente tendo seus serviços e critérios reduzidos conforme apresentado na Tabela 4.1 Neste caso a autenticação não é dependente do ambiente pervasivo e sim do dispositivo que o usuário possui. Na próxima subseção serão apresentadas as definições da arquitetura da aplicação.

Para realização dos testes iniciais e validação do modelo genérico fundamentado na taxonomia e definições apresentadas anteriormente, elaboramos uma arquitetura de funcionamento da aplicação YOUTH5G. Essa arquitetura possui todos os módulos necessários para gerenciamento e controle do usuário, dos ambientes pervasivo e ubíquo, bem como as funcionalidades dos dispositivos e parâmetros de base de dados e regras. Também são apresentadas as ligações relacionadas aos serviços disponibilizados pela SAPO PT (SAPOSERVICES, 2014) e conexões relacionadas às redes sociais.

A arquitetura do sistema YOUTH5G é constituída pelos diferentes elementos da aplicação conforme pode ser visualizado na Figura 4.10. As funcionalidades e descrições de cada módulo são apresentadas da seguinte forma:

Figura 4.10 - Arquitetura de aplicação



Fonte: Leithardt et al. (2013a).

1. *DB*: A base de dados do sistema irá ser a entidade responsável por armazenar a informação do sistema e irá comunicar com os restantes elementos através do framework Hibernate. Este framework permite um mapeamento objeto relacional entre a base de dados e o restante do sistema, diminuindo a complexidade do sistema principal;
2. Sapo Services: Um dos parceiros importantes neste projeto (SAPOSERVICES, 2014), que disponibilizou os seus serviços de geolocalização. Estes serviços irão ser utilizados pelo sistema com diversos usos em diferentes eventos existentes em Portugal, onde inicialmente a aplicação será utilizada. Sendo úteis principalmente para referenciar os eventos noturnos, de acordo com as atividades do usuário ou ambiente pervasivo/ubíquo;
3. *Ubiquitous Environment*: tem a função de receber, controlar e registrar as informações recebidas de usuários e ambientes pervasivos. Essas informações são repassadas à base de dados e consequentemente são registradas e atualizadas, tanto de solicitações recebidas de usuários quanto de ambientes pervasivos;
   3.1 *Data Processing*: neste módulo os dados do usuário, ambiente pervasivo e ubíquo são cadastrados, mantidos, controlados e atualizados;

3.2 Privacidade: Neste módulo se encontram as definições e regras de processos de controle privacidade, bem como as características individuais dos usuários e ambientes pervasivos;

4. Serviços: Este componente é o responsável por disponibilizar os serviços que serão utilizados tanto pela aplicação rodando em *Android*, como pelas aplicações das redes sociais. Estes serviços serão disponibilizados em forma de Web Services do tipo SOAP e REST, para, por exemplo, a comunicação em aplicações de redes sociais e do ambiente pervasivo;

4.1 Comunicação: Este componente do ambiente ubíquo é o responsável pelo processamento de todos os dados e pedidos que cheguem ao ambiente ubíquo. Ele irá tratar os pedidos que chegam pelo componente "services", através da interação com os outros componentes existentes, mais propriamente Sapo services, redes sociais, hibernante, etc;

4.2 *Pervasivo*: Nesse módulo são processadas as informações recebidas do usuário, bem como o dispositivo utilizado para efetuar o *login* no sistema. Nesse módulo também estão definições e regras de utilização de usuários e dispositivos, bem como as informações do próprio ambiente pervasivo.

4.3 Contexto: O componente "*context*" é o componente responsável por analisar e identificar o contexto onde o usuário está inserido, permitindo assim ao ambiente pervasivo associar as regras ao utilizador em um dado ambiente;

4.4 Autenticação: Este componente é responsável pela identificação dos usuários no sistema;

4.5 *Role Validation*: esse módulo funcionará da seguinte forma, cada utilizador tem um tipo (ex: avançado, convidado, etc.) que está associado a um determinado ambiente. Este componente tem como função em cada ambiente, definir os diferentes tipos de utilizadores, sendo que cada tipo irá ter regras diferentes;

5 *Android*: O sistema Youht5G possui uma aplicação para ambientes móveis, mais propriamente para a plataforma *Android*, inicialmente. Esta plataforma irá comunicar diretamente através de pedidos a web services no ambiente pervasivo, que por sua vez irá comunicar com o ambiente ubíquo. Através desta aplicação, o utilizador terá a possibilidade de ver os diferentes ambientes noturnos, listas de músicas a serem tocadas e partilhar diferentes tipos de informações como atividade, afluência de pessoas, etc;

6 Social: A plataforma faz interação com diferentes redes sociais, existindo a possibilidade de expansão no futuro para outras redes sociais, de modo a poder existir uma maior agregação de conhecimento vindo das redes sociais. Na rede social Facebook (Facebook, 2015) é construída uma aplicação que permite os donos/djs do bar noturno colocar a lista de músicas que será depois associada a um determinado evento e deste modo poderá ser visualizada pelo usuário no seu dispositivo móvel. Por outro lado, pretende-se também através desta rede social capturar os diferentes gostos musicais, interesses, etc. dos usuários, para poder indicar na aplicação quais os locais noturnos que são mais interessantes de acordo com seu perfil.

A contribuição do cenário e caso de uso apresentado em relação a tese é o tratamento de privacidade necessária em determinados ambientes ubíquos ao qual usuários que utilizam a aplicação YOUTH5G se encontram, principalmente relacionando outras aplicações e redes sociais. O módulo 3.2 (privacy) da Figura 4.10 possui funcionalidades que estão relacionadas ao UbiPri, tais funcionalidades podem ser desde a ativação de serviços com base na localização até o controle de acessos ou funcionalidades de dispositivos de acordo com as regras e critérios de cada ambiente.

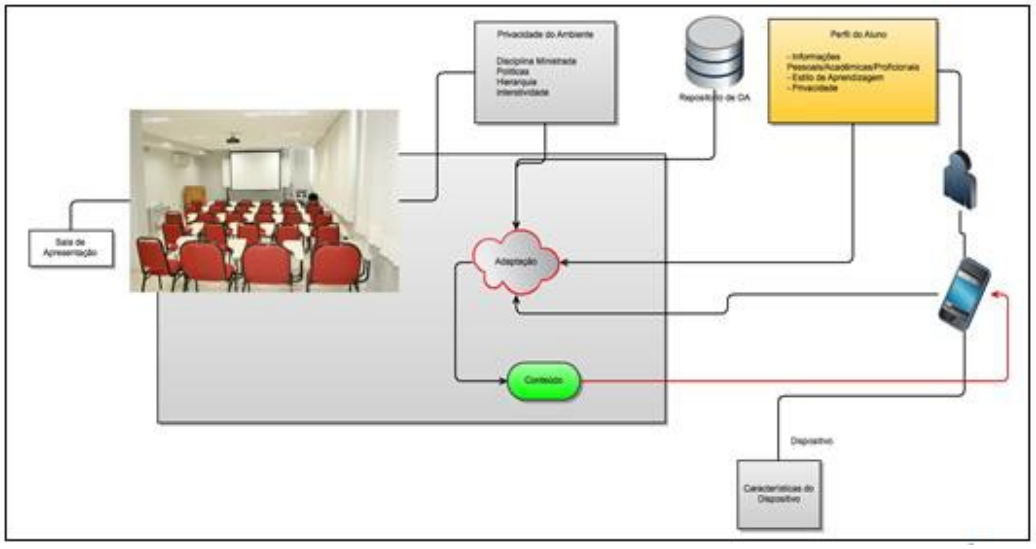
4.6.3 Cenário de aplicação 3

A solução GVwise utiliza técnicas de Big Data, mais especificamente técnicas de descoberta de conhecimento em bases de dados, inteligência computacional e de gerenciamento de dados, para a identificação precoce dos alunos pertencentes ao grupo de risco de evasão ou de baixo desempenho acadêmico. A versão atual da solução possibilita aos educadores uma fácil e precoce identificação dos estudantes que demandam apoio diferenciado, já que nem sempre as dificuldades dos alunos são de natureza meramente acadêmica. Conforme Cambruzzi et al. (2012), o software permite a realização e registro de ações por parte dos educadores e exibe graficamente dados que dão suporte à realização das ações de combate à evasão. Em outras palavras, o GVwise é uma solução de apoio aos educadores no processo de gestão e reversão da evasão e do baixo desempenho educacional.

Os resultados mensurados até o momento indicam que o GVwise é capaz de identificar, com precisão média de 80%, aqueles alunos pertencentes ao grupo de risco, além de que sua utilização por parte dos educadores é capaz de reduzir ao menos 25% a evasão de instituições de ensino superior. A Figura 4.11 apresenta a estrutura da solução proposta que

utiliza como base o armazenamento de informações de alunos, professores e demais usuários do sistema.

Figura 4.11 - Descrição da aplicação.



Fonte: Próprio autor.

O cenário apresentado na Figura 4.11 tem o funcionamento iniciado pela localização do usuário que esta em uma sala de apresentação, a localização é feita com uso de seu dispositivo móvel com base no perfil do aluno parâmetros de recebimento de mensagem são ativados. A contribuição do UbiPri se encaixa na filtragem do local e definições de critérios para o recebimento de mensagem. O sistema neste caso uma sala de apresentação possui informações de contexto que determinam que naquele local e hora não seja permitido barulho, neste caso o dispositivo do usuário poderia receber automaticamente a mensagem em modo silencioso ou exibi – lá em outro momento e/ou local. O recebimento das mensagens e ligações estão relacionados aos perfis de critérios de ambiente principalmente, mas também de acordo com o perfil do usuário naquele local, por exemplo, um usuário no mesmo local e hora com perfil avançado poderia receber a mensagem se assim fosse configurado por se tratar de um usuário Professor por exemplo, lembrando que o ambiente é que deve ser configurado para ditar as regras de utilização com base nos grupos de usuários e perfis que nele estão. A integração do cenário proposto a tese se relaciona a Tabela 4.1 onde são apresentados os critérios e recursos definidos. Também é possível utilizar os dados relacionados a base de dados do modelo UbiPri conforme a Figura 4.7, utilizando apenas alguns módulos necessários. A validação desse cenário de acordo com as regras e teorias

aplicadas ao cenário e modelo proposto foram publicadas no trabalho desenvolvido por Cambruzzi et al. (2012).

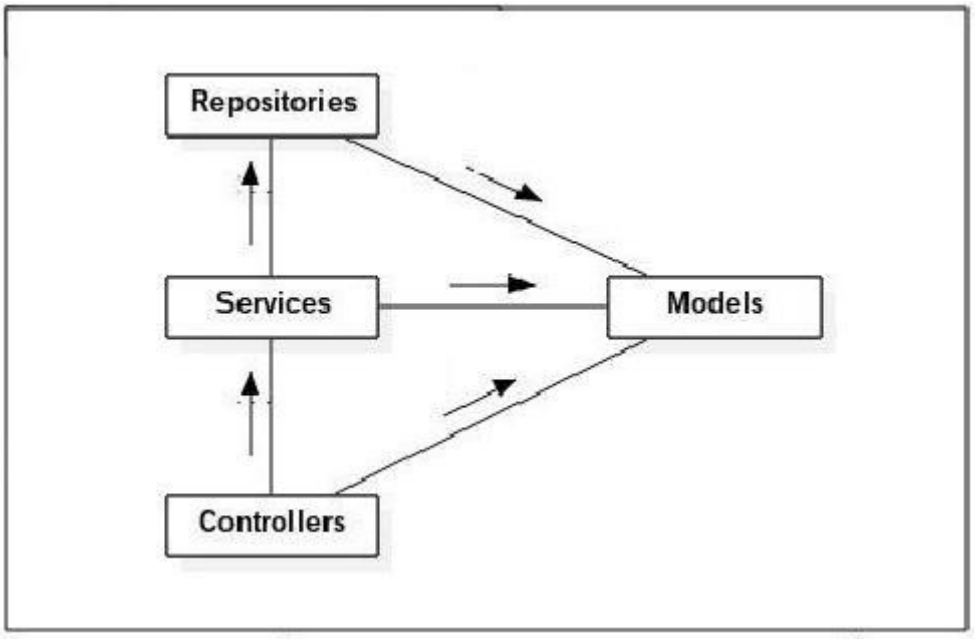
4.6.4 Cenário de aplicação 4

Este cenário de aplicação foi desenvolvido no projeto de pesquisa e desenvolvimento de um Sistema de controle e Gerenciamento Acadêmico Inteligente para ambientes de ensino (SIGA-i). O SIGA-i tem como objetivo o desenvolvimento de soluções que visam a otimização de recursos e a facilidade na troca de informações internas da instituição voltado a todos os níveis de ensino ofertados, bem como, o desenvolvimento de pesquisa científicas com base nos dados e ferramentas de solução encontradas tendo caso de uso a Faculdade de Tecnologia SENAI Porto Alegre.

O projeto SIGA-i também busca e identifica as principais demandas relacionadas ao funcionamento da instituição, e desenvolver soluções, com foco tecnológico, para a otimização de recursos e processos para facilitar a troca de informações. Com base nas soluções desenvolvidas, será possível introduzir novas tecnologias como: RFID, sensores, entre outras, com intuito de introduzir e desenvolver pesquisa de avançada na instituição.

A estrutura do SIGA-i é composta em camadas que estruturam todo o sistema, a Figura 4.12 representa o padrão seguido para o desenvolvimento da aplicação do SIGA-i, identificando suas camadas, *Model, Repository, Service e Controller,* respectivamente

Figura 4.12 - Integração e relacionamento entre as camadas de aplicação



Fonte: Próprio autor.

O relacionamento entre as classes da aplicação funciona da seguinte forma: A camada dos ***Controllers*** fica com uma responsabilidade de ser o intermediador, ou seja, ele recebe a requisição que contém a lógica de aplicação e de entrada do usuário. Logo em seguida o serviço passa para a camada ***Services***, esta é basicamente o middleware entre o *Controller* e o *Repository*, a mesma coleta dados do controlador e executa a validação, seria a lógica de negócio da aplicação. A partir dai se transfere o trabalho para os ***Repositories***, que a partir de seus métodos se realiza então o acesso através de seus respectivos métodos na classe dos ***Models*** e utiliza sua arquitetura interna para realizar serviços com a base de dados da aplicação. Neste cenário utilizado foi possível utilizar os módulos do middleware apresentado na Figura 4.1, em especial os módulos de Hardware por terem utilizado para identificação RFID entre outros dispositivos para identificação. Também possível o desenvolvimento de um aplicativo móvel utilizando o sistema operacional Android disponível em UbiPri (2015). A principal contribuição do cenário e aplicação foi o bloqueio de funcionalidades dos dispositivos móveis de usuários de acordo com o local, horário e ambiente em que se encontravam, os resultados parciais foram necessários para validar o relacionamento entre os módulos do modelo de controle e gerenciamento de privacidade.

5 PROTÓTIPO, TESTES E RESULTADOS OBTIDOS

Neste capítulo serão apresentados o modelo do protótipo desenvolvido e suas definições, funcionalidades e características necessárias para seu funcionamento. Posteriormente são apresentados os testes preliminares realizados e os resultados obtidos com base no protótipo desenvolvido aplicado a um cenário de aplicação de acordo com as necessidades de controle e gerenciamento de privacidade.

5.1 Protótipo

O protótipo proposto foi desenvolvido parcialmente em um servidor remoto que atua como autoridade sobre os dispositivos e ambientes em questões de privacidade, o mesmo possui as regras, definições e critérios necessários. O protótipo implementado, ilustrado na Figura 5.1, apresenta o cenário utilizado, onde o servidor de contextos de privacidade (4) atua como autoridade perante o dispositivo móvel (2) e recebe as localizações simbólicas inferidas através das localizações físicas (1) dos dispositivos móveis que os usuários carregam. Cada dispositivo móvel pertence a um usuário por vez enquanto em uso, podendo o mesmo dispositivo ser utilizado por mais de um usuário em tempos diferentes.

O protótipo se fundamenta também nos resultados obtidos e descritos nos trabalhos futuros abordados em Silva et al. (2009). Ao receber a localização simbólica por algum meio de localização processada pelo dispositivo móvel do usuário, o servidor passa pelo processo detalhado na seção anterior que descreve a checagem, validação e as alterações que devem ser feitas para o novo contexto do usuário em relação à privacidade do ambiente, efetuando-as via comunicação (3). Para os testes, o sistema de localização conta com um usuário e cinco ambientes, somente efetuando modificações e adaptações no dispositivo móvel do cliente conforme apresenta a Figura 5.1.

O servidor do modelo proposto foi implementado utilizando a linguagem de programação Java EE, suportando atualmente comunicações por *WebService* REST, Google *Cloud Messaging* (GCM) e Comunicação Serial. Também foi utilizado um banco de dados relacional PostgreSQL. A programação do dispositivo móvel foi em *Android* e suas APIs para acesso a informações de localização e para envio e recebimento de mensagens.

Figura 5.1 - Protótipo implementado

Fonte: Próprio autor.

O protótipo desenvolvido possui atualmente as seguintes funcionalidades: 1) Efetuar trocas de ambiente utilizando meios de localização como GPS e Tags NFC a partir do cliente móvel; 2) Identificar que acesso, em termos de privacidade, um usuário pode ter em um dado ambiente; e 3) Habilitar ou desabilitar funcionalidades do smartphone de acordo com a privacidade exigida pelo ambiente.

Inicialmente para que o processo seja realizado foram submetidas quatro fases de processamento no servidor.

- Primeiramente a autenticação do usuário e do dispositivo no serviço.
- Posteriormente a identificação da relação do usuário com o ambiente, o qual é expresso por um perfil de usuário no dado ambiente.
- Por terceiro a classificação da permissão de acesso do perfil do usuário no dado tipo de ambiente, as definições que estão na base de dados do UbiPri.
- Por último a geração das ações a serem executadas nos *smartphones* em duas etapas, ações gerais do tipo de ambiente e as ações específicas customizadas para o ambiente visado.

A autenticação fez-se necessária, primeiramente para identificar a existência do usuário no serviço e sua permissão de uso do dispositivo. Caso o usuário ou o dispositivo não existam, esse processo é encerrado dando um retorno de acesso negado, contudo o servidor possibilita o cadastro de ambos através da aplicação móvel permitindo, assim, o seu uso efetivo.

Em segundo lugar, a identificação da relação do **usuário** com o **ambiente** foi um dos problemas encontrados, visto que cada usuário pode conter diferentes definições em um mesmo ambiente sendo considerada apenas uma premissa básica como o tempo em que foi

identificado. Com isso, em parte deve ser atribuída automaticamente e em parte manualmente as definições, visto que alguns critérios não mudam de acordo com tipo de usuário e ambiente identificado. Assim, foram definidos nos testes realizados seis perfis de usuários possíveis, os quais expressam as duas premissas anteriores que são a localização e os critérios atribuídos para os casos de uso utilizados, considerando que cada ambiente possui diferentes interações para cada usuário foram especificadas algumas definições de usuários tendo por base um caso de uso educacional, sendo elas:

a) *unknown*: Usuário é desconhecido para o ambiente;

b) *transient*: Perfil de pessoa que acessa somente de passagem pelo ambiente ou é um visitante temporário;

c) *user*: Perfil de usuário que interage com o ambiente mais intensamente, estando frequentemente presente no ambiente por um tempo considerável ou utilizando serviços disponibilizados pelo ambiente;

d) *responsible*: Responsável ou funcionário do local, possui mais permissões que os usuários e afins;

e) *student*: Perfil de aluno, o qual possui permissões diferentes dos usuários comuns e dos funcionários;

f) *manager*: É a autoridade máxima do ambiente, possuindo assim o máximo de permissões e pode adicionar, remover e alterar usuários, além de poder modificar o perfil no ambiente de cada um deles.

Os três primeiros perfis (*Unknown, Transient e User*) devem ser automaticamente identificados e processados pelo sistema, enquanto os três últimos (*Responsible, Student e Manager*) são atribuídos manualmente pelo gerente do ambiente, o qual faz uso de um gerenciador de conteúdo que exige autenticação. Esta configuração faz-se necessária, pois com um número grande de usuários e ambientes, provavelmente usuários não acessarão todos os ambientes conhecidos pelo sistema. Desta forma o próprio sistema pode classificar usuários comuns ou novos enquanto em execução, dispensando assim configurações por parte do gerente do sistema. Em contra partida, usuários comuns não devem acessar todos os recursos especialmente em ambientes privados, nem serem progredidos automaticamente, para tanto, por questões de segurança, por exemplo, um aluno não deve acessar o gabinete de professor sem permissão, sendo assim, são perfis que devem ser atribuídos para cada usuário manualmente.

Depois de identificado o perfil do usuário é possível classificar que tipo de acesso o usuário possui nesse ambiente, foram primeiramente, identificadas as variáveis que implicam

no tipo de acesso do usuário no ambiente, sendo elas: o perfil do usuário no ambiente identificado na etapa anterior, o tipo de ambiente sendo acessado, o dia da semana entre dia da semana e final de semana, o turno entre diurno e noturno e se o dia é útil ou não. Importante, esclarecer que o objetivo foco desta tese não é o tratamento de perfis de usuários, ambientes, dispositivos, comunicação ou qualquer outro componente necessário para consciência de contexto, e sim para o controle e gerenciamento de privacidade. Para tanto, é necessário especificar regras e definições de perfis para validar o modelo de privacidade desenvolvido, entre outros quesitos abordados. Um exemplo de classificação utilizando tais atributos pode ser encontrado na tabela 5.1, onde cada atributo pode assumir os seguintes valores:

- *Attribute* 1: Representa o perfil do usuário dentre os seis acima citados ((a) ao f) no processo anterior;
- *Attribute* 2: Representa o tipo de ambiente dentre o qual foram considerados três tipos: bloqueado, privado e público.
- *Attribute 3*: Dia de semana o qual assume o valor binário *true* (Week), para dia da semana, e *false* (Weekend), para finais de semana.
- *Attribute 4*: Representa o turno em que o acesso está ocorrendo no ambiente, sendo considerados dois turnos, diurno e noturno
- *Attribute* 5: Por fim, este atributo indica se o dia é útil ou não em relação ao local, com base no dia da semana, visto que existem feriados e dias que não há expediente de trabalho para certos ambientes.

Tabela 5.1 - Exemplo de classificação do tipo de acesso

Class (Output)	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5
Blocked	Transient	Private	Week	Daytime	Yes
Administractive	Transient	Public	Week	Daytime	Yes
Blocked	Transient	Blocked	Week	Daytime	Yes

Fonte: Próprio autor.

Em segunda instância, após identificar as variáveis o classificador gera uma saída representada na Tabela 5.1 pela coluna **Class (Output)**. Esta saída representa o tipo de acesso que o usuário possui no ambiente ao qual está acessando. Os valores possíveis para esse tipo de acesso são fundamentados nos critérios apresentados na Seção 3.4, sendo eles: Bloqueado, Convidado, Básico, Avançado e Administrativo, no entanto foram validados nos testes os

critérios de usuário privado, administrativo e bloqueado. Esta combinação entre os valores possíveis entre os diferentes atributos e saídas gerou um arquivo de treino com 144 possibilidades de classificação. A intenção é usar um classificador no servidor a fim de classificar o nível de acesso de usuários no sistema e com isso tomar a decisão sobre as adaptações possíveis e autorizar o acesso aos recursos permitidos ao usuário, com isso, o sistema automaticamente classificaria o usuário. Nosso critério de avaliação buscava um classificador que alcançasse 100 por cento de acurácia, na literatura pesquisada, resultados próximos a 100 por cento dependem de muitas variáveis que modificam de acordo com o cenário da aplicação dentre outras definições.

Nos testes iniciais foram utilizados sete algoritmos à base de treino descrita, tais algoritmos estão disponíveis e são amplamente testados e utilizados por pesquisadores na ferramenta Weka (WEKA, 2015). Os resultados são apresentados na Tabela 5.2. Pode-se observar que os algoritmos *DecisionTable*, *Bayes Network*, J48 e BF-Tree não conseguiram atender ao requisito de precisão de 100 por cento, enquanto os algoritmos *RandomTree*, *NNge* e *MultilayerPerceptron* alcançaram os resultados satisfatórios.

Tabela 5.2 - Comparação entre algoritmos de classificação

Algoritmo de Classificação	Precisão	Instâncias Corretas	Instâncias Incorretas
Decision Table	0.928	133	11
Bayes Network	0.933	135	9
J48	0.974	140	4
Best-First Decision Tree (BT-Tree)	0.975	140	4
Random Tree	1.0	144	0
Nearest Neighbor With Generalization (NNge)	1.0	144	0
Multilayer Perceptron	1.0	144	0

Fonte: Próprio autor.

Também foram definidas regras para progressão dos perfis evolutivos, as mesmas são configuráveis no próprio sistema. Para tanto, foi utilizada a frequência (F) do usuário (u) para definir quando o mesmo deve evoluir para um perfil (P) mais avançado do ambiente, ou ser regredido para um perfil menos permissivo. Neste caso, as regras implantadas definem, para cada ambiente (a), quais limites de frequência inferior (I) e superior (S) nos quais a mudança de perfil deve ser efetuada. Considerando que o ambiente possui n perfis evolutivos, a equação descrita a seguir representa a regra de progressão implantada que tempo finalidade

definir a evolução automática de um perfil de usuário em um ambiente, de acordo com critérios e parâmetros estabelecidos.

$$P_{\diamond\diamond,a} = \{P_{\diamond\diamond,a} + 1\ , i\diamond\ F_{\diamond\diamond,a} > \diamond\diamond_a\ ;\ P_{\diamond\diamond,a} - 1\ , i\diamond\ F_{\diamond\diamond,a} < \diamond\diamond_a\ ;\ P_{\diamond\diamond,a}\ , \diamond\diamond h \diamond\diamond w \diamond\diamond\diamond\diamond\} \ \forall$$

$$P \in [1, \diamond]$$

Para os perfis não evolutivos, a frequência também é utilizada para aumentar ou diminuir o nível de acesso dos usuários em determinados ambientes. Contudo, o perfil do usuário mantém-se o mesmo e apenas o tipo de acesso é alterado. Em ambos os casos, considerou-se que a frequência pode assumir três níveis distintos: frequente, normal e infrequente. Para tanto, foram considerados, ainda: três tipos de ambiente: restrito, privado e público; para o dia de semana assume o valor binário true, e false, para finais de semana; foram considerados dois turnos, diurno e noturno; por fim, a variável dia útil indica se um dia é útil ou não em relação ao local baseado no dia da semana, visto que existem feriados e dias que não há expediente de trabalho para certos ambientes. A combinação de todas as variáveis que constituem casos possíveis no cenário estudado, resultaram em 383 possibilidades de classificação.

Em segunda instância, após identificar as variáveis, foi atribuído a cada uma das possíveis combinações um dos seguintes tipos de acesso: Bloqueado, Convidado, Básico, Avançado e Administrativo. Esses dados foram utilizados para treinamento e testes em sete diferentes algoritmos de classificação conforme apresentado na Tabela 5.3, a fim de identificar aquele com a maior precisão. Esses experimentos também foram utilizados para selecionar o algoritmo que será utilizado pelo servidor para classificar automaticamente o nível de acesso dos usuários em ambientes desconhecidos e não configurados no sistema, ou seja, onde não há todas as regras do ambiente bem definidas. Para esses casos atribui se o nível mínimo de privacidade ou o bloqueio do usuário no ambiente de acordo com os parâmetros definidos.

O experimento comparativo entre os algoritmos de classificação foi realizado com uso da ferramenta Weka (Hall, 2009; Holmes et al., 1994). A tabela com as regras (combinação de todos os atributos e seus respectivos tipos de acesso resultantes) foi dividida em conjuntos de treinamento e teste, através da técnica de validação cruzada com dez subconjuntos (10-*fold cross-validation*) conforme Kohavi (1995), onde 90% dos dados é utilizado para treinamento dos algoritmos de classificação, e os 10% restantes são utilizados para verificação dos

resultados destas regras (desconhecidas pelo classificador). Neste método, ainda, o conjunto

de teste é variado entre todos os possíveis subconjuntos dos dados de treinamento. A precisão final é apresentada na Tabela 5.3, sendo obtida através da média dos testes realizados, conforme segue:

Tabela 5.3 - Segunda comparação entre algoritmos de classificação

Algoritmo de Classificação	**Precisão**	**Instâncias Corretas**	**Instâncias Incorretas**
DecisionTable	0.887	343	40
Bayes Network	0.814	322	61
J48	0.887	341	42
Best-First Decision Tree (BT-Tree)	0.871	336	47
RandomTree	0.861	332	51
Nearest Neighbor With Generalization (NNge)	0.848	326	57
MultilayerPerceptron	0.888	341	42

Fonte: Próprio autor.

Uma vez obtido o tipo de acesso no ambiente é possível identificar que ações devem ser enviadas ao dispositivo utilizado pelo usuário, iniciando pelas ações padrão para cada funcionalidade e também considerando as regras padrão e o tipo de ambiente. Cada instância corresponde há uma funcionalidade que pode ser atribuída ao usuário, dispositivo e que varia de acordo com o dia da semana, turno, localização, etc. que podem ter variações. Para tanto é considerado um nível de acesso que é resultante do tipo de acesso do usuário no ambiente e do tipo de ambiente acessado. Um exemplo dessa combinação por funcionalidade pode ser encontrado na Tabela 5.4, onde as siglas utilizadas são:

Para Tipo de Ambiente:

- block = Bloqueado (Blocked);
- priv = Privado (Private);
- public = Público (Public).

Tipo de Acesso:

- block = Blocked;
- guest = Guest;
- basic = Basic;
- adv = advanced;
- adm = Administrative.

Exemplos de ações para cada funcionalidade:

– off: Ação enviada para a funcionalidade do dispositivo é apontada como desligada;
– on: Ação enviada para a funcionalidade do dispositivo é apontada como ligada;
– none: Nenhuma ação é enviada ao dispositivo, prevalecendo as preferências do dispositivo.

Tabela 5.4 - Regras de privacidade padrão do ambiente

	Nível de Acesso	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	Tipo de Ambiente	**block**	**block**	**priv**	**priv**	**priv**	**Priv**	**priv**	**public**
	Tipo de Acesso	**block**	**adv**	**block**	**guest**	**basic**	**Adv**	**adm**	**adm**
N	***Funcionalidade***	-	-	-	-	-	-	-	-
1	Bluetooth	off	none	off	**on**	**on**	none	none	none
2	Silent Mode	**on**	none	**on**	**on**	**on**	none	none	none
3	Vibrate Alert	**on**	none	**on**	**on**	none	none	none	none
4	Airplane Mode	off	none	off	off	off	off	none	none
5	Wi-Fi	**on**	**on**	**on**	**on**	**on**	**on**	**on**	**on**
6	Mobile Network Data Access	off	none	off	**on**	**on**	**on**	none	none
7	System Volume	off	none	off	none	none	none	none	none
8	Media Volume	off	none	off	none	none	none	none	none
9	Ringer Volume	off	none	off	none	none	none	none	none
10	Screen Timeout	off	none	off	none	none	none	none	none
11	Screen Brightness	off	none	off	none	none	none	none	none
12	SMS	off	none	off	**on**	**on**	none	none	none
13	Launch App	off	none	off	**on**	**on**	none	none	none
14	Camera Access	off	none	off	off	off	**on**	none	none
15	GPS	**on**	none	**on**	**on**	**on**	**on**	none	none

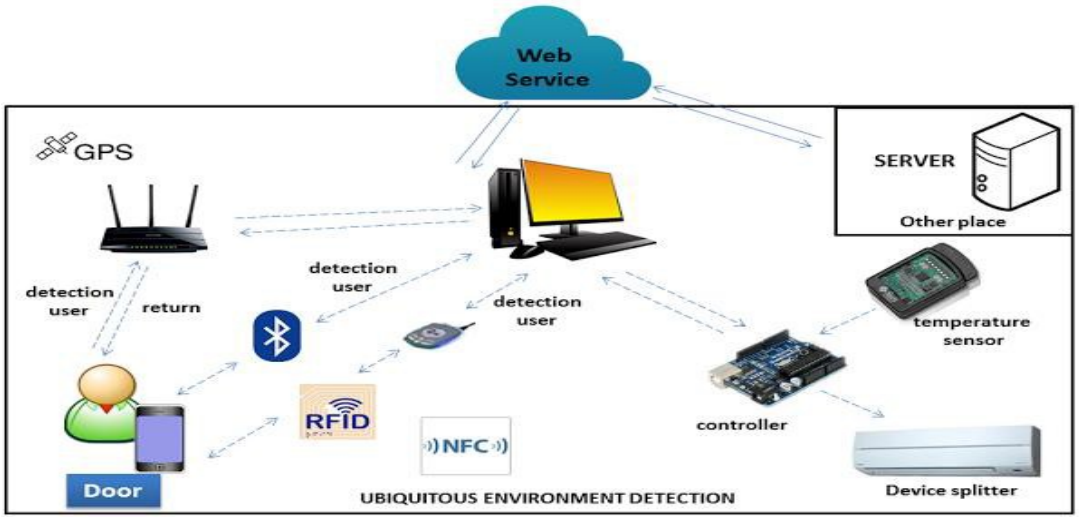
Fonte: Próprio autor.

Depois de gerada a lista de ações padrão do tipo de ambiente para serem enviadas ao dispositivo, é executada uma segunda etapa de seleção de ações que sobrescreve as regras padrões por regras customizadas pelo gerente do ambiente. Um exemplo de regra customizado seria um aluno novo que fora classificado como convidado e como regra customizada ele pode entrar no gabinete de determinado professor por determinado tempo, contudo ele continua a não pode utilizar determinados dispositivos, usufruir de serviços oferecidos no ambiente pelo tipo critério que ele está relacionado, neste caso convidado.

Para tanto uma regra é associada a um nível de acesso e a um ambiente, apenas reescrevendo em tempo de execução na aplicação a regra default. Para gerenciar tais ações customizadas é interessante construir um gerenciador de conteúdo que possa automatizar. A base de dados contém informações de todos os ambientes pervasivos e tem a função de armazenar informações para serem gerenciadas pelo ambiente ubíquo. Vejamos outro exemplo: Uma universidade é composta por diversos campi, a Universidade como um todo possui informações individualizadas de todas as suas faculdades que estão relacionadas, ou

seja, domínio principal, com base nesse domínio principal foi denominado como ambiente ubíquo, sendo assim, dentro desses ambientes há regras, definições e critérios diferenciados. Esse ambiente ubíquo possui vários ambientes e dispositivos integrados e relacionados a ele devido à mobilidade que existe conforme apresentado na Figura 5.2. Para controlar esses ambientes o modelo gerenciador de privacidade possui vários módulos integrados cada qual com suas funcionalidades e responsabilidades, conforme a identificação e características apresentadas na Figura 5.2.

Figura 5.2 - Descrição ambiente pervasivo



Fonte: Leithardt et al. (2012).

Conforme a descrição do ambiente pervasivo apresentado na Figura 5.2, o funcionamento ocorre da seguinte forma:

1: O usuário (*Door*) é detectado no ambiente ubíquo através de informações que chegam ao módulo de controle e gerenciamento de dados. Esse procedimento é possível devido à utilização e integração do módulo gerenciamento de monitoração que compõe o *middleware* híbrido, conforme descrito no capítulo 3, especificamente o modelo de privacidade apresentado na Figura 3.7.

Passo 1.1: O *Control Module* (CM) verifica o cadastro do solicitante através de validação na base de dados, mas antes de retornar a liberação do acesso é realizada uma série de verificações no ambiente pervasivo em que o mesmo se encontra, a fim também de atualizar dados na base, como disponibilidade de ambientes pervasivos, serviços disponíveis, localizações, etc.

2: É emitida uma solicitação de verificação e validação ao módulo de dados (DATA MODULE “DM”) com a finalidade de obter os critérios e definições do ambiente onde o usuário foi detectado, para tanto essas solicitações são realizadas através da conexão utilizando o DATA INFORMATION que possui apenas a finalidade de estabelecer a conexão entre o DATA CONTROLLER e o DATA MODULE.

3: O módulo de dados (DM) possui funções e cálculos matemáticos necessários para enviar ao CM com todas as definições e regras que foram obtidas através de variações de outros módulos que compõem o modelo gerenciador.

3.A: Ao receber a solicitação do DM é enviada simultaneamente uma solicitação de atualização ao módulo PRICMU, que tem como função controlar informações de usuários.

3.B: A outra solicitação é enviada ao módulo PRIENV que tem como função controlar informações do ambiente pervasivo. Então é realizada a atualização de dados do ambiente.

3.A: Recebidos os dados no módulo PRICMU, que possui as informações relacionados ao usuário, este por sua vez encaminha-os ao módulo PRICOM de controle de comunicação.

3.A.1: É verificado e atualizado o tipo de comunicação que foi utilizado para a solicitação.

3.A.2: É repassado ao próximo módulo PRIDEV (que faz o controle de dispositivos) para que seja identificado e atualizado o tipo de dispositivo da solicitação.

3.A.3: Após a identificação do dispositivo, os dados são encaminhados ao módulo PRIPRO que trata do perfil e contexto. É validado o perfil de acordo com os dados recebidos de usuário + comunicação + dispositivo para que sejam realizadas as atualizações necessárias e, por conseguinte, encaminha-los ao próximo módulo de controle.

3.A.4: Com base nos dados recebidos dos módulos anteriores são identificadas as possíveis adaptações para o usuário + comunicação + dispositivo utilizado + perfil e encaminhadas ao módulo de dados para que seja realizado o processamento das possibilidades de definições e critérios que se aplicam de acordo com as variações que possam ter em diferentes dias, horários, usuários entre outras variáveis.

3.B: Recebidos os dados no módulo PRIENV, este é atualizado da situação, repassando suas disponibilidades relacionadas ao ambiente ao módulo posterior onde são definidos os critérios de utilização do ambiente pervasivo.

B.1: É verificado e atualizado o tipo de critério de acordo com o ambiente que enviou à mensagem de atualização, depois de feita a atualização de critérios e definições é encaminhado para o módulo PRIHIS, em desenvolvimento.

B.2: no módulo PRIHIS são armazenados os históricos de informações. Após o recebimento dos dados do módulo de critérios, este módulo verifica quais foram os históricos de utilização do ambiente e critérios definidos conforme atualizações realizadas nos módulos responsáveis anteriores como, por exemplo: poderia ter informações de utilização de determinado ambiente fora do padrão em outros períodos, ou em relação a outros usuários do mesmo ambiente, podendo ser estendido o uso de mecanismo de controle de trilhas e/ou confiança a fim de verificar também a possibilidade de possíveis fraudes e questões relacionadas à segurança. Para tanto, os dados atualizados são encaminhados ao módulo PRISEC.

B.3: no módulo de controle de segurança PRISEC são tratadas questões e parâmetros definidos anteriormente com base em cada ambiente, critérios e históricos recebidos, com isso são geradas informações necessárias para que sejam disponibilizados serviços ao solicitante. Segundo Hübner (2012), um serviço pode ser a utilização de recursos do ambiente, o compartilhamento de arquivos e dados no ambiente e entre os usuários que ali estão níveis de privacidade controlada e exigida, etc. Essas informações são repassadas ao módulo PRISER que trata do gerenciamento de privacidade e serviços.

B.4: após serem realizados todos os tratamentos necessários dos dados nos módulos anteriores, o módulo PRISER realiza a verificação dos serviços disponíveis com base nas definições estipuladas em cada módulo anterior. Com isso, repassa ao módulo de dados as variáveis de serviço com base no ambiente identificado + critérios definidos + histórico + segurança + serviços.

Relembrando que conforme descrito na seção 4.4 modelo de privacidade, os módulos atuam de forma independente com características e funcionalidades próprias que podem variar de acordo com as regras previamente estabelecidas, cadastradas e gerenciadas. Uma vez definidas essas regras, cada módulo estabelece os parâmetros com base nas definições do módulo anterior. Sendo assim, para um mesmo ambiente ubíquo é possível ter vários ambientes com regras e definições diferentes. E um mesmo usuário poderá utilizar um ou vários ambientes diferentes cada, tendo um critério definido que pode mudar com base no dispositivo utilizado, na comunicação, entre outros quesitos que serão calculados no módulo de dados.

5.2 Arquitetura e Funcionamento Cliente

Com base na arquitetura apresentada na Figura 5.1, existem dois principais elementos dentro da arquitetura UbiPri, o servidor de contextos de privacidade e os dispositivos móveis clientes. O funcionamento no usuário cliente está implementado usando a plataforma *Android* em duas versões diferentes. A primeira versão infere a localização utilizando dados GPS em um repositório local de ambientes importados do servidor e envia a informação de forma síncrona para o servidor, que por sua vez retorna através desse método as ações que devem ser aplicadas no cliente móvel. A segunda versão foi desenvolvida de acordo com o dados obtidos por (NUNES, 2013) que utilizando as APIs (*Application Programming Interface*) de Serviço e *BroadcastReceiver* que recebem as ações de maneira assíncrona, através da API *Cloud Message* do Google. Desta forma, essa segunda aplicação envia ao servidor a localização de maneira assíncrona e recebe as ações também de maneira assíncrona.

Essas duas aplicações cliente demonstram que a arquitetura UbiPri suporta receber contextos de localização e executar ações tanto de maneira assíncrona como síncrona.

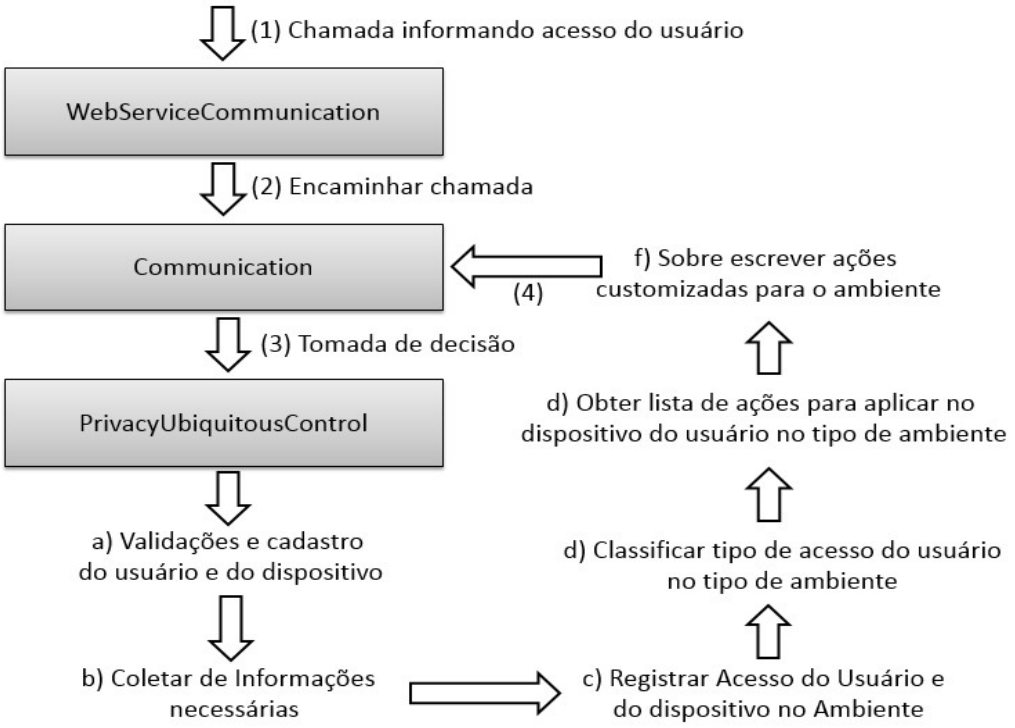
5.3 Arquitetura e Funcionamento Servidor

O servidor é responsável por receber as informações contextuais de localização e tomar a decisão de que ações devem ser feitas pelo dispositivo presente no ambiente baseado no tipo de usuário e o tipo de ambiente. Para tanto na implementação atual são utilizadas duas classes (*Communication* e *Privacy Control Ubiquitous*) e um processo de tomada de decisão do middleware UbiPri apresentados pela Figura 5.3, os quais são descritos abaixo:

1. A classe *WebServiceRestCommunication* recebe uma chamada remota usando a tecnologia *WebService Rest* através de um dos métodos: onChangeCurrentUserLocalization (se a chamada for assíncrona) ou onChangeCurrentUserLocalizationWithResponse. Em ambos os casos os parâmetros recebidos serão: *login*, senha do usuário, o dispositivo de origem da localização, o identificador do ambiente e como opcional o parâmetro indicando se o usuário está entrando ou saindo de um ambiente. As informações desse método são passadas para a classe *Communication*.
2. A classe *Communication* possui os métodos de igual nome utilizados para o recebimento de mensagens, além disso, é responsável por enviar as mensagens oriundas da decisão da classe *PrivacyControlUbiquitous*. Essa configuração se

deve ao fato de ser possível utilizar diversas tecnologias de comunicação simultaneamente para envio e recebimento de informações. Após receber os parâmetros das chamadas *onChange CurrentUser Localization WithResponse* e *onChange CurrentUser Localization* os parâmetros são repassados para a classe *PrivacyControlUbiquitous* através dos métodos *onChange CurrentUser Localization ReturnActions* e *onChange CurrentUser Localization WithReturn Asynchronous Actions* que fazem o mesmo processo de tomada de decisão de ações, diferenciando-se pela forma de retornar a ação.

Figura 5.3 - Processo de tomada de decisão



Fonte: Próprio autor.

O funcionamento das regras e conseguentemente a ativação e/ou desativação de funções do dispositivo do usuário são relacionados de acordo com a Tabela 5.4. Onde as ações de ativação / desativação são customizadas de acordo com as regras e definições do ambiente. A evolução automática do perfil do usuário será um ponto a ser desenvolvido conforme consta na seção 6.9, claramente descrita na sub-seção 6.9.2. No entanto, os demais processos funcionam da seguinte forma:

3. O processo de tomada de decisão, o qual já foi descrito anteriormente, foi implementado usando a seguinte sequência:

a) autenticação: autenticando o usuário, identificando se o usuário e dispositivo estão cadastrados;

b) busca de dados: busca as informações do ambiente, o dispositivo e as informações do usuário e seu perfil no ambiente (*Unknown, transient, user, student, responsible* e *administrator ou manager)*;

c) geração de controle: gera um log da mudança de posição e identifica se no momento atual é turno diurno, ou noturno, se é dia da semana ou final de semana e se é dia útil ou não.

4. tratamento de informações: com as informações do perfil do usuário no ambiente, o tipo de ambiente, o turno, se é dia da semana ou se é dia útil, procede-se a classificação com base em todas as 144 possibilidades, sendo algumas descritas na tabela 5.2 para definir o tipo de acesso que o usuário possui no ambiente; Com a informação do tipo de acesso do usuário no ambiente e o tipo de ambiente, é possível chegar à lista de ações que podem ser aplicadas no dispositivo visado, as quais estão descritas como funcionalidades. Após, as ações padrão podem ser sobrescritas por customizadas se houver alguma para aquele ambiente. Enviar as ações aos dispositivos utilizando a classe *Communication* se assíncrona ou retornando pelo método, se síncrona.

Nos experimentos realizados e resultados obtidos que podem ser visualizados nos Anexos A e B, as funções foram testadas e validadas parcialmente em 5 ambientes cadastrados, para tanto foram considerados 5 usuários, sendo as funcionalidades cadastradas diferenciadas a cada um deles, por exemplo horários, acesso a determinados ambientes, funções dos dispositivos diferenciados. Foram feitos testes em dois dispositivos reais com sistema operacional *Android* e também utilizando os emuladores disponibilizados pela plataforma *Android*. Os experimentos, resultados obtidos também foram publicados em Leithardt et al. (2013a), Leithardt et al. (2013b), Leithardt et al. (2013c) e Leithardt et al. (2014).

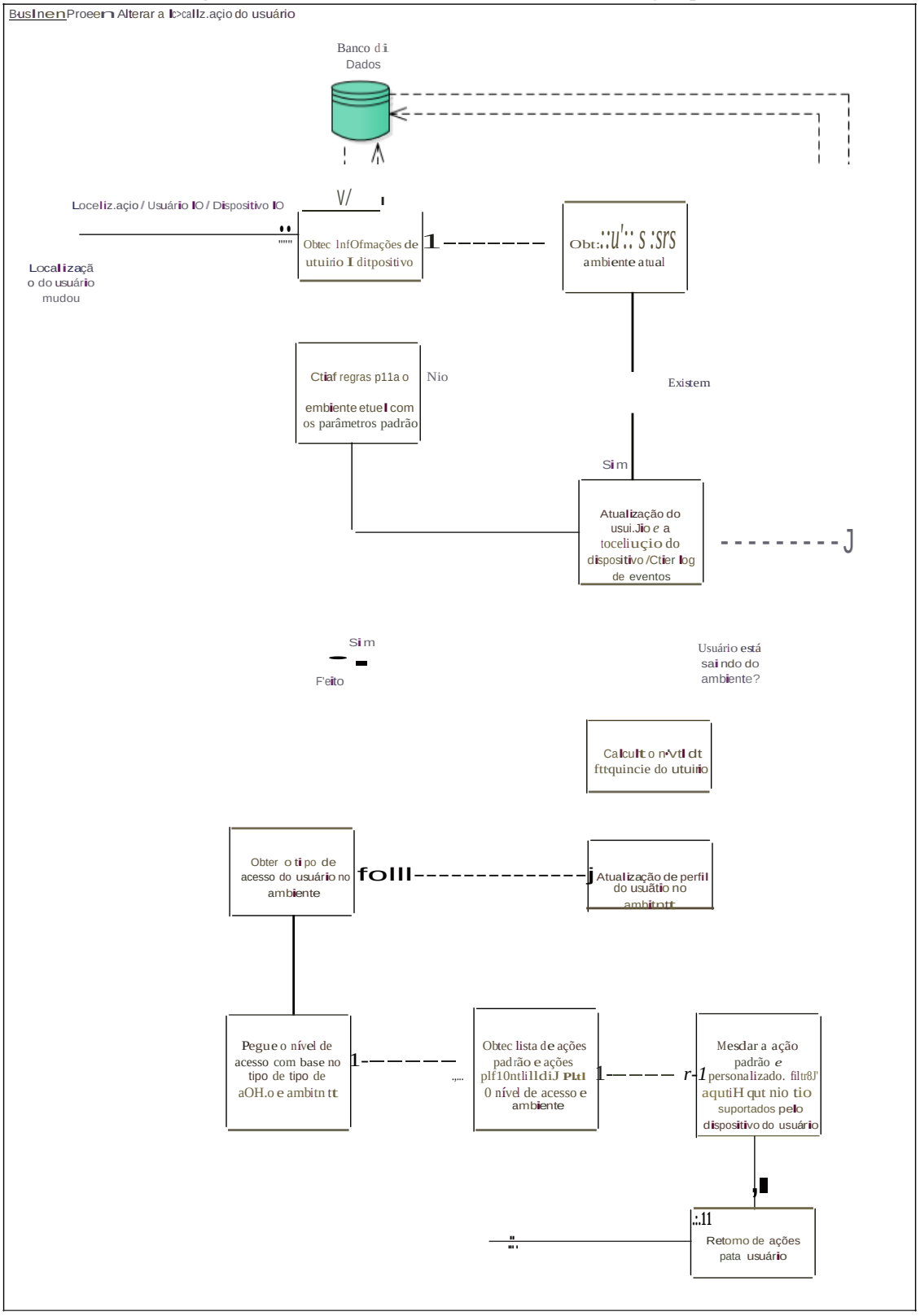
Com relação aos processos de atualização do *status* dos usuários é definido que o principal processo do sistema é a atualização da localização do usuário, devolvendo a este, se aplicável, as ações que devem ser aplicadas ao dispositivo com base em uma série de parâmetros. Este processo se inicia no dispositivo móvel do usuário, quando este detecta que o usuário entrou ou saiu de um ambiente. Efetuada a detecção, o dispositivo do usuário envia uma mensagem para o servidor com a identificação do ambiente, além da identificação do

usuário e dispositivo. O aplicativo desenvolvido está disponível gratuitamente e está disponível em UbiPri (2015).

Com base nestas informações o servidor obtém do banco de dados informações adicionais sobre o usuário e seu dispositivo, bem como as regras definidas para o usuário dentro do ambiente em que ele entrou/saiu. Caso estas regras não tenham sido definidas, o que ocorre quando o usuário entra em um ambiente pela primeira vez, o servidor se encarregará de criar as regras com parâmetros pré-definidos. Com todas estas informações armazenadas, o servidor atualiza a localização atual do usuário e de seu dispositivo, além de criar um log no banco de dados, descrevendo o evento que acabou de ocorrer. Caso o usuário esteja saindo do ambiente, não existe a necessidade de continuar o processo, pois os próximos passos são destinados à definição das ações que devem ser aplicadas ao dispositivo quando este entra em um ambiente.

Do contrário, o processo segue normalmente com o cálculo do nível de frequência atual do usuário para o ambiente o qual ele está entrando. Este cálculo é feito com base em regras definidas para o ambiente, que estipulam um período de tempo e um intervalo de frequência em porcentagem. O cálculo da porcentagem de frequência analisa os logs de check-ins do usuário para aquele ambiente dentro do período estipulado e verifica o nível de frequência ao qual o usuário pertence. Com o nível de frequência calculado, o servidor verifica se o perfil do usuário dentro do ambiente pode ou não ser evoluído. Para que isto ocorra o perfil atual do usuário precisa estar na lista dos perfis evolutivos, que são: desconhecido, transiente e usuário. Sendo um destes perfis, se o nível de frequência do usuário for alto (frequente) ele irá evoluir para o próximo perfil (se existir um perfil maior). Entretanto, se o nível de frequência for baixo o usuário terá seu perfil rebaixado. No caso de o nível de frequência ser normal, o perfil permanece inalterado, a Figura 5.4 apresenta graficamente os fluxos de processos realizados para tomada e evolução do perfil. Finalmente, com as informações do perfil do usuário no ambiente, o nível de frequência, o tipo de ambiente, e parâmetros temporais (dia de semana, dia útil, turno) o tipo de acesso do usuário e, subsequentemente, o nível de acesso do usuário.

Figura 5.4 -Processo de tomada de decisão evolução perfil



Fome: Próprio autor.

Através do nível de acesso do usuário as ações padrão para o ambiente são obtidas, e utilizando a identificação do ambiente as regras customizadas são obtidas. Ao final do processo as regras padrão são substituídas pelas regras customizadas (merge), caso haja relação direta entra elas (ações direcionadas para a mesma funcionalidade), e então devolvidas para o usuário para serem aplicadas no dispositivo. A metodologia utilizada nos experimentos, testes e resultados foi comparativa. Para tanto, foram comparados algoritmos conforme apresentados nas Tabelas 5.2 e 5.3, também foram comparados ambientes e cenários diferentes conforme apresentado na seção 4.5, onde foi possível comparar além de cenários diferentes, aplicações e funcionalidades diferentes. A metodologia de comparação também se fundamenta na Tablea 5.1 onde são apresentados exemplos de classificação do tipo de acesso, com isso desenvolveu se uma aplicação que pode ser visualizada em UbiPri (2015), validando Middleware para Controle e Gerenciamento de Privacidade em Ambientes Ubíquos.

5.4 Melhorias nos processos de evolução automática

Com vistas ao tratamento de dados para evolução automática do perfil do usuário é necessário a implementação de ajustes e melhorias nos processos, dentre os quais:

a) Com a elaboração do processo que realiza a evolução automática dos perfis dos usuários nos ambientes, pretende – se desenvolver mecanismo que gerencie todos os usuários e que inicialmente sejam considerados desconhecidos (*Unknown*) para todos os ambientes, isto é, sem registro no banco de dados, após passaria para o perfil de transitório (*Transient*) e por fim para usuário (*User*), dependendo da frequência ou outras variáveis que sejam consideradas vitais para classificação do usuário;
b) Terminar o classificador de tipo de acesso;
c) Desenvolver o mecanismo para inferência de parâmetros temporais do classificador;
d) Ampliar o número de ambientes que atualmente integram três grandes regiões definidas como pública, privada e restrita, desenvolver uma região pessoal;
e) Desenvolver um mecanismo de emissão de alerta com base na localização do usuário.

6 CONTRIBUIÇÕES CIENTÍFICO ACADÊMICA DA TESE

6.1 Grupos de Pesquisa

2014 - Coordenador de Grupo de Pesquisa (GPPD-i) – FIERGS / SENAI / RS

GRUPO DE PESQUISA: O Grupo de Processamento Paralelo e Distribuído Inteligente (GPPD-i) tem como foco o desenvolvimento de soluções nas áreas de computação paralela e distribuída e está ligado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. O grupo está localizado dentro da Faculdade de Tecnologia SENAI Porto Alegre (FATEC), na cidade de Porto Alegre, Rio grande do Sul, Brasil. A FATEC está vinculada e diretamente relacionada ao Serviço Nacional de Aprendizagem Industrial do Estado do Rio Grande do Sul (SENAI/RS) e a Federação das Indústrias do Estado do Rio Grande do Sul (FIERGS).

2011 – Integrante do GPPD INF UFRGS

O Grupo de Processamento Paralelo e Distribuído (GPPD) trabalha em soluções para as várias plataformas de processamento paralelo e distribuído, incluindo os *chips multicore*, arquiteturas com memória compartilhada, *clusters e grids* computacionais, e sistemas P2P e ubíquos. O GPPD trabalha em nível da arquitetura, desenvolve soluções para o gerenciamento de máquinas de processamento de alto desempenho, atua em programação paralela e em sistemas de armazenamento de dados, e em middleware para sistemas ubíquos (pervasivos) e jogos maciçamente multijogadores.

6.2 Projetos de Pesquisa

2015 - Coordenador - Projeto de Pesquisa: Solução inteligente para controle de ambientes acadêmicos integrada à gestão empresarial - (SAGAGE) “submetido Edital Senai de Inovação”.

As instituições acadêmicas ainda fazem uso de ferramentas manuais para gestão. Esses métodos manuais além de possuírem certo custo e um alto desperdício de tempo, oferecem. pouca segurança e imprecisão em suas informações, resultando em perdas para a instituição. O objetivo deste projeto é automatizar o fluxo de informações e o gerenciamento de usuários em cenários acadêmicos. Para isso, a tecnologia RFID será utilizada em conjunto com os atuais sistemas de gestão empresariais (ERP), resultando em uma nova ferramenta aplicável

aos mais variados sistemas comerciais, reduzindo custos e identificando gargalos no ambiente acadêmico. O sistema SAGAGE possui três principais diferenciais competitivos: (i) controle automático de identificação de usuários; (ii) controle de permissão e acesso a ambientes (salas de aulas, salas de atividades, administrativos, entre outros); (iii) e o envio de notificações personalizadas conforme o perfil e o posicionamento do usuário. Desta forma, as informações estarão centralizadas em um ERP para possibilitar em tempo real, uma análise do posicionamento dos usuários para o seu gerenciamento e controle (geração de relatórios demonstrando o tempo gasto de cada usuário em determinado espaço, por exemplo). Nosso desafio tecnológico é realizar a integração entre o SAGAGE, a rede de sensores Solentech e o ERP utilizado pela instituição. (SOLENTECH/SENAI/FIERGS).

2014 – Coordenador – Sistema de Controle Pervasivo para Ambientes de Ensino com a Utilização de Sensores sem Fios. (IEL/SENAI/CNPq)

Descrição: O Edital SENAI SESI de Inovação é um exemplo de uma ação anual de abrangência nacional em parceria com os Departamentos Regionais (DRs) do SENAI, do SESI e do SENAI/Cetiqt, envolvendo suas Unidades e profissionais; e com o MCTI Ministério de Ciência, Tecnologia e Inovação/CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico. O objetivo do projeto é a identificação de usuários, ambientes e dispositivos para processamento e tratamento das informações. Dentre as diversas formas de comunicação e identificação existente o foco é a utilização de Identificadores por Rádio Frequência (RFID).

Situação: Em andamento; Natureza: Pesquisa.

Alunos envolvidos: Graduação: (2)/ Especialização: (1)/ Mestrado acadêmico: (4)/ Mestrado profissionalizante: (1)/ Doutorado: (1).

Integrantes: Valderi Reis Quietinho Leithardt - Coordenador/ Guilherme Dal Bianco - Integrante/ Gustavo Bervian Brand - Integrante/ Elmário Gomes Dutra Júnior - Integrante. Cooperação: UFRGS, UFLA e Universidade de Coimbra PT.

Financiador(es): CNPq – IEL / SENAI / FIERGS.

2014 - Coordenador - Projeto de Pesquisa (SIGA-i)

O Projeto Sistema de Controle e Gerenciamento Acadêmico Inteligente (Siga-i) tem como objetivo o desenvolvimento de soluções que visam à otimização de recursos e a facilidade na troca de informações internas da instituição voltado a todos os níveis de ensino ofertados, bem como, o desenvolvimento de pesquisa científicas com base nos dados e

ferramentas de solução encontradas tendo como caso de uso a Faculdade de Tecnologia SENAI Porto Alegre (FATEC). O projeto Siga tem por objetivo identificar as principais demandas relacionadas ao funcionamento da instituição, e desenvolver soluções, com foco tecnológico, para a otimização de recursos e processos para facilitar a troca de informações. Com base nas soluções desenvolvidas, será possível introduzir novas tecnologias como (RFID, sensores, entre outros) com intuito de introduzir e desenvolver a pesquisa avançada na instituição. (SENAI/FIERGS).

2010 - UbiArch - Ubiquitous Architecture for Context Management and Application Development.

Descrição: A computação ubíqua (*ubiquitous computing*) e pervasiva (*pervasive computing*) ou simplesmente ubicomp é uma área promissora com aplicações em diversas áreas do conhecimento. Segundo Mark Weiser, considerado o pai da ubicomp devido a um artigo seu publicado em 1991 "as tecnologias mais profundas são aquelas que desaparecem". Elas se integram à vida cotidiana até se tornarem indistinguíveis da mesma. O artigo de Weiser prevê que os computadores pessoais desaparecerão do nosso olhar e passarão a fazer parte de todos os objetos, de forma integrada e onipresente (ou ubicomp). Ele compara este fenômeno ao desaparecimento dos motores, que encolheram até passarem a fazer parte de pequenos objetos do dia a dia sem, no entanto, tornarem-se visíveis aos olhos. Embora a proposta original de Weiser quanto à Computação Ubíqua ainda esteja distante de uma prática cotidiana alicerçada por produtos de mercado, sua proposta vem se materializando, pouco a pouco, através da disponibilização de tecnologias como PDAs, Smartphones, a miniaturização de processadores, circuitos integrados e sensores cada vez mais presentes em diferentes áreas da sociedade além da consolidação de padrões para redes sem fio como o Bluetooth, ZigBee e o IEEE 802.11. Nesse contexto, este projeto tem como foco de pesquisa a concepção de uma arquitetura de *software* e aplicações que contemplem o uso de informações de contexto provindas de diferentes fontes de dados, como por exemplo sensores. A plataforma escopo do mesmo se mostra uma etapa indispensável a ser consolidada no caminho de propostas como a de Weiser. Nesta perspectiva a Computação Ubíqua constituirá ainda um campo fértil para ofertas de produtos e desenvolvimento de pesquisas nos próximos anos.

Alunos envolvidos: Mestrado acadêmico (5) e Doutorado (4).

Integrantes: Valderi Reis Quietinho Leithardt - Integrante/ Cristiano André da Costa - Integrante/ Cláudio Fernando Resin Geyer - Coordenador/ Jorge Luis Victória Barbosa - Integrante/ Anubis Graciela de Moraes Rossetto - Integrante/ Carlos Oberdan Rolim -

Integrante/ João Ladislau Barbará Lopes - Integrante/ Rodrigo Santos de Souza - Integrante/ Adenauer Corrêa Yamin - Integrante/ Guilherme Antônio Borges - Integrante.

Financiador(es): Universidade Federal do Rio Grande do Sul - Outra.

6.3 Artigos completos publicados em periódicos

Leithardt, V. R. Q. ; Nunes, D. ; Rossetto, A. G. M. ; Rolim, C.O. ; Geyer, C.F.R. ; Silva, Jorge Sá . Privacy Management Solution in Ubiquitous Environments Using Percontrol. **Journal of Ubiquitous Systems and Pervasive Networks**, v. 5, p. 21-28, **2014**.

Leithardt, V. R. Q. ; Borges, Guilherme ; Rossetto, A. G. M. ; Rolim, C.O. ; Geyer, C.F.R. ; CORREIA, L. H. A. ; NUNES, D. ; SILVA, Jorge Sá . A Privacy Taxonomy for the Management of Ubiquitous Environments. **Journal of Communication and Computer**, v. 10, p. 129, **2013**.

6.4 Livros publicados/organizados ou edições

PADOIN, E. L. ; LEITHARDT, V. R. Q. ; PINTO, V. G. . WSPPD 2012 - X Workshop de Processamento Paralelo e Distribuído.. 10. ed. Porto Alegre - RS: Evangraf, 2012. v. 100. 78p .

6.5 Capítulos de livros publicados

ROLIM, CARLOS O. ; Rossetto, Anubis G. ; Leithardt, Valderi R. Q. ; Borges, Guilherme A. ; dos Santos, Tatiana F. M. ; Souza, Adriano M. ; Geyer, Cláudio F. R. . An Ubiquitous Service-Oriented Architecture for Urban Sensing. In: Springer-Verlag Berlin Heidelberg. (Org.). Communications in Computer and Information Science. ved.Berlin Heidelberg: Springer Berlin Heidelberg, 2015, v. 498, p. 1-10.

6.6 Trabalhos completos publicados em anais de congressos

Leithardt, Valderi R.Q. ; BORGES, G. A. ; CARRERA, I. M. ; Rossetto, A. G. M. ; Rolim, C.O. ; Nunes, D. ; SILVA, S. J. ; Geyer, C.F.R. Mobile Architecture for Identifying users in Ubiquitous Environments Focused on Percontrol. In: UBICOMM 2013, 2013, Porto. The

Seventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, **2013**. p. 145-151.

Leithardt, Valderi R.Q. ; CORREIA, LUIZ H. A. ; BORGES, G. A. ; ROSSETTO, Anubis Graciela de Moraes ; Rolim, Carlos Oberdan ; GEYER, C. F. R. ; RAPOSO, D. ; Nunes, D. ; SILVA, S. J. . A Solution for Identifying Users in Pervasive Environments Using Percontrol. In: XXXIX Conferencia Latino-americana en Informática (CLEI 2013), 2013, Naiguatá, Vargas. CLEI 2013. Vargas: IEEE, **2013**.

Leithardt, V. R Q; Rolim, C.; Rossetto, A; Geyer, C.; Dantas, M. A R; Silva, J.S.; Nunes, D., "Percontrol: A pervasive system for educational environments," Computing, Networking and Communications (ICNC), 2012 International Conference on, vol., no., pp.131, 136, Feb. 2 **2012** – IEEE - doi: 10.1109/ICCNC.2012.6167396

Leithardt, Valderi R.Q. ; ABECH, M. ; MORAES, R. ; Cambruzzi, W. L. ; Mendes, C. ; COSTA, C. A. ; GEYER, C. F. R. ; SILVA, S. J. . Uma Proposta para Gerenciamento de Privacidade em Ambientes Pervasivos Direcionado ao Controle Educacional. In: 17ª. Congresso de Informática Educativa (TISE), 2012, Santiago. Congresso Internacional de Informática Educativa TISE 2012. Santiago, **2012**. v. 8. p. 30-32.

Leithardt, V.R.Q.; Geyer, C.; Sá Silva, J.; Silva, R., "Use of Data Replication in WSNs Directed to Special Needs," Communications Workshops (ICC), 2010 IEEE International Conference on , vol., no., pp.1,5, 23-27 May **2010** - doi: 10.1109/ICCW.2010.5503914

Leithardt, V. R. Q.; Anjos, Julio, C.S.; Fontoura, E. S.; GEYER, C. F. R. Avaliação de Simuladores para RSSFs. In: Conferência IADIS Ibero Americana, 2010, Algarve/Portugal. IADIS International Journal on WWWInternet. CIAWI, **2010**.

Leithardt, V. R. Q. ; SILVA, R. ; SILVA, S. J. ; GRANJAL, J. ; GEYER, C. F. R. ; J. Rodrigues . Approaches to Node and Service Discovery in 6lowPAN. In: VII Workshop de Processamento Paralelo e Distribuído, 2009, Porto Alegre. VII Workshop de Processamento Paralelo e Distribuído (WSPPD). Porto Alegre: UFRGS, **2009**. v. 1. p 1-4.

Leithardt, V. R. Q.; GEYER, C. F. R.; Tavares. Estudo Comparativo entre Protocolos de Roteamento Utilizados em Redes de Sensores sem Fios. In: Congresso Trinacional da Fronteira, 2009, Foz do Iguaçu. IV Congresso da Academia Trinacional da Fronteira. Foz do Iguaçu: PTI Itaipu Binacional, **2009**. v. 4.

6.7 Orientações Graduação, Iniciação Científica.

Dionata Ferraz. Projeto SIGA-i. 2015. **Bolsista de Iniciação Científica com recursos FAPERGS.** (Grupo de Pesquisas em Processamento Paralelo e Distribuído Inteligente – GPPD-i) Faculdade de Tecnologia SENAI Porto Alegre.

Ezequiel Lafuente Coelho. Projeto SIGA-i. 2015. **Bolsista de Iniciação Científica com recursos FAPERGS.** (Grupo de Pesquisas em Processamento Paralelo e Distribuído Inteligente – GPPD-i) Faculdade de Tecnologia SENAI Porto Alegre.

Ângelo Victor Israel Muniz. Projeto SIGA-i. 2015. **Bolsista de Iniciação Científica com recursos CNPq.** (Grupo de Pesquisas em Processamento Paralelo e Distribuído Inteligente – GPPD-i) Faculdade de Tecnologia SENAI Porto Alegre.

Johnny William Gil da Luz. SISTEMA DE CONTROLE PARA GERENCIAMENTO DE AMBIENTES AUTOMATIZADOS. 2014. **Trabalho de Conclusão de Curso**. (Graduação em Automação Industrial) - Faculdade de Tecnologia SENAI Porto Alegre.

Luiza de Souza. Conception and Implementation of a Tiny Smart Environment Platform. 2013. **Trabalho de Conclusão de Curso**. (Graduação em Ciência da Computação) - Universidade Federal do Rio Grande do Sul.

BRUNO ROMEU NUNES. An automation system for ubiquitous computing. 2013. **Trabalho de Conclusão de Curso**. (Graduação em Ciência da Computação) - Universidade Federal do Rio Grande do Sul.

6.8 Orientações Pós-Graduação e Pesquisa.

Adrian Lemes Caetano. Projeto de pesquisa Sistema de Controle Pervasivo para Ambientes de Ensino com a Utilização de Sensores sem Fios. **2015**. **Bolsista IEL/FIERGS/**

CNPq. (Grupo de Pesquisas em Processamento Paralelo e Distribuído Inteligente – GPPD-i) Faculdade de Tecnologia SENAI Porto Alegre.

Wagner Kolberg. Projeto de pesquisa Sistema de Controle Pervasivo para Ambientes de Ensino com a Utilização de Sensores sem Fios. **2015**. **Bolsista IEL/FIERGS/ CNPq.** (Grupo de Pesquisas em Processamento Paralelo e Distribuído Inteligente – GPPD-i) Faculdade de Tecnologia SENAI Porto Alegre.

Michel Stevan Simor. A conscientização de usuários sobre segurança da informação em ambientes corporativos; **2015**. Monografia (Aperfeiçoamento/**Especialização em Segurança da Informação e Gestão de Riscos**) - CESME Complexo de Ensino Superior Meridional, Passo Fundo - RS.

Alexandre Conteratto. Serviços de Acesso Seguro – EDURAM; **2014**. Monografia (Aperfeiçoamento/**Especialização em Inovação Tecnológica em Sistemas Distribuídos e Redes**) - CESME Complexo de Ensino Superior Meridional, Passo Fundo - RS.

Eduardo Arend. Uma Proposta para utilização de QOS em redes IP; **2014**. Monografia (Aperfeiçoamento/**Especialização em Inovação Tecnológica em Sistemas Distribuídos e Redes**) - CESME Complexo de Ensino Superior Meridional, Passo Fundo - RS.

Gian Carlo Bahú. Computação e Perícia Forense - Ferramentas Aplicadas na Recuperação de Dados; **2014**. Monografia (Aperfeiçoamento/**Especialização em Inovação Tecnológica em Sistemas Distribuídos e Redes**) - CESME Complexo de Ensino Superior Meridional, Passo Fundo - RS.

Jonatas Menegazzo. Um Estudo de Caso aplicado a Tecnologia RFID; **2014**. Monografia (Aperfeiçoamento/**Especialização em Inovação Tecnológica em Sistemas Distribuídos e Redes**) - CESME Complexo de Ensino Superior Meridional, Passo Fundo - RS.

Matheus Frizzo Weirich. Monitoramento de Rede Usando Ferramenta Zabbix; **2013**. Monografia (Aperfeiçoamento/**Especialização em Inovação Tecnológica em Sistemas Distribuídos e Redes**) - CESME Complexo de Ensino Superior Meridional, PF - RS.

6.9 Planejamento de pesquisas futuras

O planejamento das atividades a serem realizadas está organizado em seções denominadas de submissão de registro e patente, metas a serem alcançadas e o cronograma com datas e definições das atividades após a defesa da TESE, conforme segue.

6.9.1 Submissão de registro e patente

UbiPri – Sistema de Controle e Gerenciamento de Privacidade em Ambientes Computacionais Inteligentes.

6.9.2 Conclusões da tese e trabalhos futuros

Os resultados da tese se mostraram promissores e satisfatórios sob o ponto de vista teórico prático e demonstraram forte contribuição cientifica que pode ser comprovada com as várias publicações em seminários, simpósios, congressos e revistas da área. As publicações foram realizadas em importantes meios de divulgação nacional e internacional com qualis capes reconhecido em toda a comunidade acadêmica. Também foi possível realizar um estudo e embasamento teórico para comprovar a viabilidade e necessidade de controlar e gerenciar a privacidade de dados em ambientes ubíquos. A contribuição principal foi o middleware proposto com foco no ambiente ubíquo diferentemente dos mecanismos que até então eram propostos com foco em usuários, dispositivos e comunicações, com isso, comprova se também ser uma solução inovadora e que mereceu conceito máximo na defesa e possibibilidade de continuidade das pesquisas em pós doutorado.

Outro destaque inovador foi a elaboração de uma taxonomia de privacidade fundamentada na literatura pesquisada utilizando referencial teórico acadêmico de renomados pesquisadores da área, sendo possível apresentar uma definição taxonômica necessária para o controle e gerenciamento de privacidade focado no ambiente ubíquo. Outro destaque da tese foram os diferentes cenários e aplicações no qual o modelo de controle e gerenciamento de privacidade ubíqua denominado UbiPri pode ser testado, comparado e validado, sendo possível o desenvolvimento de projetos em parcerias com empresas e agência de fomento como o CNPq. Para finalizar, está disponibilizado no google play store o aplicativo que pode ser adquirido gratuitamente e que atende alguns requisitos e parâmetros apresentados nesta tese. Esperamos que as pesquisas realizadas, os resultados obtidos no desenvolvimento desta

tese também possam contribuir para os demais pesquisadores da área. Sugere – se que em trabalhos futuros sejam abordados os seguintes tópicos:

a) Implementação de experimentos, testes, validações e resultados obtidos no modelo protótipo desenvolvido em outros cenários e sistemas operacionais;
b) desenvolvimento do mecanismo de controle e gerenciamento de privacidade para envio de alertas e mensagens de acordo com a localização e regras;
c) desenvolvimento de mecanismos do módulo de controle de histórico PRIHIS, para gerenciar e controlar usuários em ambientes de acordo com o histórico de uso;
d) escrita e submissão de artigo na conferência security and privacy (S&P) IEEE – A1;
e) implementação de requisitos de segurança e criptografia do modelo de privacidade;
f) escrita e submissão de registro de patente do modelo de controle e gerenciamento de privacidade ubíqua;
g) escrita e submissão de projetos de pesquisa junto ao CNPq e demais órgãos de fomento, empresas e indústrias;
h) orientação de trabalhos acadêmicos de graduação e pós-graduação na área de privacidade em ambientes ubíquos;
i) escrita e proposta para Pós - Doutorado.

Além dos vários pontos abordados, sugere-se que outros não menos importantes, também sejam tratados em trabalhos futuros, tais como: tratamento de dados em larga escala, tratamento de dados distribuídos, requisitos relacionados à segurança de dados, tolerância a falhas e algoritmos visando melhoria de desempenho, sugere se neste último item que sejam aprofundados algoritmos relacionados a inteligência artificial.

REFERÊNCIAS

ARAUJO, Regina Borges de. Computaçao ubiqua: principios, tecnologias e desesafios. In: SIMPOSIO BRASILEIRO DE REDES DE COMPUTADORES, 21, 2003, Natal. **Anais...**Natal : UFRN/DIMAP, 2003. p. 45-115

BARDRAM, Jakob E.; KJAER, Rasmus E.; PEDERSEN, Michael. Context-aware user authentication–supporting proximity-based login in pervasive computing. In: UBIQUITOUS COMPUTING, 2003, Zurich – Suiça. **Proceedings…**:[S.1.]: Springer Berlin Heidelberg, 2003. p. 107-123. (Ubicomp 2003).

BERESFORD, Alastair R.; STAJANO, Frank. Location privacy in pervasive computing. **Pervasive Computing, IEEE**, v. 2, n. 1, p. 46-55, 2003.

CAMBRUZZI, W. et al. Um Modelo para Gerenciamento de Múltiplas Trilhas Aplicado a Sistemas de Apoio à Educação. In: XXIII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 2012, Rio de Janeiro. **Anais...**[S.l. : s.n.], 2012. p. 1-10.

CAMPBELL, Roy et al. Towards security and privacy for pervasive computing. In: **Software Security-Theories and Systems**. Springer Berlin Heidelberg, 2003. p. 1-15.

CHEN, H.; FININ, T.; JOSHI, A.; Using OWL in a pervasive computing broker. In: 3TH WORKSHOP ON ONTOLOGIES IN AGENT SYSTEMS, 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2005, Melbourne. **Proceedings...**[S.l. : s.n.], 2005. p. 9-16.

COSTA, Cristiano André da; YAMIN, Adenauer C.; GEYER, Claudio Fernando Resin. Toward a general software infrastructure for ubiquitous computing. **IEEE Pervasive Computing**, v. 7, n. 1, p. 64-73, 2008.

DEY, A.K., Providing architectural support for building context-aware applications, Ph.D. Thesis, College of Computing, **Georgia Institute of Technology**, Atlanta, December 2000.

DIGITAL AGENDA FOR EUROPE. Disponível em <http://ec.europa.eu/information_society/policy/ecomm/todays_framework/privacy_protectio n/index_en.htm>. Acesso em: 10 de Mar. de 2014.

EL DEFRAWY, Karim; TSUDIK, Gene. Privacy-preserving location-based on-demand routing in MANETs. **Selected Areas in Communications, IEEE Journal on**, v. 29, n. 10, p. 1926-1934, 2011.

ESQUIVEL, A.; HAYA, P.; ALAMÁN, X. Fair Trade Metaphor as a Control Privacy Method for Pervasive Environments: Concepts and Evaluation. In: INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING AND AMBIENT INTELLIGENCE & THE 6TH INTERNATIONAL WORKSHOP ON AMBIENT ASSISTED LIVING, 2014, Switzerland. **Proceedings...**:[S.1.]: Springer, 2015. pp 337-344. (UCAMI & IWAAL 2014).

FACEBOOK 2015, Rede social Facebook, disponível em https://www.facebook.com/FacebookBrasil 2015. Acessado em Novembro 2015.

FOURSQUARE. Disponível em: <https://pt.foursquare.com/>. Acessado em Julho de 2015.

GETGLUE. Disponível em: <http://getglue.com/>. Acessado em Maio de 2015.

GÖRLACH, Andreas; HEINEMANN, Andreas; TERPSTRA, Wesley W. Survey on location privacy in pervasive computing. In: PRIVACY, SECURITY AND TRUST WITHIN THE CONTEXT OF PERVASIVE COMPUTING. **Proceedings...**:[S.l.]: Springer, 2005. p. 23-34.

HENRICKSEN, Karen. et al. Extending context models for privacy in pervasive computing environments. In: PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS. Third IEEE International Conference on. 2005, Hawaii - USA. **Proceedings...**[S.l. : s.n.], 2005. p. 20-24.

HUBNER, Simone Fischer; MATHEW, Wright. Privacy Enhancing Technologies. In: INTERNATIONAL SYMPOSIUM, 12th, 2012, Vigo-Spain. **Proceedings...**:[S.l.]: Springer-Verlag Berlin Heidelberg**,** 2012, p.221-238. (PETS 2012).

IOANNIS Krontiris, Tassos Dimitriou, A platform for privacy protection of data requesters and data providers in mobile sensing, **Computer Communications**, Volume 65, 1 July 2015, Pages 43-54, ISSN 0140-3664, http://dx.doi.org/10.1016/j.comcom.2015.02.005.(http ://www.sci encedirect.com/science/article/pii/S0140366415000560).

LACHELLO, Giovanni; HONG, Jason. End-user privacy in human-computer interaction. **Foundations and Trends in Human-Computer Interaction**, v. 1, n. 1, p. 1-137, 2007.

LANGHEINRICH, Marc. Privacy by Design – Principles of Privacy-Aware Ubiquitous Systems. In: INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING, 2001, London, UK. **Proceedings**…ACM, 2001. p. 273-291. (UBICOMP '01).

LANGHEINRICH, Marc.; BORRIELLO, Gaetano.; HOLMQUIST, Lars Erik.; 2002 – "A Privacy Awareness System for Ubiquitous Computing Environments" In: UBICOMP 2002: UBIQUITOUS COMPUTING: 4th International Conference Göteborg, 2002, Sweden, **Proceedings**…[S.l. : s.n.], 2002. pp. 237-245.

KAGAL, Lalana; FININ, Tim; JOSHI, Anupam. Trust-based security in pervasive computing environments. **Computer, IEEE,** v. 34, n. 12, p. 154-157, 2001.

KALEMPA, Vivian Cremer. **Especificando privacidade em ambientes de computação ubíqua.** 2009. 140 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Catarina (UFSC), Florianópolis. 2009.

KRUMM, John. A survey of computational location privacy. **Personal and Ubiquitous Computing.** V. 13, n. 6, p. 391-399, 2008.

KUBICKI, Sébastien; LEPREUX, Sophie; KOLSKI, Christophe. RFID-driven situation awareness on TangiSense, a table interacting with tangible objects. **Personal and Ubiquitous Computing**. V. 16, n. 8, p. 1079-1094, dez, 2012.

LEHIKOINEN, Jaakko T.; LEHIKOINEN, Juha; HUUSKONEN, Pertti. Understanding privacy regulation in ubicomp interactions. **Personal and Ubiquitous Computing,** v. 12, n. 8, p. 543-553, 2008.

LEITHARDT, Valderi Reis Quietinho. **Modelo gerenciador de serviços para plataformas pervasivas sensíveis ao contexto**. 2008. 86 f. Dissertação (Mestrado em Ciência da Computação) – Faculdade de informática, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, 2008.

LEITHARDT, V. R. Q.; GEYER, C. F. R.; TAVARES, J. Estudo comparativo entre protocolos de roteamento utilizados em redes de sensores sem fios. In: CONGRESSO TRINACIONAL DA FRONTEIRA, IV Congresso da Academia Trinacional da Fronteira, 2009. Foz do Iguaçu-PR: **Anais...** Foz do Iguaçu – PR: PTI Itaipu Binacional, 2009. P. 1-10.

LEITHARDT, V.R. Q.; GEYER, C. F. R.; SILVA, J. S. Uma proposta de Middleware híbrido para RSSFs. In: ESCOLA REGIONAL DE ALTO DESEMPENHO (ERAD). 11ª Erad 2011, Porto Alegre-RS. **Anais…**Porto Alegre: SBC – Instituto de Informática da UFRGS, 2011. p. 118-119.

LEITHARDT, V. R. Q. et al. Percontrol: A pervasive system for educational environments. In: COMPUTING, NETWORKING AND COMMUNICATIONS, 2012, Hawai – USA. **Proceedings**…IEEE Computer Society, 2012. p.131-136. (ICNC 2012).

LEITHARDT, V. R. Q. et al. Mobile Architecture for Identifying Users in Ubiquitous Environments Focused on Percontrol. In: INTERNATIONAL CONFERENCE ON MOBILE UBIQUITOUS COMPUTING, SYSTEMS, SERVICES AND TECHNOLOGIES, 2013 Porto-Portugal. **Proceedings**…:[S.1.]: IARIA, 2013a. p. 145-151.

LEITHARDT, Valderi R. Q. et al. A Privacy Taxonomy for the Management of Ubiquitous Environments. **Journal of Communication and Computer**, v. 10, p. 1529-1553, 2013b.

LEITHARDT, V. R. Q. et al. A Solution for Identifying Users in Pervasive Environments Using Percontrol. In: CONFERENCIA LATINOAMERICANA EN INFORMÁTICA (CLEI 2013), XXXIV, 2013, Naiguatá - Venezuela. **Proceedings**…IEEE Computer Society, 2013c. p.1-10. (CLEI 2013).

LEITHARDT, Valderi R. Q. et al. Privacy Management Solution in Ubiquitous Environments Using Percontrol. **Journal of Ubiquitous Systems & Pervasive Networks**, v. 5, n. 2, p. 21-28, 2014.

LI, Na. et al. Privacy preservation in wireless sensor networks: A state-of-the-art survey. **Ad Hoc Networks**, v. 7, n. 8, p. 1501-1514, 2009.

LUPIANA, D.; O'DRISCOLL, C.; MTENZI, F. Taxonomy for ubiquitous computing environments. In: NETWORKED DIGITAL TECHNOLOGIES, First International Conference on. 2009. **Proceedings...** IEEE Computer Society, 2009. p. 469-475. (NDT'09).
MATTES, Leonardo et al. Linguagem lógica formal para expressar segurança em ambientes pervasivos. In: IV WORKSHOP EM SEGURANÇA DE SISTEMAS COMPUTACIONAIS. 2004, Gramado - RS. **Anais...**:[S.1.] : SBC, 2004. P. 1-10.

MOSCHETTA, Eduardo et al. Flexibilizando graus de colaboração, segurança e privacidade na descoberta de serviços em ambientes ubíquos. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, XXVI (SBRC 2008). Porto Alegre: **Anais…**:[S.1.]: SBC, 2008. p. 707-720.

MONTSERRAT Ros, Matthew D'Souza, Adam Postula, and Ian Maccoll. 2013. Wireless outdoor personal area network using adaptive inquiry scanning for location-based services. **Personal Ubiquitous Comput.** (February 2013), 387-398. DOI=10.1007/s00779-011-0501-2.

NUNES, Bruno Romeu et al. **An automation system for ubiquitous computing**. 2013. 63f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

NUNES, David et al. A web service-based framework model for people-centric sensing applications applied to social networking. **Sensors**, v. 12, n. 2, p. 1688-1701, 2012.

PEREIRA, Vasco et al. A taxonomy of wireless sensor networks with QoS. In: NEW TECHNOLOGIES, MOBILITY AND SECURITY, 4th IFIP International Conference on, 2011, Paris - France. **Proceedins...** IEEE Computer Society, 2011. p. 1-4. (NTMS 2011),

PLODERER, Bernd; HOWARD, Steve; THOMAS, Peter. Collaboration on social network sites: amateurs, professionals and celebrities. **Computer Supported Cooperative Work (CSCW)**, v. 19, n. 5, p. 419-455, 2010.

RFID JOURNAL. Disponível em: <http://www.rfidjournal.com/>. Acesso em Abril de 2014.

RODRIGUES, Vagner José do Sacramento. **Gerência de privacidade para aplicações sensíveis ao contexto em redes móveis**. 2006. 136 f. Tese (Doutorado em Informática) – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro - RJ, 2006.

ROLIM, C.; O. et al. Análise de uma rede neural híbrida como base para um mecanismo de predição de situação. In: CONGRESSO DA SOCIEDADE BRASILEIRA DA COMPUTAÇÃO (CSBC 2012), IV Simpósio de Computação Pervasiva Ubíqua, 2012. Curitiba - PR, **Anais…**:[S.1.] : SBC, 2012a. p.1-10.

ROLIM, C.; O. et al. Comparison of a Multi output Adaptative Neuro-Fuzzy Inference System (MANFIS) and Multi Layer Perceptron (MLP) in Cloud Computing Provisioning. In: X WORKSHOP EM CLOUDS E APLICAÇÕES – WCGA, 29th Brazilian Symposium on Computer Networks and Distributed Systems, 2012, Ouro Preto – MG. **Proceedings...**: [S.1.]: SBC, 2012b. p.1-10.

ROLIM, C.; O. et al. Towards a Novel Engine to Underlie the Data Transmission of Social Urban Sensing Applications. In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 17th ICEIS, 2015, Barcelona - Espanha. **Proceedings…** SCITEPRESS, 2015. P.662-667.

ROSSETTO, Anubis G. M. et al. An adaptive fault tolerance approach to enhance the execution of applications on multi–cluster grid configurations from mobile grid interfaces in wireless networks. **International Journal of High Performance Systems Architecture,** v. 3, n. 4, p. 202-215, 2011.

SAHA, Debashis; MUKHERJEE, Amitava. Pervasive computing: a paradigm for the 21st century. **Computer, IEEE**, v. 36, n. 3, p. 25-31, 2003.

SAPO SERVICES. Disponível em: <https://store.services.sapo.pt/en/>. Acesso em Maio de 2014.

SATYANARAYANAN, Mahadev. Pervasive computing: Vision and challenges. **Personal Communications**, IEEE, v. 8, n. 4, p. 10-17, 2001.

SHANKAR, Kalpana et al. Aging, privacy, and home-based computing: Developing a design framework. **Pervasive Computing, IEEE**. 2012, p. 46-54.

SILVA, Ricardo et al. A comparison of approaches to node and service discovery in 6lowPAN wireless sensor networks. In: SYMPOSIUM ON QOS AND SECURITY FOR WIRELESS AND MOBILE NETWORKS. 5th, 2009, Tenerife_Ilhas Canarias – Espanha. **Proceedings…**ACM, 2009. p. 44-49. (Q2SWinet 2009).

SOUZA, Luiza de. et al. **Conception and implementation of a Tiny Smart Environment Platform.**. 2013. 99f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

UBIPRI**,** Disponível em https://play.google.com/store/apps/details?id=com.gppdi.ubipri&hl=pt_BR – Acessado em Agosto 2015.

Warren and Bradeis: “the Right to Privacy” in (**HARVARD LAW REVIEW vol. 04, fls. 193**); December 1890. Disponível em: http://www.english.illinois.edu/-people-/faculty/debaro n/582/582%20readings/right%20to%20privacy.pdf.html. Acessado em Julho 2014.

WAZE. Disponível em: <http://www.waze.com/guided_tour/>. Acesso em Maio de 2015.

WEISER, Mark. **The computer for the twenty-first century**. 1991. Disponível em: <http://web.media.mit.edu/~anjchang/ti01/weiser-sciam91-ubicomp.pdf>. Acesso em: 20 de Maio de 2015.

WEKA, Data Mining Software in Java. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>. Acesso em novembro 2015.

YITAO, DUAN.; JOHN, CANNY. Zero-knowledge Test of Vector Equivalence and Granulation of User Data with Privacy. In: INTERNATIONAL CONFERENCE ON GRANULAR COMPUTING, 2006, Atlanta, USA. **Proceedings…**IEEE Computer Society, 2006. P. 1-6. (GrC 2006).

ZHU, Fen; MUTKA, Matt W.; NI, Lionel M. Service discovery in pervasive computing environments. **IEEE Pervasive computing,** v. 4, n. 4, p. 81-90, 2005.

ANEXO A: SERVIDOR UBIPRI

A.1. Requisitos

Nesta seção estão apresentados e descritos os requisitos do sistema.

Tabela A - Requisitos do Servidor

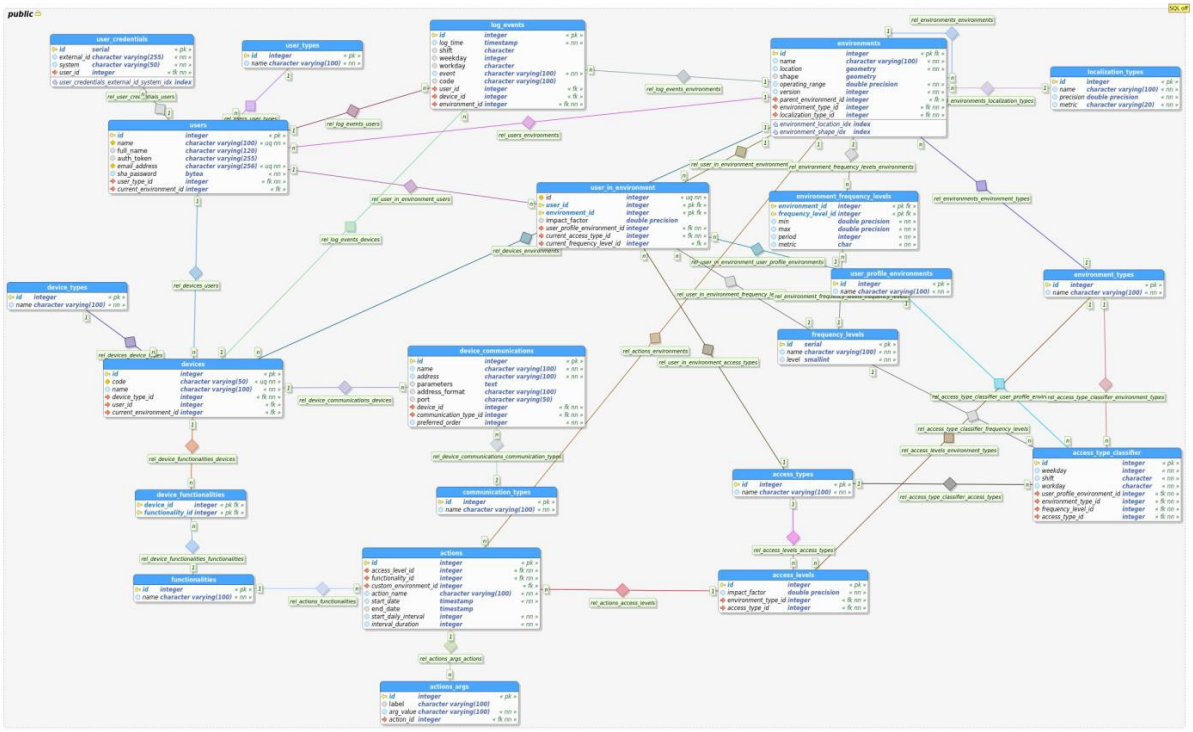
Ref.	Requisito
RF001	Aplicação de ações no dispositivo do usuário com base em uma série de parâmetros de context
RF002	Controle da frequência dos usuários nos ambientes, classificando-os em pouco frequentes, normal, ou frequentes, com base em regras pré-definidas por ambiente.
RF003	Evolução automática do perfil do usuário no ambiente com base na frequência do mesmo no ambiente.
RF004	Cadastro de funcionalidades de dispositivos, ex.: volume de toque, conexão WiFi ou 3G, etc.
RF005	Cadastro de diferentes níveis de acesso, cada um com suas ações específicas.
RF006	Cadastro de ações. Cada ação é designada a um nível de acesso e uma funcionalidade de dispositivo. Ex.: ação de ligar a WiFi.
RF007	Cadastro de ações específicas a um ambiente ou período de tempo. São ações que só serão aplicadas a determinado ambiente e/ou em determinado período de tempo, sendo este um período finito ou recorrente. Isso permite a criação de exceções para um ambiente em determinados períodos, como um evento público em um ambiente normalmente classificado como privado.
RF008	Classificação de ambientes em tipos, com base no nível de privacidade exigido: restrito, privado ou público.
RF009	Identificação do perfil do usuário dentro de um ambiente específico. Perfis padrão: desconhecido, transiente, usuário, responsável, estudante ou gerente.
RF010	Identificação do tipo de acesso de um usuário em determinado ambiente com base no tipo de ambiente, perfil do usuário neste ambiente, nível de frequência no ambiente, além de informações temporais (dia de semana, dia útil, turno do dia).
RF011	Cadastro de ambientes com informações sobre a localização geográfica do mesmo (latitude, longitude e raio de alcance), tipo de técnica preferencial para detecção do ambiente (GPS, RFID, NFC, etc.) e opcionalmente o desenho do ambiente (polígono).
RF012	Registro de log de entrada/saída de usuário nos ambientes, com dados temporais, e o dispositivo utilizado.
RF013	Cadastro de usuários, bem como de seus dispositivos e as funcionalidades suportadas pelos mesmos.

Fonte: Do autor.

A.2. Modelo Entidade Relacionamento

O modelo físico do banco de dados na Figura A descreve as entidades e seus relacionamentos, ambos necessários para a realização do sistema conforme os requisitos elencados na seção anterior.

Figura A - Modelo Entidade Relacionamento



Fonte: Do autor.

Nos próximos parágrafos as entidades e seus respectivos atributos serão descrevidos em detalhes.

A tabela ***environments*** armazena as informações sobre ambientes, em especial a localização, em formato *geometry* proveniente do PostGIS, sendo na prática um ponto geográfico com latitude e longitude. Opcionalmente também é possível armazenar a forma do ambiente, i.e. o desenho do ambiente utilizando coordenadas geográficas, formando um polígono.

Ambientes também possuem um tipo de localização preferencial para detecção, este armazenado na tabela ***localization_types***, que define o tipo de localização, a precisão máxima que a técnica de localização consegue entregar e a métrica por ela utilizada.

Ambientes também possuem controle de versão, este consiste em um valor inteiro que é incrementado toda vez que uma alteração for feita nos dados do ambiente.

A classificação dos ambientes é feita pela tabela ***environment_types***. O tipo de ambiente é um dos parâmetros utilizados para a determinação do tipo de acesso do usuário. Os tipos de ambiente padrão são: público, privado e restrito.

Outro parâmetro utilizado para determinar o tipo de acesso do usuário é o nível de frequência, definido pela tabela ***frequency_levels***. Nela são armazenados o nome do nível de frequência e um valor inteiro correspondente ao nível (iniciando em 1, e quanto maior o valor maior a frequência representada).

Para que a determinação do nível de frequência do usuário em um ambiente seja possível, é necessária a criação de regras específicas para o ambiente que consigam definir se um usuário está sendo frequente ou não. Estas regras ficam armazenadas na tabela ***environment_frequency_levels***. Ela define para cada par de ambientes e nível de frequência o período que será avaliado para determinar a frequência, a métrica do período (dias, semanas, meses, anos), e a frequência mínima e máxima (em porcentagem) para a escolha do nível de frequência.

Outra tabela necessária para determinar o nível de frequência do usuário é a ***log_events***. Toda vez que um usuário entra ou saí de um ambiente, um registro é criado nesta tabela, indicando o usuário, o dispositivo que ele estava utilizando, a data e hora (e turno, dia de semana, dia útil) em que o evento ocorreu e o ambiente onde o evento ocorreu.

O nível de frequência, além de servir para determinar o tipo de acesso do usuário, também é utilizado para evoluir o perfil do usuário no ambiente. Os perfis de usuário do ambiente são definidos na tabela ***user_profile_environments***. E para cada combinação de usuário e ambiente, o perfil de usuário é definido na tabela ***user_in_environment***.

Através do perfil do usuário no ambiente, juntamente com o tipo de ambiente, o nível de frequência, o turno do dia, se é ou não um dia da semana e dia útil, o tipo de acesso do usuário é determinado. As diferentes combinações de parâmetros ficam armazenadas na tabela ***access_type_classifier***.

De acordo com o tipo de acesso e tipo de ambiente é possível então determinar o nível de acesso do usuário, definido pela tabela ***access_levels***. Cada nível de acesso possui um conjunto de ações (tabela ***actions***) que devem ser aplicados ao dispositivo do usuário. Cada ação faz referência a uma determinada funcionalidade dentre a lista de possíveis funcionalidades na tabela ***functionalities*** e a ação a ser

tomada com relação à funcionalidade, como ligar a WiFi ou reduzir o volume do toque do telefone.

Ações podem também ser específicas para um único ambiente, ocorrer em um período específico ou ocorrer periodicamente durante intervalos de tempo. Outra informação opcional para as ações são argumentos adicionais que podem ser enviados para o dispositivo do usuário.

A lista de funcionalidades abrange diversos tipos de dispositivos, mas é contra produtivo enviar todas as ações para determinado dispositivo. Para que um filtro possa ser aplicado, cada dispositivo (definido em ***devices***) possui uma relação com as funcionalidades, armazenado em ***device_functionalities***, possibilitando assim que apenas as ações relevantes ao dispositivos sejam enviadas ao mesmo.

Os dispositivos são identificados por um código único, este gerado pelo próprio dispositivo. Eles também são classificados em tipos (tabela ***device_types***) e pertencem a um único usuário.

Usuários (tabela ***users***) também são classificados em tipos, e possuem credenciais (tabela ***user_credentials***) que identificam o sistema de origem do usuário.

A.3. Processo: atualizar a Localização do Usuário

O principal processo do sistema é a atualização da localização do usuário, devolvendo a este, se aplicável, as ações que devem ser aplicadas ao dispositivo com base em uma série de parâmetros e, este processo está representado na Figura C.

Este processo se inicia no dispositivo móvel do usuário, quando este detecta que o usuário entrou ou saiu de um ambiente. Efetuada a detecção, o dispositivo do usuário envia uma mensagem para o servidor com a identificação do ambiente, além da identificação do usuário e dispositivo.

Com base nestas informações o servidor obtém do banco de dados informações adicionais sobre o usuário e seu dispositivo, bem como as regras definidas para o usuário dentro do ambiente em que ele entrou/saiu. Caso estas regras não tenham sido definidas, o que ocorre quando o usuário entra em um ambiente pela primeira vez, o servidor se encarregará de criar as regras com parâmetros pré-definidos.

Com todas estas informações em mãos, o servidor atualiza a localização atual do usuário e de seu dispositivo, além de criar um log no banco de dados, descrevendo o evento que acabou de ocorrer.

Caso o usuário esteja saindo do ambiente, não existe a necessidade de continuar o processo, pois os próximos passos são destinados à definição das ações que devem ser aplicadas ao dispositivo quando este entra em um ambiente.

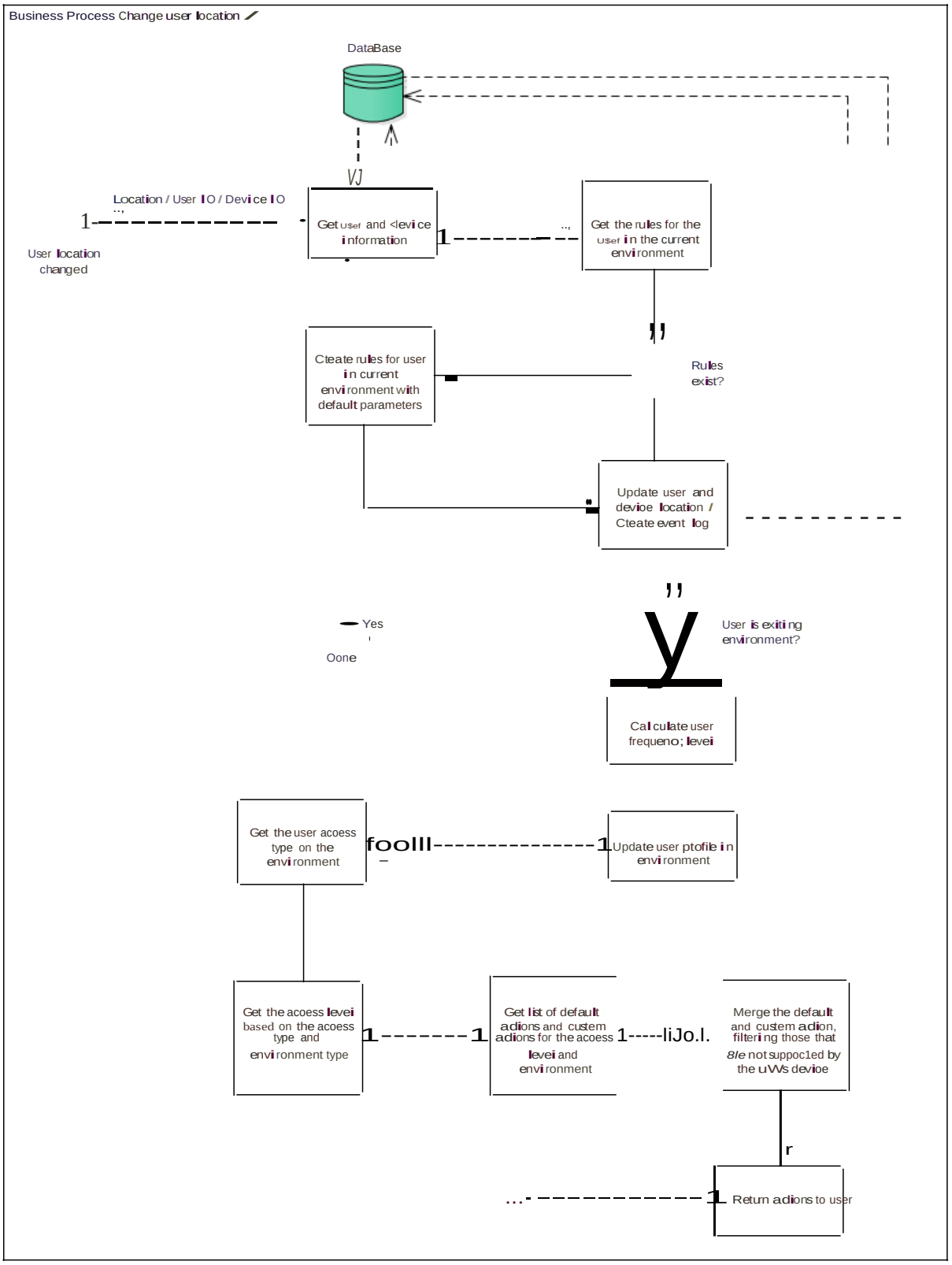
Do contrário, o processo segue normalmente com o cálculo do nível de frequência atual do usuário para o ambiente o qual ele está entrando. Este cálculo é feito com base em regras definidas para o ambiente, que estipulam um período de tempo e um intervalo de frequência em porcentagem. O cálculo da porcentagem de frequência analisa os logs de check-ins do usuário para aquele ambiente dentro do período estipulado e verifica o nível de frequência ao qual o usuário pertence.

Com o nível de frequência calculado, o servidor verifica se o perfil do usuário dentro do ambiente pode ou não ser evoluído. Para que isto ocorra o perfil atual do usuário precisa estar na lista dos perfis evolutivos, que são: desconhecido, transiente e usuário. Sendo um destes perfis, se o nível de frequência do usuário for alto (frequente) ele irá evoluir para o próximo perfil (se existir um perfil maior). Entretanto, se o nível de frequência for baixo o usuário terá seu perfil rebaixado. No caso de o nível de frequência ser normal, o perfil permanece inalterado.

Finalmente, com as informações do perfil do usuário no ambiente, o nível de frequência, o tipo de ambiente, e parâmetros temporais (dia de semana, dia útil, turno) o tipo de acesso do usuário e, subsequentemente, o nível de acesso do usuário.

Através do nível de acesso do usuário as ações padrão para o ambiente são obtidas, e utilizando a identificação do ambiente as regras customizadas são obtidas. Ao final do processo as regras padrão são substituídas pelas regras customizadas (merge), caso haja relação direta entra elas (ações direcionadas para a mesma funcionalidade), e então devolvidas para o usuário para serem aplicadas no dispositivo.

Figura C- Processo de negócio sobre a mudança de localização do usuário



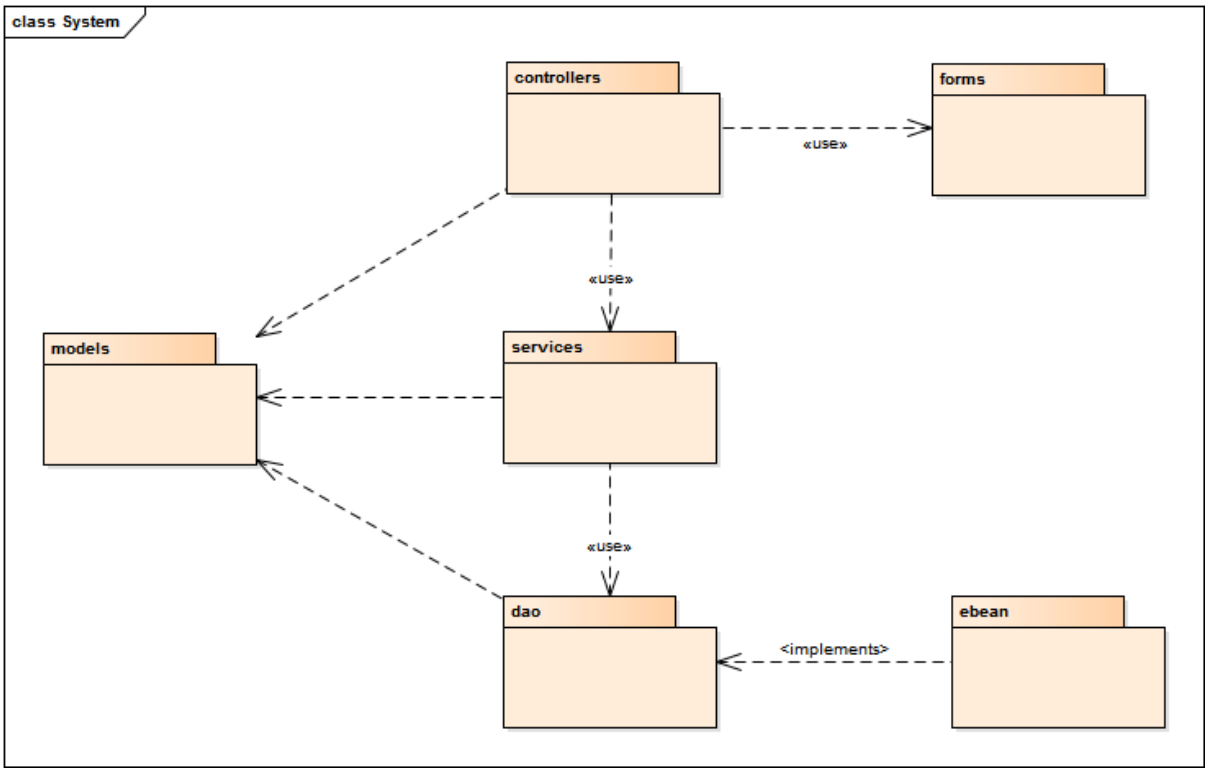
Fmte:Do aula:

A.4. Arquitetura

A Figura 2 apresenta as camadas do sistema. O ponto de entrada é a camada de ***controllers***. Eles são responsáveis por receber as requisições dos clientes, extrair e validar os dados obtidos das requisições, repassar os dados da requisição para a camada de ***services*** ou ***data access layer (DAO)*** e retornar o resultado para o cliente no formato correto. A validação dos dados de entrada é feita através de classes da camada *forms*, que definem os nomes e tipos dos campos de entrada, além de definirem outras regras de validação.

Funções de CRUD (*create, read, update and delete*) não necessitam de uma camada de serviço, a menos que estas operações simples impliquem no disparo de outras operações e/ou eventos. Qualquer tarefa que vá além destas operações simples está localizada na camada de ***services***.

Figura C - Processo de negócio sobre a mudança de localização do usuário



Fonte: Do autor.

A camada dao define as interfaces para acesso a dados provenientes do banco de dados. A relação entre as entidades da camada model e as classes da camada de dao é 1:1, ou seja, para cada entidade do modelo existe uma classe para acesso a dados. O subpacote ebean é a implementação utilizada para a camada de acesso a dados.

A camada de services é responsável pelos processos mais complexos do sistema, ela faz uso de diversas classes da camada dao e model. Nas seções abaixo serão detalhadas as classes que compõe as camadas dao e services.

A.4.1 Services

A camada de serviços possui três classes responsáveis por processos relacionados à privacidade dos ambientes (*IPrivacyService*), autenticação de usuários (*Authenticator*) e relógio do sistema (*IClock*). Na Figura 3 estão ilustradas estas três interfaces, bem como a implementação padrão para elas, utiliza no servidor de produção.

O uso de interfaces para os serviços facilita a criação de testes unitários e de integração, permitindo a utilização de implementações alternativas para que os testes possam ser realizadas para um componente sem a presença de outros componentes.

Isso se aplica principalmente ao serviço de autenticação. A implementação em produção depende de um sistema externo para autenticar os usuários, Para tanto, foi utilizado a base de dados do Sistema Integrado de Geranciamento Acadêmico Inteligente (SIGA-i), que atua como um *Identity Provider*. Para que testes de integração possam ser executados sem a necessidade de o SIGA-i estar acessível, uma implementação alternativa do serviço de autenticação é utilizada.

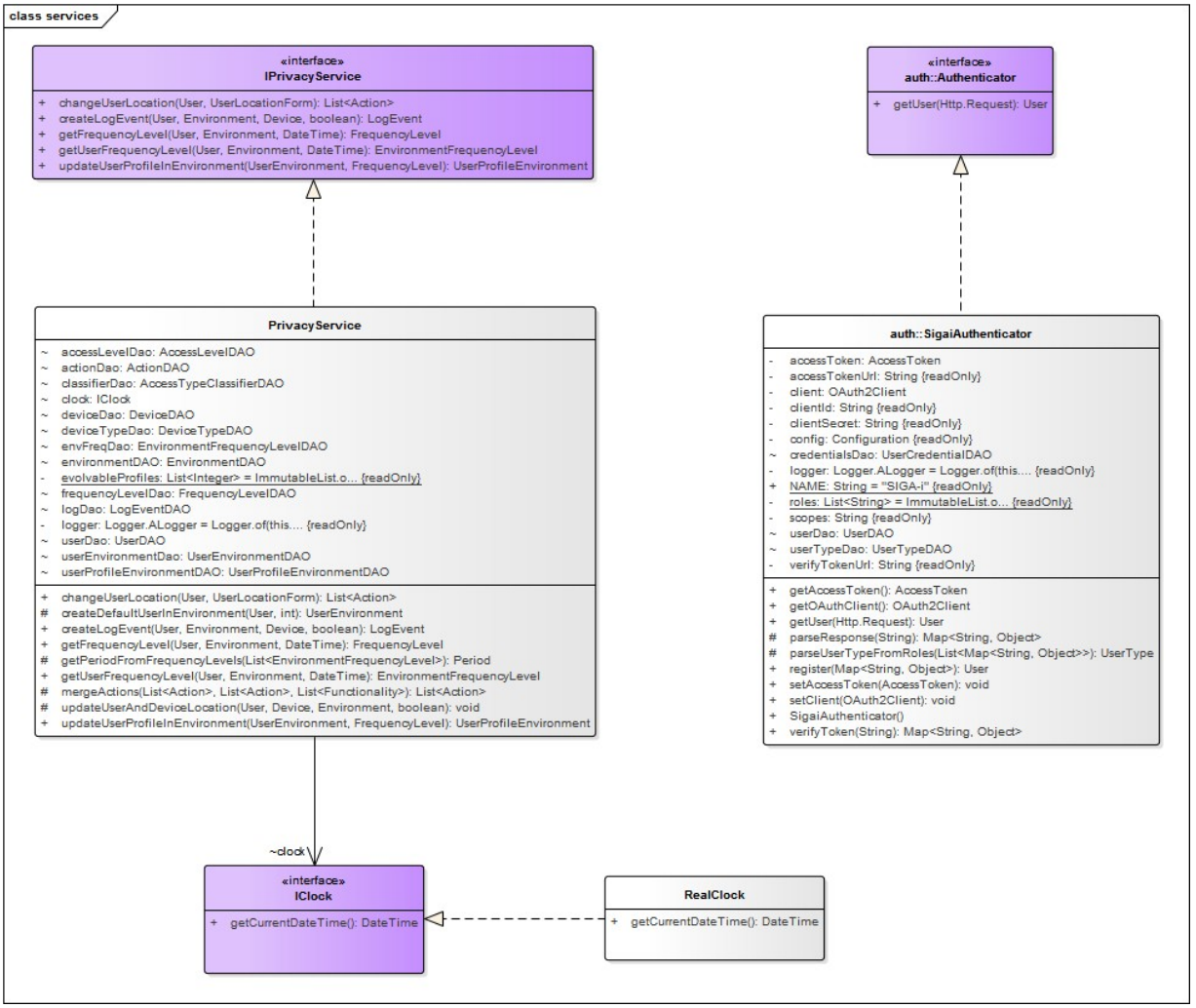
O serviço de autenticação define apenas um método que recebe como parâmetro o objeto de requisição HTTP recebido do cliente e retorna um objeto usuário caso a autenticação seja bem sucedida, ou um *null* caso contrário. A implementação para o SIGA-i (*SigaiAuthenticator*) extraí o cabeçalho que contém o token de autenticação do O *Authenticator* 2.0, verifica com o SIGA-i se o token é valido e retorna o usuário ao qual o token pertence.

O principal serviço do sistema é o de privacidade (*IPrivacyService*). Ele define os métodos para quando o usuário muda de ambiente, criação de log de mudança de ambiente, cálculo do nível de frequência de um usuário no ambiente, e atualização do perfil do usuário no ambiente com base no nível de frequência.

A.4.2. Data Access Object (DAO)

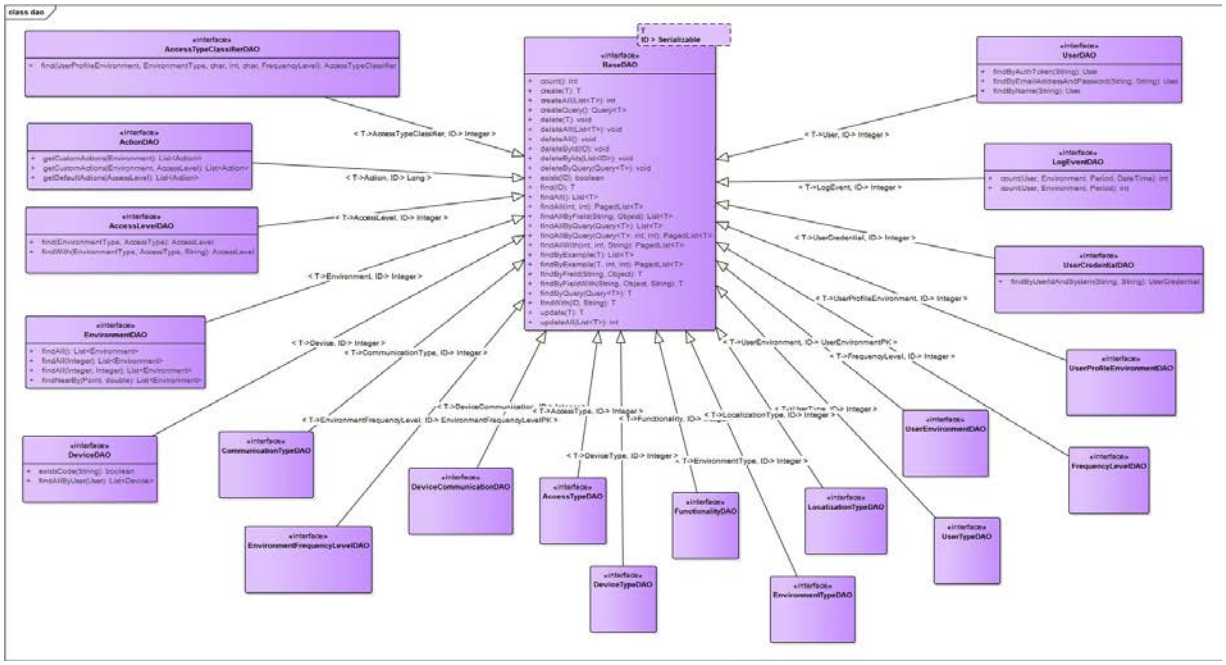
A camada de acesso a dados define as interfaces para acesso a dados das entidades da camada ***model***. Todas as interfaces estendem a interface *BaseDAO*, que define os métodos básicos para criação, atualização, consulta e remoção de dados. As interfaces para as entidades apenas definem métodos complementares, normalmente consultas mais complexas que não podem ser expressas com os métodos padrão da *BaseDAO*. Além disso, tal interface ainda recebe como tipos genéricos o tipo de entidade que ele irá gerenciar e o tipo da chave primária desta entidade

Figura D - Camada de Serviços



Fonte: Do autor.

Figura E - Camada DAO (*Data Access Objects*)



Fonte: Do autor.

A.5. API para Troca de Mensagens

A interface de comunicação entre o servidor e os clientes é um Web Service RESTful com suporte ao formato de dados JSON. A tabela abaixo lista os caminhos existentes, método utilizado e controlador que lida com a requisição. Tais interfaces são detalhadas posteriormente.

Tabela B – Interface da API REST

Ref.	Método	Recurso	Controller
API00	GET	/user	UserController@getUser
API01	PUT	/user	UserController@updateUser
API02	GET	/user/devices	UserController@listDevices
API03	POST	/user/devices	UserController@addDevice
API04	PUT	/user/location	UserController @updateLocation
API05	GET	/user/devices/:code	DeviceController @getDevice
API06	PUT	/user/devices/:code	DeviceController @updateDevice
API07	GET	/user/devices/:code/functionalities	DeviceController @listFunctionalities
API08	GET	/environments	EnvironmentController @getAll
API09	GET	/environments/:id	EnvironmentController@get

As interfaces são descritas conforme apresentados nos códigos, conforme segue:

GET /user

Retorna informações sobre o usuário logado.

Resposta:

```
1  {
2      "id" :  1 ,
3      "name" :  "valderi" ,
4      "fullName" : "Valderi R. Q. Leithardt" ,
5      "emailAddress" : "valderi@gmail.com" ,
6      "currentEnvironment" : null,
7      "userType" : {
8      "id" :1,
9      "name" : "Normal"
10     } ,
11     "devices" :  [ ]
12 }
```

PUT /user

Atualiza as informações do o usuário logado.

Corpo da requisição:

```
1  {
2      "name" :"valderi" ,
3      "fullName" : "Valderi Leithardt" ,
4      "emailAddress" : "valderi@gmail.com "
5  }
```

Resposta:

```
1  {
2      "id" : 1 ,
3      "name" : "valderi" ,
4      "fullName" : "Valderi Leithardt" ,
5      "emailAddress" : "valderi@gmail.com" ,
6      "currentEnvironment" : null,
7      "userType" : {
8          "id" : 1 ,
9          "name" :"Normal"
10     } ,
11     "devices" : [ ]
12 }
```

GET /user/devices

Lista os dispositivos do usuário.

Resposta:

```
[
    {
        "id" : 41 ,
        "code" :  " 6 9 8DC1 9D4 8 9C4E4DB7 3E2 8A7 1 3EAB0  7B" ,
        "name" : "Galaxy S5" ,
        "type" :  {
            "id" :  1 ,
            "name" :"Android"
        }
    }
]
```

POST /user/devices

Adiciona um novo dispositivo para o usuário.

Corpo da requisição:

```
{
    "code" : " 6 9 8DC1 9D4 8 9C4E4DB7 3E2 8A7 1 3EAB0  7B" ,
    "name" :" Galaxy S5" ,
    "deviceType" : "Android"
}
```

Resposta:

```
{
"id" : 41,
"code" :  " 6 9 8DC1 9D4 8 9C4E4DB7 3E2 8A7 1 3EAB0 7B" ,
"name" :  "Galaxy S5" ,
"type" : {
    "id" : 1,
    "name" :  "Android"
    } ,
    "currentEnvironment" : null ,
    "functionalities" : [ ] ,
    "communications" :  [ ]
}
```

Retorna informações do dispositivo do usuário identificado pelo código code.

Resposta:

```
{
    "id" : 41,
    "code" :   " 6 9 8DC1 9D4 8 9C4E4DB7 3E2 8A7 1 3EAB0 7B" ,
    "name" : "Galaxy S5" ,
    "type" : {
        "id" : 1 ,
        "name" : "Android"
    } ,
    "currentEnvironment" : null ,
    "functionalities" :  [
        { "id": 1, "name" : "WiFi"   }
    ] ,
    "communications" :  [
        {  "id" : 1, "name" : "GPS"  }
    ]
}
```

PUT /user/location

Atualiza a localização do usuário e dispositivo e retorna uma lista de ações a serem aplicadas no dispositivo.

Corpo da requisição:

```
{
    "deviceCode" : "6 9 8DC1 9D4 8 9C4E4DB7 3E2 8A7 1 3EAB0 7B" ,
    "environmentId" : 1 ,
    "exiting" : false
}
```

Resposta:

```
[
    {
        "id" : 57 ,
        "accessLevel" : {
        "id" : 8 ,
        "impactFactor" : 0.0 ,
        "accessType" : {
        "id" : 5 ,
        "name" : "Administrative"
        } ,
        "environmentType" : {
        "id" : 3 ,
        " name " : "Public"
        }
        } ,
        "functionality" : {
        "id" : 5 ,
        "name" : " Wi−F i "
        } ,
```

```
        "environment" : null ,
        "action" : "on" ,
        "startDate" : null ,
        "endDate" : null ,
        "startDailyInterval" : null ,
        "durationInterval" : null ,
        "args" : [ ]
    }
]
```

GET /environments

Retorna lista de ambientes.

Parâmetros de consulta:

Tabela C - Parâmetros de consulta de ambientes

Parâmetro	Tipo	Descrição	Controller
lat	double	Latitude do ponto central para busca	null
lon	double	Longitude do ponto central para busca	null
radius	double	Tamanho do raio em metros do ponto central para busca	10
limit	double	Limite de ambientes a ser retornado (não se aplica a busca por localização)	50

Resposta:

```
[
{
        "id": 1,
        "name" : "Porto Alegre" ,
        "location" : {
        "type" : "Point " ,
        "coordinates" :  [ −30.072296142578118 ,−51.17763595581054,  10.0  ]
        } ,
        "shape" : {
        "type" : "Polygon" ,
        "coordinates" :  [
        [
        [ −51.198184967041 , −29.9612808227539, 0.0 ] ,
        [  −51.2952117919922,  −30.1073989868164 , 0.0  ] ,
        [ −51.216136932373  , −30.2264022827148, 0.0 ] ,
        [ −51.0650444030762, −30.0949935913086,  0.0  ] ,
        [ −51.1136016845703,  −29.9714050292969,  0.0  ] ,
```

```
18
19          [  −51.198184967041,−29.9612808227539,0. 0   ]
]
20          ]
21          } ,
22              "operatingRange" : 17550.786 ,
23              "version" : 1,
24              "localizationType" : {
25              "id" : 1 ,
26              "name" : "GPS" ,
27              "precision" : 600.0 ,
28              "metric" : "m2 "
29          } ,
30              "environmentType" : {
31              "id" : 3 ,
32              "name" : "Public"
33          } ,
34          "parentId " : null,
35          "level": 0,
36          "customActions" :   [   ] ,
37          "distance" : 0.0
38      }
39  ]
```

GET /environments/:id

Retorna as informações do ambiente com a id informada. O formato da resposta é o mesmo da listagem de ambientes, apenas com a ausência do array.

Erros do Cliente

Esta seção apresenta os tipos de erro que um cliente da API poderá receber, bem como a descrição do motivo da ocorrência do erro.

HTTP/1.1 404 Not Found

Este erro ocorre quando a entidade buscada não foi encontrada no servidor. Corpo da mensagem:

```
1   { "message" : "<entity>not found." }
```

HTTP/1.1 409 Conflict

Este erro ocorre quando a entidade a ser criada já existe no servidor. Corpo da mensagem:

```
1 {"message":"<entity> already exists."}
```

HTTP/1.1 422 Unprocessable Entity

```
1 [
2     { "field_name" : ["Error message" ] }
3 ]
```

A.6. Implementação

Este capítulo apresenta os detalhes de implementação do sistema, com a especificação das plataformas, frameworks e bibliotecas utilizadas.

O servidor foi desenvolvido em Java com a Play Framework para aplicações Web. Todos os dados do sistema são armazenados em um banco de dados PostreSQL com a extensão PostGIS para armazenamento e consulta de dados espaciais.

As camadas de services e dao se aproveitam do suporte a injeção de dependências do Play através da definição de interfaces genéricas e implementações padrão para todas as classes. Na prática isso significa que a implementação de um componente de uma destas camadas pode ser alterado sem que outras partes do sistema precisem ser modificadas, contanto que a nova implementação obedeça o contrato já estabelecido.

Para dar um exemplo concreto, atualmente o servidor utiliza para o acesso ao banco de dados a biblioteca Ebean ORM. Trocar o Ebean por outra ORM, como o Hibernate, significaria apenas criar uma implementação diferente das interfaces da dao e alterar a configuração do injetor de dependência.

A autenticação do sistema é feita através do protocolo OAuth 2.0 juntamente com o Identity Provider design pattern. O sistema faz uso de um terceiro para autenticar e autorizar os usuários, neste caso o sistema SIGA-i, um sistema de gestão acadêmica também desenvolvido pelo Grupo GPPDi.

Os testes unitários fazem uso do JUnit em conjunto com o Mockito para criação de mocks de classes em testes unitários. Os testes de aceitação também fazem uso do Guava

para emular uma aplicação Play completa para testes que envolvem mais de um componente, inclusive com acesso a um banco de dados de testes.

Tabela C - Softwares utilizados pelo servidor e suas versões

Software	**Versão**
Oracle Java JDK	1.8.0_45
Play Framework	2.4
PostgreSQL	9.3.6
PostGIS	2.1.2

A estrutura do banco de dados é controlada por evoluções, arquivos SQL que definem modificações para a estrutura do banco de dados, além das operações que devem ser aplicadas para desfazer tais alterações. Para os testes ainda foram definidas fixtures que contém dados para serem carregados automaticamente no banco de dados. As fixtures são definidas em arquivo no formato YAML.

A API REST suport até o momento apenas o formato JSON. A serialização/-deserialização de objetos é feita através da biblioteca Jackson JSON. Objetos simples utilizam as annotations da biblioteca para controle do formato de serialização, enquanto que objetos mais complexos, como objetos do PostGIS (Point, Polygon), fazem uso de serializadores/deserializadores customizados (JsonDeserializer e JsonSerializer).

A.7. Administração

Além da API REST o servidor também fornece um painel de administração para edição de ambientes. A Figura 5 mostra a tela de listagem de ambientes.

Na edição dos dados do ambiente (Figura 7) é possível alterar o nome do ambiente, tipo de ambiente e tipo de localização.

Figura E - Lista de Ambientes

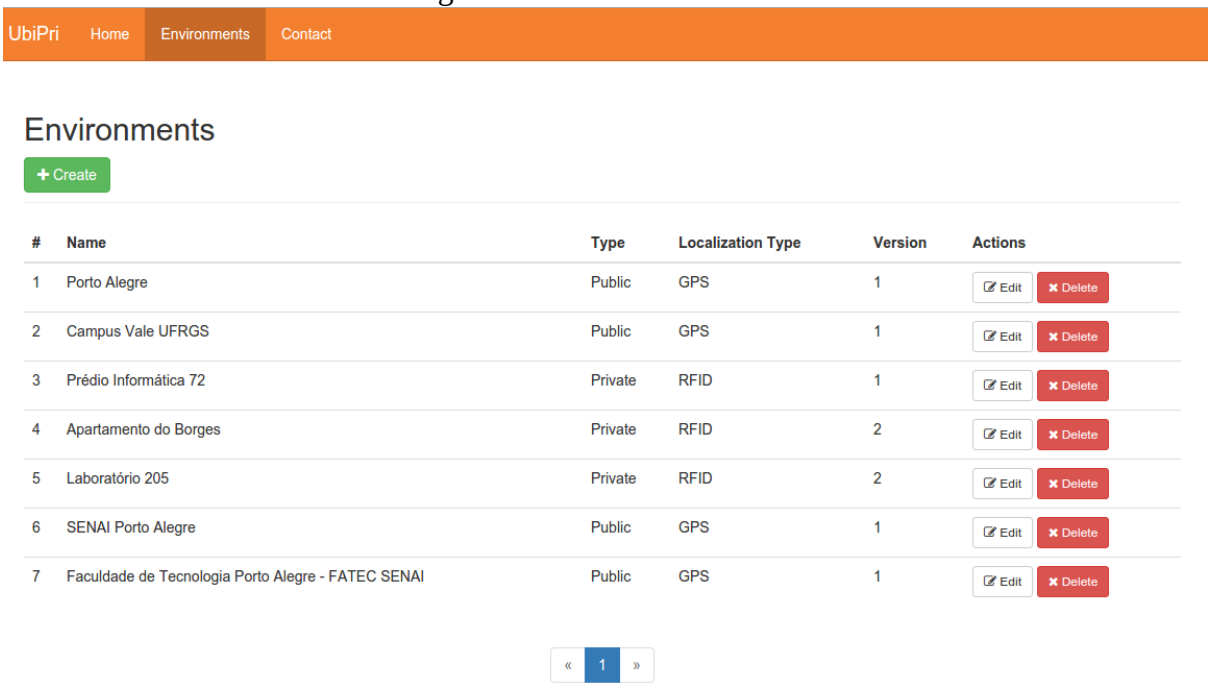
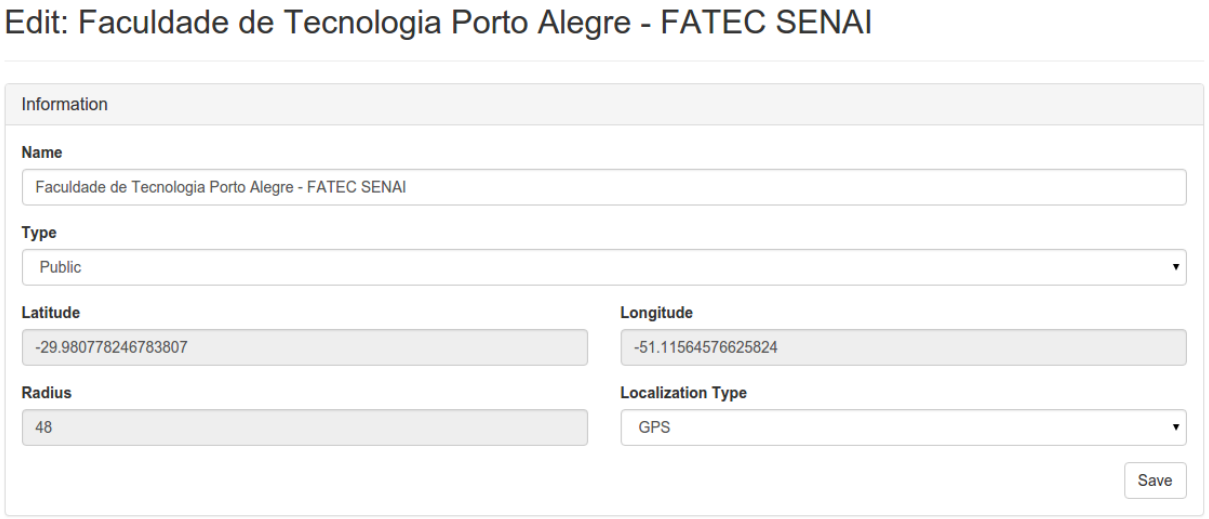


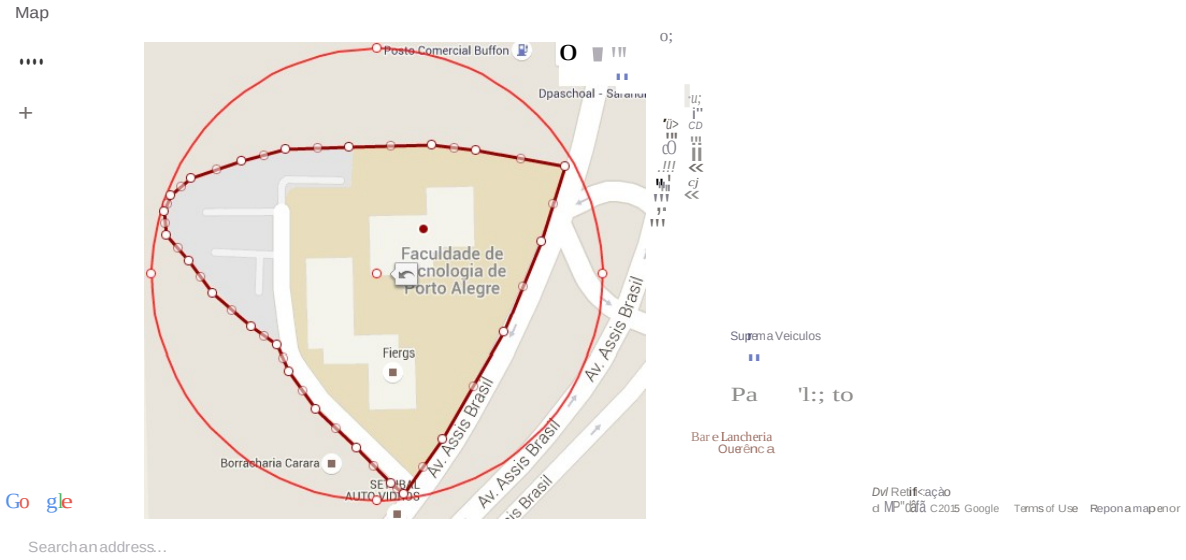
Figura F - Edição de Dados do Ambiente



Já os dados da localização são editados apenas pelo mapa, com a possibilidade de definir o ponto central do ambiente, o raio de alcance do ambiente e, opcionalmente, o desenho do ambiente na forma de polígono. O mapa e as ferramentas de desenho são fornecidos pelo Google Maps, e os polígonos são transformados em WKT pela biblioteca Wicket[1].

[1] https://github.com/arthur-e/Wicket

Figura G- Edição de Mapa do Ambiente



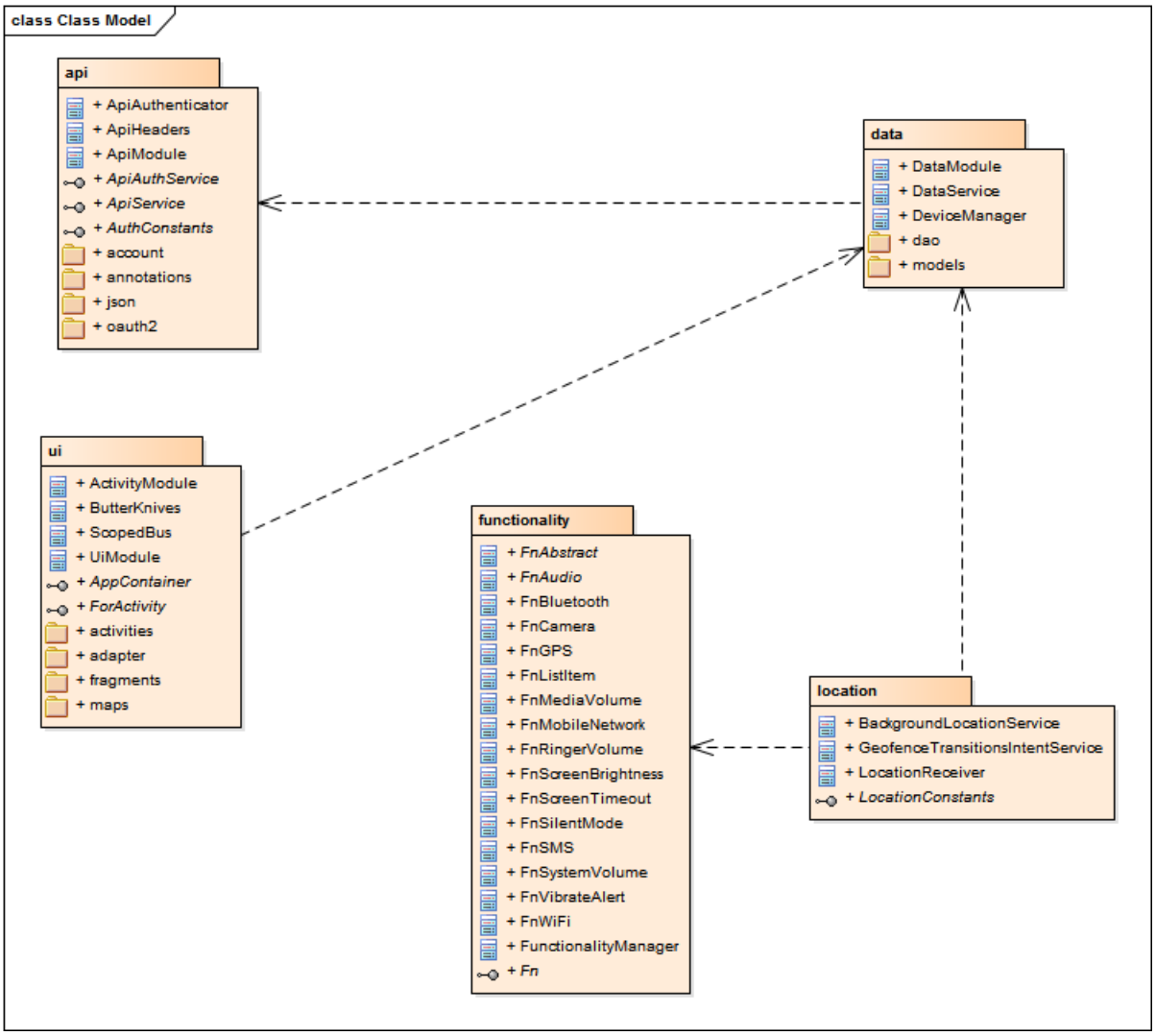
ANEXO B: CLIENTE UBIPRI

B.1. Android

O cliente do sistema UbiPri foi desenvolvido para a plataforma móvel Android, com suporte para todas as versões a partir da Ice Cream Sandwich (4.0.3).

O projeto foi desenvolvido utilizando a IDE Android Studio, ferramenta oficial para desenvolvimento na plataforma. A estrutura do projeto está dividída em cinco módulos principais, como pode ser visto na Figura H.

Figura H - Estrutura da Aplicação



O módulo api é responsável pela comunicação entre a aplicação e o servidor, fazendo para tanto uso da biblioteca Retrofit[2] , um cliente HTTP para Android que permite a criação de clientes de API através da definição de uma interface Java com annotations para especificação de URLs e métodos HTTP. A biblioteca fica responsável pela realização das chamadas HTTP, serialização e deserialização de objetos, verificações de segurança, etc.

Dentro da api também foi feita a implementação do AccountManager para gerenciamento de contas utilizando o protocolo OAuth 2.0. Através deste componente o usuário faz a autenticação na aplicação e partir deste momento este componente se encarrega de salvar os tokens de acesso e utilizá-los nas chamadas HTTP para o servidor.

O módulo data agrega a camada de modelos e DAO, além de prover uma interface integrada na forma de um facade para as principais operações da aplicação, envolvendo o acesso a API no servidor, aplicação de lógica de negócios e armazenamento de dados no dispositivo móvel.

O armazenamento de dados fica a cargo dos componentes da DAO. Internamente eles fazem uso da bibliocata ActiveAndroid[3], que implementa o pattern Active Record, para armazenamento de dados no SQLite.

Para controlar as funcionalidades dos dispositivos, recurso necessário para a aplicação das regras do ambiente (e.g. ligar/desligar o GPS, ativar/desativar o modo silencioso), existe o módulo functionality. Nele é definida uma interface padrão para controle de funcionalidades de forma individual (uma implementação por funcionalidade). Também fora criado um gerenciador de funcionalidades (FunctionalityManager) que permite a aplicação em massa de regras do ambiente, além de verificação de quais funcionalidades estão disponíveis no dispositivo.

No módulo location estão definidos os background services e broadcast receivers que dão inicio ao monitoramento de localização (utilizando o componente FusedLocationProviderApi), carregamento e monitoramento de geofences, ambas tarefas iniciadas pelo BackgroundLocationService.

Uma vez iniciadas as tarefas de monitoramento, o componente LocationReceiver passa a receber em intervalos fixos de tempo a localização atual do dispositivo. O monitoramento contínuo da localização também auxilia na melhora da precisão da localização para o serviço que monitora as geofences, este tratado pelo componente GeofenceTransitionsIntentService.

---

[2] http://square.github.io/retrofit/
[3] https://github.com/pardom/ActiveAndroid

O GeofenceTransitionsIntentService recebe notificações de entrada/saída de geofences carregadas previamente. O aplicaitivo define 99 geofences provenientes do servidor e uma centésima geofence chamada de Master Geofence. Quando for detectado que o usuário saiu da Master Geofence, o componente irá disparar um Intent para o BackgroundLocationService atualizar a lista de geofences com base na nova localização do dispositivo.

Existe ainda a opção, desativada por padrão, para verificar se o usuário está dentro de um ambiente com base no polígono que descreve a área do ambiente através da biblioteca Spatial4j[4] . Esta funcionalidade é complementar a detecção de geofences e só é útil quando a precisão da localização do dispositivo é muito alta.

Por fim, o módulo da ui define todos os componentes que fazem parte da interface de usuário da aplicação, como activities, adapters, fragments, etc. Componentes da ui fazem uso da biblioteca Butter Knife[5] para injeção de componentes de layout.

Comum a todos os módulos da aplicação está a existência de classes com sufixo Module. Elas definem dependências a serem usadas por outros componentes da aplicação através de injeção de dependência (DI), funcionalidade provida pela biblioteca Dagger[6].

Através do uso de injeção de dependências fica mais fácil a criação de testes sem a necessidade, por exemplo, de se ter um servidor funcionando para receber as chamadas HTTP, já que a API do servidor pode ser emulada.

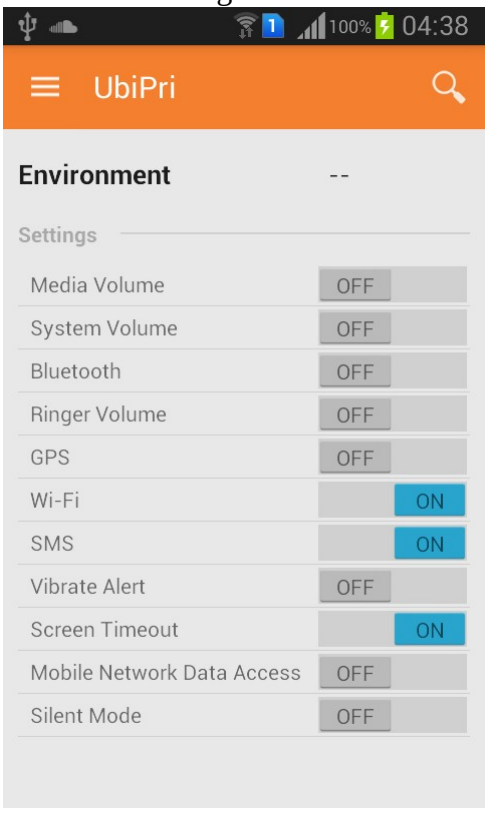
B.2. Resultados

O UbiPri foi publicado na Google Play. Uma demonstração foi realizada durante o Mundo SENAI 2015 na Faculdade de Tecnologia de Porto Alegre (FATEC). O ambiente fora configurado para que o toque dos telefones celulares fosse colocado em modo silencioso quando detectado que o usuário havia adentrado o ambiente.

A Figura J apresenta os logs de entrada e saída de usuários durante o Mundo SENAI ao longo do tempo. As barras em azul indicam a entrada de usuários enquanto que as vermelhas indicam que o usuário saiu do ambiente.

[4] https://github.com/locationtech/spatial4j
[5] https://github.com/JakeWharton/butterknife
[6] http://square.github.io/dagger/

Figura I - Screenshots do Aplicativo para A ndroid



(a) Tela Principal

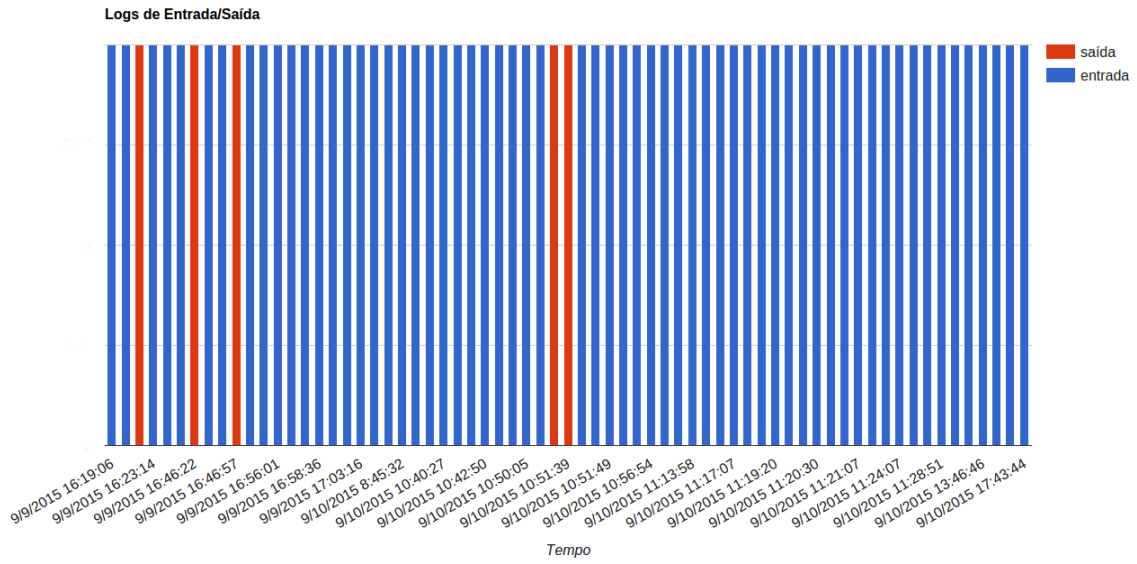
(b) Splash

B.3. Trabalhos Futuros

Uma das constatações feitas através do experimento do aplicativo foi da qualidade em termos de precisão da detecção da localização através de GPS. Mesmo com o uso de múltiplas técnicas (GPS em conjunto com informações de antenas de telefone e WiFi) a precisão da localização não é suficiente para garantir com total certeza que um usuário está ou não dentro de um ambiente.

Além disso, ambientes in-doors impossibilitam o uso do GPS o que reduz mais ainda a precisão da detecção da localização. Outra implicação do GPS é o aumento no consumo da bateria, somado ao tempo necessário para que a localização tenha uma precisão aceitável. Todos estes fatores demonstram que é necessária a utilização de métodos alternativos para detecção de ambientes.

Figura J – Log de Entradas e Saídas durante o Mundo SENAI



Dentre estas técnicas existem NFC, RFID, redes WiFi e Bluetooth. Nas duas primeiras técnicas o usuário fica encarregado de realizar o check-in no ambiente, enquanto que nas duas últimas técnicas bastaria o usuário estar dentro do alcance de uma rede cadastrada no sistema para que fosse considerado como dentro do ambiente.

Apesar de o controle sobre o alcance de redes WiFi e Bluetooth ser limitado (vários fatores de influência, alguns deles difíceis de superar), ainda assim seria possível definir o nível de força do sinal para que o usuário fosse considerado dentro de um ambiente, ou mesmo através do uso de mais de uma rede.

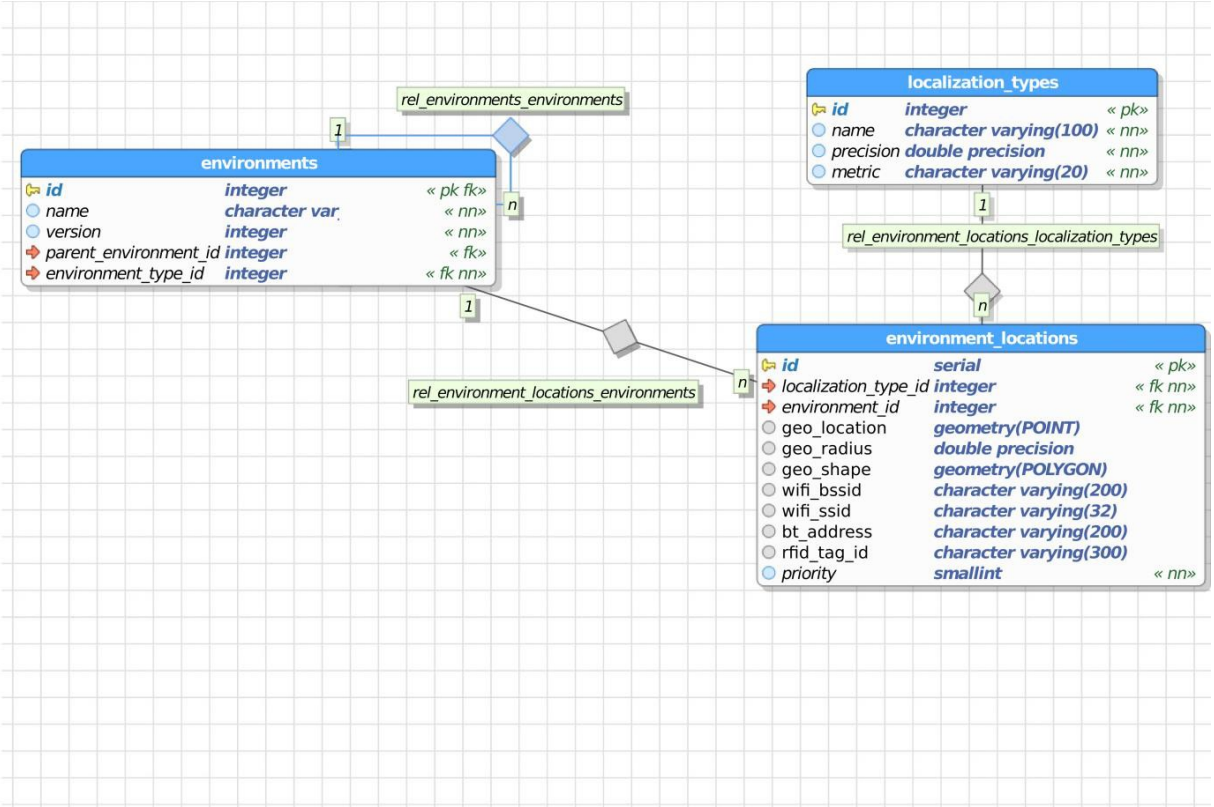
A Figura K mostra a alteração no banco de dados que possibilita o suporte a múltiplas técnicas de localização. O que ocorreu foi a adição da tabela environment_locations entre a tabela environments e localization_types.

Na tabela localization_types ficam armazenadas as técnicas utilizadas para localização (geolocation, WiFi, Bluetooth, RFID, NFC), em conjunto com a precisão entregada.

Já na nova tabela, environment_locations, são armazenadas as técnicas de localização que poderão ser utilizadas na detecção de determinado ambiente. Um ambiente pode ter mais de uma técnica de localização, e ele pode ainda ter vários registros nesta tabela com a mesma técnica de localização, o que ocorre nos casos onde diversas redes de WiFi podem ser utilizadas para detectar o ambiente.

A escolha da técnica preferencial no lado do cliente é definida através da prioridade, valor inteiro que inicia em um (maior prioridade) e é incrementado a medida do necessário (menor prioridade).

Figura K – Adaptação do banco de dados para suporte a múltiplas técnicas de localização



Para manter a estrutura do banco de dados simples e evitar uso de joins os campos de dados de todas as técnicas existentes foram colocados na mesma tabela, cabendo assim a parte servidora remover os campos que não são relevantes a determinada técnica ao devolver o resultado ao cliente.

Neste cenário o servidor devolveria ao cliente uma resposta com a lista de ambientes no seguinte formato:

```
1 {
2     "id":1,
3     "name":"Porto Alegre",
4     "version":1,
5     "environmentType":{
6         "id":3,
```

```
121

7         "name":"Public"
8      },
9      "parentId":null,
10     "level":0,
11     "customActions": [ ],
12     "locations": [
13        {
14           "type": "Geolocation",
15           "radius":17550.786,
16           "location": {
17              "type":"Point",
18              "coordinates" : [ −30.072296142578118, −51.17763595581054 , 10.0 ]
19           },
20           "shape": {
21           "type":"Polygon",
22           "coordinates": [
23              [
24                 [ −51.198184967041 , −29.96128082275 39, 0.0 ]
,
25                 [ −51.295211791992 2, −30.107398986816 4, 0.0 ]
,
26                 [ −51.216136932373 , −30.226402282714 8, 0.0 ],
27                 [ −51.065044403076 2, −30.094993591308 6,0.0 ]
,
28                 [ −51.113601684570 3, −29.971405029296 9,0.0 ],
29                 [ −51.198184967041 , −29.96128082275 39,0.0]
30              ]
31           ]
32           },
33        },
34           {
35              "type": "WiFi",
36              "bssid": "00:09:5B:1C:AA:1D",
37              "ssid": "POA−Net"
```

```
38                  }
39
40  }
```