

Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Pós-Graduação em Ciência da Computação

Uma Plataforma para Integrar Dispositivos
Eletrônicos em Ambientes Pervasivos

Luís Henrique Leal Ries

Orientador: Prof. Dr. Fabiano Passuelo Hessel

Porto Alegre, Agosto de 2007

LUÍS HENRIQUE LEAL RIES

**Uma Plataforma para Integrar Dispositivos
Eletrônicos em Ambientes Pervasivos**

Dissertação apresentada como
requisito para obtenção do grau
de Mestre, pelo Programa de
Pós-Graduação em Ciência da
Computação da Faculdade de
Informática da Pontifícia Universi-
dade Católica do Rio Grande do Sul

Orientador: Prof. Dr. Fabiano Passuelo Hessel

Porto Alegre, Agosto de 2007

Dados Internacionais de Catalogação na Publicação (CIP)

R559p Ries, Luís Henrique
Uma plataforma para integrar dispositivos eletrônicos
em ambientes pervasivos / Luís Henrique Ries. – Porto
Alegre, 2007.
80 f.
Diss. (Mestrado) – Fac. de Informática, PUCRS
Orientador: Prof. Dr. Fabiano Passuelo Hessel
1. Informática. 2. Computação Pervasiva.
3. Arquitetura de Computador. 4. Dispositivos Eletrônicos.
I. Título.

CDD 004.21

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Uma Plataforma para Integrar Dispositivos Eletrônicos em Ambientes Pervasivos**", apresentada por Luís Henrique Leal Ries, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas Embarcados e Sistemas Digitais, aprovada em 22/03/2007 pela Comissão Examinadora:

Prof. Dr. Fabiano Passuelo Hessel –
Orientador

PPGCC/PUCRS

Prof. Dr. Eduardo Augusto Bezerra –

PPGCC/PUCRS

Prof. Dr. Jorge Luís Victória Barbosa –

UNISINOS

Homologada em 09/08/07, conforme Ata No. 18 pela Comissão Coordenadora.

Prof. Dr. Fernando Luís Dotti
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 – P. 32 – sala 507 – CEP: 90619-900

Fone: (51) 3320-3611 – Fax (51) 3320-3621

E-mail: ppgcc@inf.pucrs.br

www.pucrs.br/facin/pos

*A minha esposa e a meus pais, pelo carinho,
compreensão, apoio e dedicação nesse trabalho.*

Agradecimentos

Essa seção é voltada para eu homenagear todos aqueles que me ajudaram e estiveram ao meu lado, nos momentos difíceis e descontraídos. Às vezes, uma palavra, um carinho ou até um gesto pode iluminar uma pessoa e destiná-la a um novo caminho. Por isso, gostaria de agradecer e dedicar esta dissertação a todos aqueles que, de forma direta ou indireta, me auxiliaram nessa jornada. Em especial, eu agradeço:

a Deus pela força e pela luz que guiou meus caminhos e permitiu a realização de mais um sonho na minha vida.

a minha esposa, Andréa, por estar ao meu lado incentivando nos momentos difíceis.

a meus pais, Vitor Hugo e Beatriz, que deram o rumo certo da vida, me educando com amor, dignidade e respeito.

ao meus irmãos, familiares e parentes por palavras consoladoras e incentivadoras.

ao meu orientador, Fabiano Hessel, que me auxiliou nas tomadas de decisão e teve grande paciência esclarecendo dúvidas e corrigindo erros correntes não só do trabalho, mas da vida.

aos outros professores do PPGCC, principalmente àqueles da área de Sistemas Digitais que sempre deram apoio e incentivo nas atividades.

aos meus companheiros de trabalho do GSE e do GAPH, especialmente Márcio Martins (grande irmão nesses dois anos), Gabriel, Rohde, David, Márcio Farias, Laerte, Zé, Everson e Leonel por me auxiliar a resolver problemas ao longo da trajetória e por momentos alegres.

aos meus amigos do "Brooklin" pela amizade e pelos momentos descontraídos.

ao pessoal da Unisinos que me recebeu de portas abertas para o desenvolvimento da aplicação Agenda Pervasiva para esse trabalho.

e ao CNPQ pelo apoio financeiro ao projeto PNM (Projeto Nacional de Microeletrônica).

Por fim, agradeço também a minha inspiração e a força de vontade para finalizar esse trabalho.

Em especial, gostaria de agradecer a minha tia, madrinha e segunda mãe, Elza Marques.

Obrigado tia, por sempre me iluminar, seja neste ou em outro plano.

Resumo

Atualmente, grupos de pesquisa da área acadêmica e da indústria tem se direcionado a um novo paradigma. Computação pervasiva ou computação ubíqua permite o desenvolvimento de aplicações inteligentes tais como casas inteligentes, celulares inteligentes e computadores embutidos em trajes. O objetivo da computação pervasiva é fornecer acesso computacional de maneira invisível e onipresente. No entanto, a implementação de uma arquitetura para essas aplicações implica em muitos requisitos e desafios. Um desses desafios é como permitir comunicação entre diversos dispositivos heterogêneos em um ambiente pervasivo. Com esses dispositivos, a rede pervasiva torna-se mais heterogênea e com diferentes protocolos e sub-redes de comunicação. Essa heterogeneidade pode gerar problemas de qualidade e de atraso na rede. Com isso, a aplicação não fornece o princípio de invisibilidade. O objetivo desse trabalho é a proposição de uma plataforma hardware e software que seja capaz de integrar dispositivos em ambientes pervasivos. Essa plataforma permite os dispositivos compartilharem serviços e recursos e se comunicarem, entre si e com a rede. Para avaliar essa plataforma, uma aplicação chamada Agenda Pervasiva é desenvolvida a fim de mostrar a execução e as contribuições da plataforma proposta.

Palavras-chave: Computação Pervasiva. Arquitetura. Plataforma Hardware e Software. Dispositivos. Ambientes Pervasivos.

Abstract

In recent years, research groups in both academia and industry have driven a new paradigm. Pervasive computing or ubiquitous computing allows smart application designs such as smart house, smart phones and wearable computing. The goal of pervasive computing is to provide computational access in invisible and ubiquitous way. However, architecture design for these applications considers a set of new requirements and challenges. One of these challenges is how to enable communication between various heterogeneous devices on a pervasive environment. Because of these devices, the pervasive network becomes more heterogeneous and with several communication protocols and subnets. This heterogeneity can generate quality and delay problems on the network. Therefore, it disables the invisibility principle of application. This work proposes a hardware and software platform for integrating devices into pervasive environments. This platform enable devices share services and resources and communicate with each other and with network. To evaluate this platform, the Pervasive Scheduler application is designed in order to show contributions of the proposed platform.

Keywords: Pervasive Computing. Architecture. Hardware and Software Platform. Devices. Pervasive Environments.

Lista de Figuras

1.1	Exemplo de uma casa inteligente	16
2.1	Rede atuando em uma residência.	22
2.2	Estrutura de uma Arquitetura para um Ambiente Pervasivo.	23
2.3	Fisionomia e Estrutura de uma Plataforma HW/SW.	25
3.1	Formação de um dispositivo interativo: (a) Blocos de isopor para formar a infra-estrutura do dispositivo; (b) Plataformas Calder Toolkit; (c) Dispositivo interativo.	33
4.1	Arquitetura do Trabalho Estruturada em Camadas.	37
4.2	Infra-estrutura HW para Residência Inteligente.	38
4.3	Funcionamento do Gerenciador de Contexto na Arquitetura.	43
5.1	Modelo de Plataforma.	46
5.2	Plataforma do Trabalho Estruturada em Camadas.	47
5.3	Fisionomia da Plataforma OMAP.	48
5.4	Diagrama de Classes do Módulo de Comunicação.	51
5.5	Diagrama de Classes do Gerenciador de Serviços.	52
5.6	Diagrama de Classes da Unidade de Reconfiguração.	53
5.7	Diagrama de Classes do Sistema de Autenticação e Controle de Acesso.	53
5.8	Diagrama de Classes do Auxiliar de Contexto.	54
5.9	Diagrama de Classes do Monitor.	55
6.1	Infra-estrutura WIFI da aplicação Agenda Pervasiva.	59
6.2	Serviços específicos para administrador no Terminal de Acesso.	61
6.3	Serviços específicos para usuários default no Terminal de Acesso.	62
6.4	Inicialização da aplicação Agenda Pervasiva no componente OMAP1.	63
6.5	Inicialização da aplicação Agenda Pervasiva no componente OMAP2.	63
6.6	Inicialização da aplicação Agenda Pervasiva no componente Gerente.	64
6.7	Inicialização da aplicação Agenda Pervasiva no componente Servidor de Localização Proxy.	65
6.8	Inicialização da aplicação Agenda Pervasiva no componente Servidor de Perfil.	65
6.9	Inicialização da aplicação Agenda Pervasiva no componente Terminal de Acesso.	66
6.10	Contribuição do componente Gerente para a realização do serviço de criação do usuário.	67
6.11	Contribuição do componente Servidor de Perfil para a realização do serviço de criação do usuário.	67
6.12	Contribuição do componente Terminal de Acesso para a realização do serviço de criação do usuário.	68

6.13	Funcionamento do componente OMAP1 para a execução da Agenda Pervasiva. .	69
6.14	Funcionamento do componente Gerente para a execução da Agenda Pervasiva. .	70
6.15	Funcionamento do componente Servidor de Localização para a execução da Agenda Pervasiva.	71
6.16	Funcionamento do componente Servidor de Perfil para a execução da Agenda Pervasiva.	71

Lista de Tabelas

2.1	Exemplos de dispositivos para cada tipo de dispositivo	21
3.1	Análise das plataformas relacionadas ao trabalho	34
5.1	Avaliação da Plataforma Pervasiva Proposta.	56
6.1	Exemplo de Funcionamento: Usuário, compromisso e localização.	60

Lista de Símbolos e Abreviaturas

- AMA** - Agente Móvel da Aplicação
- API** - *Application Programming Interface*
- CORBA** - *Common Object Request Broker Architecture*
- DDR-RAM** - *Double Data Rating - Random Access Memory*
- DHCP** - *Dynamic Host Configuration Protocol*
- DSP** - *Digital Signal Processing*
- DVD** - *Digital Video Disc*
- E/S** - Entrada/Saída
- GIOP** - *General Inter ORB Protocol*
- GPIO** - *General Purpose Input/Output*
- GTSH** - *Gator Tech Smart House*
- HD** - *Hard Disk*
- HNS** - *Holo Name System*
- HW** - *Hardware*
- HW/SW** - *Hardware/Software*
- I/O** - *Input/output*
- IP** - *Internet Protocol*
- NFS** - *Network File System*
- OMAP** - *Open Multimedia Application Platform*
- ORB** - *Object Request Broker*
- PC** - *Personal Computer*
- PDA** - *Personal Digital Assistant*
- QoS** - *Quality of Service*

RBA - *Room-Bridge Adapter*

RFID - *Radio Frequency Identification*

ROM - *Read Only Memory*

SO - *Sistema Operacional*

SOC - *System on a Chip*

SW - *Software*

TCP/IP - *Transmission Control Protocol/Internet Protocol*

TFTP - *Trivial File Transfer Protocol*

TV - *Televisão*

UART - *Universal asynchronous receiver/transmitter*

UPnP - *Universal Plug and Play*

USB - *Universal Serial Bus*

XML - *Extensible Markup Language*

YUM - *Yellowdog Updater Modified*

Sumário

Capítulo 1: Introdução	15
1.1 Motivação	16
1.1.1 Cenário 1 - Casa Inteligente	16
1.1.2 Cenário 2 - Novas funcionalidades para automóveis	17
1.2 Objetivos	17
1.3 Organização do Texto	18
Capítulo 2: Conceitos Gerais do Trabalho	19
2.1 Ambientes Pervasivos	19
2.2 Dispositivos Pervasivos	20
2.3 Redes Pervasivas	21
2.4 Arquitetura Pervasiva	22
2.5 Plataforma HW/SW	24
2.6 Requisitos para Computação Pervasiva	25
Capítulo 3: Estado da Arte	27
3.1 Arquiteturas Pervasivas	27
3.1.1 AURA	27
3.1.2 GAIA	28
3.1.3 GTSH	28
3.1.4 Homeros	29
3.1.5 PHOLO	29
3.1.6 Avaliação	29
3.2 Plataformas para Computação Pervasiva	30
3.2.1 Gateways	30
3.2.2 Plataforma para objetos inteligentes	31
3.2.3 Plataforma de interação com o usuário	32
3.2.4 Avaliação	33
Capítulo 4: Definição de uma Nova Arquitetura	36
4.1 Arquitetura Proposta	36
4.2 Camada Física	37
4.3 Camada SO	38
4.4 Middleware	39
4.4.1 Comunicação	39

	14
4.4.2 Gerenciador de Recursos e Serviços	40
4.4.3 Adaptação e Reconfiguração	41
4.4.4 Gerenciador de Segurança	41
4.4.5 Gerenciador de Contexto	42
4.4.6 Monitor	42
4.5 Camada de Aplicação	43
Capítulo 5: Modelagem e Implementação da Plataforma Proposta	45
5.1 Modelagem da Plataforma	45
5.2 Projeto da Plataforma: Modelo em Camadas	46
5.3 Implementação da Plataforma	47
5.3.1 Plataforma HW	47
5.3.2 Sistema Operacional	49
5.3.3 Middleware	50
5.4 Avaliação da Plataforma	56
Capítulo 6: Aplicação Agenda Pervasiva e Avaliação da Plataforma	57
6.1 Aplicação Agenda Pervasiva	57
6.1.1 Descrição da Aplicação	57
6.1.2 Infra-estrutura do Ambiente para a Aplicação	58
6.1.3 Exemplo de Funcionamento da Aplicação	59
6.1.4 Implementação da Aplicação	60
6.2 Resultados Obtidos	60
6.2.1 Serviços Disponíveis no Sistema	61
6.2.2 Experimento 1: Inicialização do ambiente para Agenda Pervasiva	62
6.2.3 Experimento 2: Avaliação do Funcionamento dos Serviços	66
6.2.4 Experimento 3: Avaliação do Funcionamento da aplicação Agenda Pervasiva	69
6.3 A Contribuição da Plataforma para Aplicação e Utilização de outros Recursos	72
6.3.1 Middleware	72
6.3.2 Escalabilidade: adição de novas salas	73
6.3.3 Proxy para tratar a heterogeneidade	73
Capítulo 7: Conclusão e Trabalhos Futuros	74
7.1 Conclusão	74
7.2 Contribuições de Pesquisa	75
7.3 Trabalhos Futuros	76
Referências Bibliográficas	77

Capítulo 1

Introdução

Os avanços tecnológicos na microeletrônica (como vários sistemas em um único chip - contribuindo por minimizar o tamanho e custo dos dispositivos) e na comunicação (como redes sem fio e redes Ad-Hoc [1]) levaram diversos grupos de pesquisa a direcionar seus projetos a um novo cenário computacional. Esse cenário, que contempla aplicações com novas funcionalidades em ambientes inteligentes, é chamado de computação pervasiva.

Computação pervasiva ou computação ubíqua é a proposta de um novo paradigma computacional que disponibiliza ao usuário acesso computacional de modo invisível [2] e onipresente [3]. O conceito de invisibilidade permite que o usuário não precise conhecer a tecnologia para desfrutar de seus benefícios [2]. Onipresença possibilita ao usuário o uso da tecnologia em qualquer lugar e sempre que precisar [3].

Considerada como a Terceira Era da Informática [4], a computação pervasiva surgiu em 1991 com a visão de Weiser [5]. Weiser vislumbrava a idéia que no futuro os computadores habitariam os mais triviais objetos: etiquetas de roupas, canetas, interruptores de luz, etc. No entanto, em 1991, não existia tecnologia suficiente para concretizar a visão de Weiser [2].

Após uma década de progresso na computação (disponibilização de novos dispositivos como PDA, relógios inteligentes e eletrodomésticos para casas inteligentes [6]) e na comunicação (avanços em padrões de redes sem-fio como Bluetooth [7], IEEE 802.11 [8] e em redes inteligentes como Ad-Hoc [1]), a proposta de Weiser vem, pouco a pouco, se materializando.

No entanto, para concretizar a proposta da computação pervasiva, é necessário superar os desafios existentes. Um ambiente pervasivo demanda aplicações inteligentes e sensíveis ao contexto, ou seja, aplicações que detectam mudanças no ambiente e se adaptem a elas da melhor maneira possível. Além disso, um ambiente pervasivo é composto por uma variedade de dispositivos de diferentes capacidades em termos de capacidade de hardware, sistema operacional, e conectividade da rede (com ligações cabeadas ou sem fio). Essa heterogeneidade deve ser tratada a fim de disponibilizar os serviços e recursos da rede, tratando até mesmo problemas de falhas e desconexões. Outro fator importante para a computação pervasiva é possibilitar a integração de dispositivos usados pelo usuário no dia-a-dia (por exemplo, forno microondas, rádios e cafeteiras) em ambientes pervasivos.

Com o objetivo de superar esse último desafio, esse trabalho propõe uma plataforma HW/SW que seja capaz de integrar dispositivos eletrônicos em um ambiente pervasivo.

1.1 Motivação

A computação pervasiva possui muitos aspectos motivacionais para a pesquisa acadêmica e para a indústria. Essa nova área de pesquisa é capaz de unir diversos aspectos motivacionais em uma única situação de nosso cotidiano (cenário). Nessa seção, dois cenários são utilizados para ilustrar o conceito de computação pervasiva.

1.1.1 Cenário 1 - Casa Inteligente

Esse cenário mostra a utilização da computação pervasiva em uma casa inteligente [6]. A casa inteligente apresenta novas funcionalidades e novas tecnologias para permitir ao usuário mais conforto, praticidade e segurança em sua vida pessoal. A Figura 1.1 mostra um projeto de casa inteligente. A seguir, alguns exemplos de funcionalidades adicionadas a essa casa inteligente são apresentados.

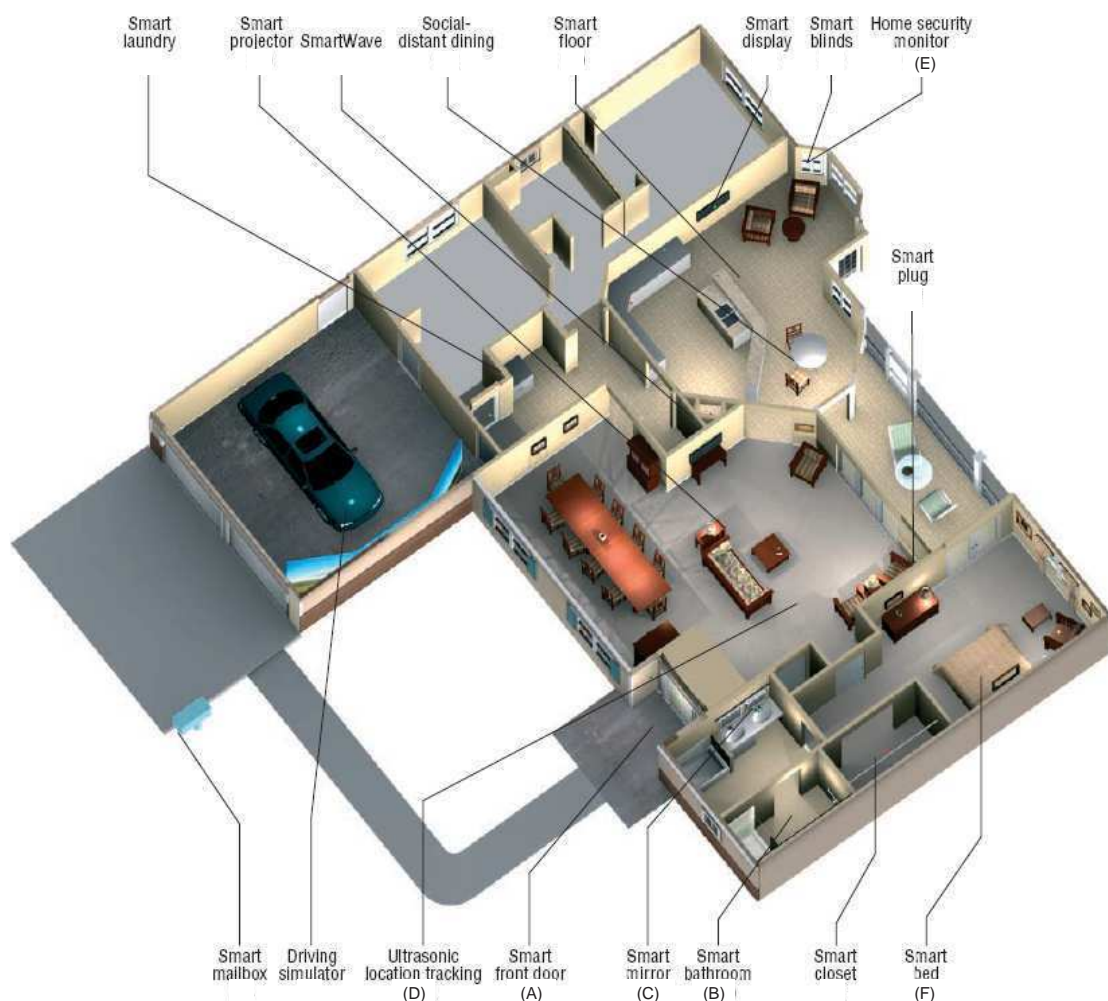


Figura 1.1: Exemplo de uma casa inteligente

Porta da casa inteligente. Essa funcionalidade inclui uma etiqueta de identificação com frequência de rádio para a identificação e autorização dos residentes. Além disso, a porta da casa inteligente inclui microfone, câmera, abertura automática da porta, tranca elétrica e caixa acústica a fim de permitir aos moradores a comunicação com visitantes e a admissão destes.

Banheiro inteligente. Esse banheiro possui sensores capazes de detectar e notificar ao serviço central a falta de sabonetes, papéis higiênicos e outros produtos higiênicos. O chuveiro possui reguladores de temperatura da água para prevenir queimaduras ao usuário e os sanitários contêm descarga automática. Outras tecnologias para medir valores biométricos dos ocupantes, como peso, altura e temperatura, podem ser adicionadas a essa funcionalidade.

Espelho inteligente. Cada espelho inteligente da casa pode permitir a exibição de importantes mensagens e lembretes - por exemplo, um agendamento de medicamentos - aos usuários quando necessários.

Refrigerador inteligente. Um futuro refrigerador executará o monitoramento da disponibilidade e consumo de alimentos, detectará alimentos fora do prazo de validade, criará listas de compras e fornecerá conselhos para preparação de pratos de acordo com os alimentos armazenados no refrigerador.

Monitor de Segurança da Casa. Um sistema de segurança continuamente verificará todas as janelas e portas e, a pedido, informará aos residentes quais janelas e portas estão abertas ou fechadas.

Cama inteligente. A cama inteligente tem equipamentos especiais para monitorar a forma de dormir dos usuários para se adaptar aos desejos do usuário.

1.1.2 Cenário 2 - Novas funcionalidades para automóveis

A utilização da computação pervasiva pode permitir a automóveis novas funcionalidades para atender o usuário. Exemplos de novas funcionalidades para automóveis utilizando a computação pervasiva são apresentados a seguir [9].

Serviços móveis de negócio. Esses serviços permitem ao motorista buscar dados, documentos e relatórios de sua empresa, encomendar produtos através da Internet e conduzir outras transações, ajudando a reduzir perdas econômicas e frustrações associadas ao congestionamento do trânsito.

Serviços baseados em localização. Utilizando a posição corrente do veículo, os serviços baseados em localização podem fornecer aos ocupantes conteúdos sobre o tráfego do trânsito, localização de outros usuários e informações sobre serviços comerciais próximos ao veículo como lanchonetes, borracharias, etc.

Tráfego de dados. Veículos equipados com comunicadores sem fio e sensores especiais podem coletar e transmitir dados ambientais e dados sobre o tráfego em tempo real - por exemplo, informações sobre hábitos do motorista, congestionamento de estradas e nível de poluição.

Autoconserto veicular. Veículos com sofisticadas capacidades de computação e comunicação poderiam atuar com ferramentas para facilitar a manutenção do veículo e prevenir acidentes nas estradas. Veículos também poderiam transmitir mensagens de socorro no caso de acidentes.

1.2 Objetivos

O objetivo desse trabalho é a proposta e a implementação de uma plataforma HW/SW que permita a integração de dispositivos eletrônicos em um ambiente pervasivo. Essa plataforma possui as seguintes funcionalidades no ambiente pervasivo:

- conectar dispositivos eletrônicos usados pelo usuário no dia-a-dia na rede pervasiva;

- buscar e disponibilizar serviços e recursos permitidos, existentes na rede, para dispositivos conectados na plataforma;
- permitir o acesso dos dispositivos conectados na plataforma a serviços e recursos pervasivos existentes no ambiente;
- disponibilizar os recursos permitidos dos dispositivos integrados na plataforma para todos os outros dispositivos existentes na rede;
- permitir a integração de sensores, atuadores e dispositivos de identificação para possibilitar a capacidade de detecção e controle de contexto no ambiente;
- possibilitar a adição de novos serviços e recursos a partir da integração de um ou mais dispositivos integrados na plataforma (configurabilidade).

Com o propósito de definir os serviços e componentes necessários na computação pervasiva para a plataforma, esse trabalho tem como objetivo secundário definir uma arquitetura pervasiva. Essa arquitetura é projetada para executar as aplicações, lidando com os requisitos e desafios existentes na computação pervasiva.

1.3 Organização do Texto

O presente trabalho é composto por sete capítulos. O capítulo 2 descreve os conceitos básicos relacionados à computação pervasiva assim como algumas definições importantes definidas ou adotadas no trabalho. Esse capítulo também salienta os principais requisitos necessários para a computação pervasiva. No capítulo 3, esforços de pesquisa relacionados à arquitetura e à plataforma são relacionados para auxiliar na definição da arquitetura e na elaboração do modelo de plataforma proposto. O capítulo 4 apresenta a arquitetura pervasiva estruturada em camadas. A arquitetura é utilizada para definir os principais serviços e componentes para a plataforma proposta. O capítulo 5 descreve o modelo de plataforma proposto no trabalho. Nesse capítulo, são apresentados o projeto e a implementação de plataforma que segue como premissa esse modelo proposto. No capítulo 6, uma aplicação chamada Agenda Pervasiva é apresentada. Experimentos realizados são mostrados e avaliados, com o objetivo de viabilizar o funcionamento do modelo proposto. E por fim, conclusões e trabalhos futuros são discutidos no capítulo 7.

Capítulo 2

Conceitos Gerais do Trabalho

A computação pervasiva é um novo paradigma que contempla um novo cenário computacional com aplicações com novas funcionalidades em ambientes inteligentes. Essas aplicações são distribuídas e utilizam recursos de uma grande variedade de dispositivos interligados em uma rede altamente heterogênea.

Esse capítulo tem como objetivo definir alguns conceitos relacionados à computação pervasiva assim como apresentar algumas definições do trabalho. Os conceitos relacionados à computação pervasiva são apresentados inicialmente. Em seguida, a definição de arquitetura e plataforma é apresentada. Por fim, esse capítulo descreve os principais requisitos necessários para o desenvolvimento de aplicações na computação pervasiva.

2.1 Ambientes Pervasivos

Ambiente pervasivo é o espaço físico onde a computação atua de forma "discreta" para fornecer onipresença aos usuários. Ou seja, ambiente pervasivo é o lugar onde as aplicações pervasivas interagem com os usuários, de forma invisível e onipresente. Um ambiente é representado por um conjunto entidades tais como dispositivos, usuários, redes e locais relacionados ao contexto computacional da aplicação [10]. Atualmente, a computação pervasiva contempla uma variedade de aplicações atuando em diferentes ambientes pervasivos. São eles:

- **Ambientes Acadêmicos:** tem como objetivo auxiliar os professores na produção de material didático e das aulas, e ajudar os estudantes com as anotações das aulas e com a tele-educação, proporcionando ao estudante comparecer a uma aula mesmo estando distante de sua universidade [11]. Exemplos de aplicações pervasivas dessa área são Classroom 2000 [12] e o Smart Classroom [11].
- **Ambientes Corporativos:** oferece suporte às empresas em seu ambiente corporativo, desde segurança e sigilo até reuniões reais e virtuais de funcionários. Exemplos de projeto nessa área incluem o Kimura [13] e o iRoom (Interactive Room) [14].
- **Ambientes Turísticos:** tem como objetivo auxiliar os usuários como um guia turístico eletrônico. Ou seja, com base na localização do usuário, essa aplicação pode indicar caminhos de determinados lugares, encontrar pessoas e objetos em um ambiente e identificar locais visitados nesse mesmo ambiente. Exemplos de ambientes são museus, shoppings, etc. O Rememberer [15] da Hewlett Packard Laboratories é um exemplo de aplicação nessa área.

- Ambientes Residenciais: fornecem serviços aos habitantes de uma casa a fim de proporcionar conforto, praticidade e segurança a esses habitantes. Exemplos desse tipo de aplicação podem ser vistos no Gator Tech Smart House [6] e no projeto EasyLiving [16].
- Ambientes para Negócios: ajudam os usuários a efetuarem suas compras, proporcionando agilidade, conforto e economia. Essas aplicações registram os produtos ao serem colocados nos carrinhos, informam as promoções e localização dos produtos. O MyGrocer [17] é um exemplo de aplicação para esses ambientes.
- Ambientes Hospitalares: auxiliam médicos e enfermeiros a diagnosticar e medicar seus pacientes, e controlam o tratamento e o estado atual de um paciente. Um exemplo de aplicação para esse tipo de ambiente é o ACDI [18] da Universidade de Aarhus da Dinamarca.
- Outros domínios: outros domínios de aplicações englobam automóveis, laboratórios, aeroportos, penitenciárias, etc. A aplicação Vehicular Mobile Commerce [9] pode ser considerado como uma aplicação no domínio automotivo.

2.2 Dispositivos Pervasivos

A computação pervasiva abrange uma imensa quantidade de diferentes dispositivos. Exemplos de dispositivos incluem controles, sensores e atuadores para residências e automóveis, eletrodomésticos inteligentes, ar-condicionado, aquecedor, relógios e etiquetas inteligentes. Esses dispositivos são elementos fundamentais nas aplicações pervasivas.

Os dispositivos pervasivos normalmente são sistemas embarcados que atuam com diversas finalidades nas aplicações pervasivas. Além de atuar com um propósito específico, um dispositivo pervasivo deve ser capaz de se comunicar com outros dispositivos, a fim de permitir a construção de inteligência no ambiente, que é refletida nas aplicações. Essa inteligência surge através da distribuição de serviços dos dispositivos entre si, onde um dispositivo pode ser capaz de fornecer o serviço de outro dispositivo somente para satisfazer o usuário. Um exemplo disso seria a requisição de um usuário a um serviço existente em seu PDA para executar na televisão de sua sala de estar.

Com a tentativa de visualizar os objetivos específicos dos dispositivos, uma classificação foi elaborada por [19], seguindo quatro tipos de dispositivos: controles inteligentes, dispositivos de acesso à informação, dispositivos fixos de ambientes, dispositivos de entretenimento. A Tabela 2.1 apresenta exemplos de dispositivos seguindo essa classificação.

Os controles inteligentes são caracterizados por serem muito pequenos (tiny) e normalmente se integram a outros dispositivos (câmeras, sensores, atuadores etc) com o objetivo de detectar o contexto do ambiente e tentar manter o controle sobre o mesmo. Os dispositivos de acesso à informação são dispositivos com grande capacidade de mobilidade e comunicação, utilizados para as tarefas diárias e profissionais do usuário.

Os dispositivos fixos de ambiente são dispositivos ligados ou conectados a equipamentos ou produtos eletrônicos. Esses dispositivos são caracterizados por interagirem entre si e com os controles inteligentes visando maior conforto e praticidade dos usuários.

Os dispositivos de entretenimento são dispositivos embarcados que proporcionam lazer e entretenimento ao usuário e são projetados para executar aplicações multimídia.

Tabela 2.1: Exemplos de dispositivos para cada tipo de dispositivo

Tipos de Dispositivos	Controles Inteligentes	Dispositivos de Acesso à Informação	Dispositivos Fixos de Ambiente	Dispositivos de Entretenimento
Exemplos de Dispositivos	Controles Residenciais	Telefones Celulares	Geladeiras Inteligentes	Câmeras Digitais
	Etiquetas Inteligentes	Smartphone	Terminais de Caixa Eletrônico	Games
	Cartões Inteligentes	PDA	Sistemas de Navegação	Televisão
	Termostatos	Laptop	Monitores Médicos	MP3 Players

2.3 Redes Pervasivas

As redes pervasivas são caracterizadas por elevados níveis de heterogeneidade para executar as aplicações com um bom desempenho e por conectar variados tipos de dispositivos. Esses elevados níveis de heterogeneidade implicam em uma rede com diferentes subredes, onde cada subrede possui um protocolo de comunicação diferente de outra subrede.

A Figura 2.1 mostra um exemplo de rede pervasiva atuando em uma aplicação de uma residência inteligente. Nesse exemplo pode ser visto a heterogeneidade das subredes, através das três subredes diferentes: TCP/IP, IEEE1394 [20] e LonWorks [21]. Nas subredes existem diferentes dispositivos que se comunicam através de um protocolo de comunicação. Por exemplo, a impressora se comunica com o HD externo através da rede TCP/IP, e usando o protocolo de comunicação TCP/IP. As subredes se interligam através de roteadores, que permite a comunicação de dois dispositivos em subredes diferentes. Por exemplo, a comunicação entre a câmera de vigilância que se encontra na subrede LonWorks e o PC da subrede TCP/IP é feita pelo roteador. Conseqüentemente, o pacote que tinha informações para o protocolo de comunicação LonTalk foi convertido pelo roteador para o protocolo de comunicação TCP/IP.

Outro aspecto chave de uma rede pervasiva é a conectividade. Para obter os princípios da computação pervasiva, a rede deve fornecer uma conectividade sem fronteiras, satisfazendo o conceito de comunicação de qualquer lugar e a qualquer momento [3] e possibilitar mobilidade dos usuários, dos dispositivos e da aplicação. O problema da conectividade acontece quando o dispositivo necessita desconectar constantemente e indisponibiliza seus serviços por tempo indeterminado, como é o caso de dispositivos móveis.

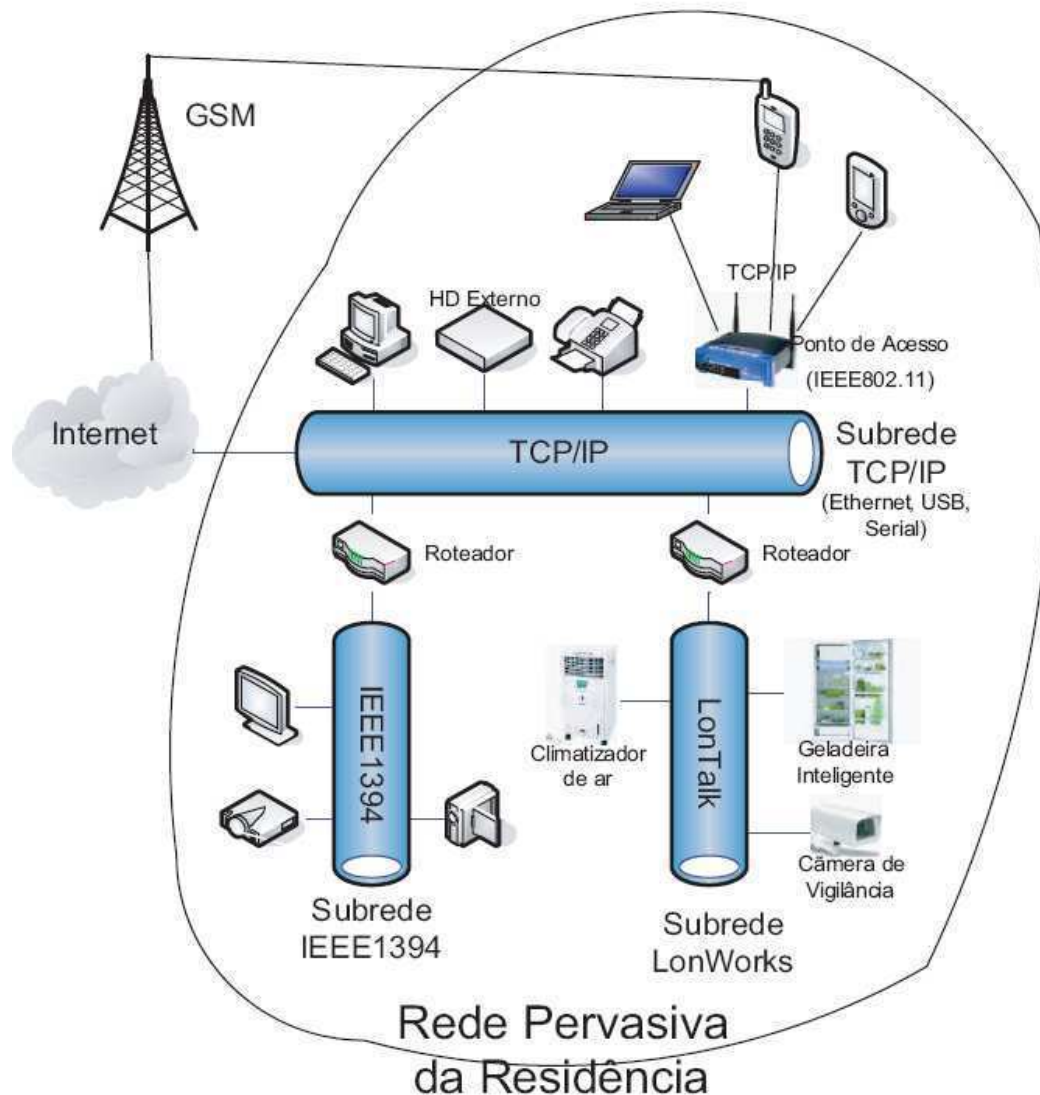


Figura 2.1: Rede atuando em uma residência.

2.4 Arquitetura Pervasiva

O conceito dado para arquitetura nesse trabalho é relacionado a aplicações distribuídas em um ambiente pervasivo, ou seja, é definido pela integração de todos os dispositivos pervasivos existentes no ambiente. Nesse trabalho, arquitetura pervasiva é definida como estrutura e organização do ambiente pervasivo, onde os componentes de hardware e software, localizados nos dispositivos desse ambiente, comunicam-se e compartilham recursos para disponibilizar ao usuário acesso computacional de modo invisível e onipresente.

A Figura 2.2 mostra a estrutura de uma arquitetura pervasiva¹. Essa arquitetura é baseada em uma arquitetura de sistemas distribuídos, definido por Tanenbaum [22]. Um ambiente pervasivo permite a execução de vários dispositivos se comunicando entre si através de uma rede pervasiva. Esse ambiente pode ser organizado em camadas que representa logicamente a execução das aplicações definidas no ambiente (camada de aplicação) e a execução física de um dispositivo (hardware). Na intermediação dessas estão o sistema operacional e o middleware, que respectivamente definem a abstração de toda complexidade do hardware para programação e execução das aplicações e a abstração da heterogeneidade da plataforma (dispositivos e rede) e das dependências do sistema operacional [22].

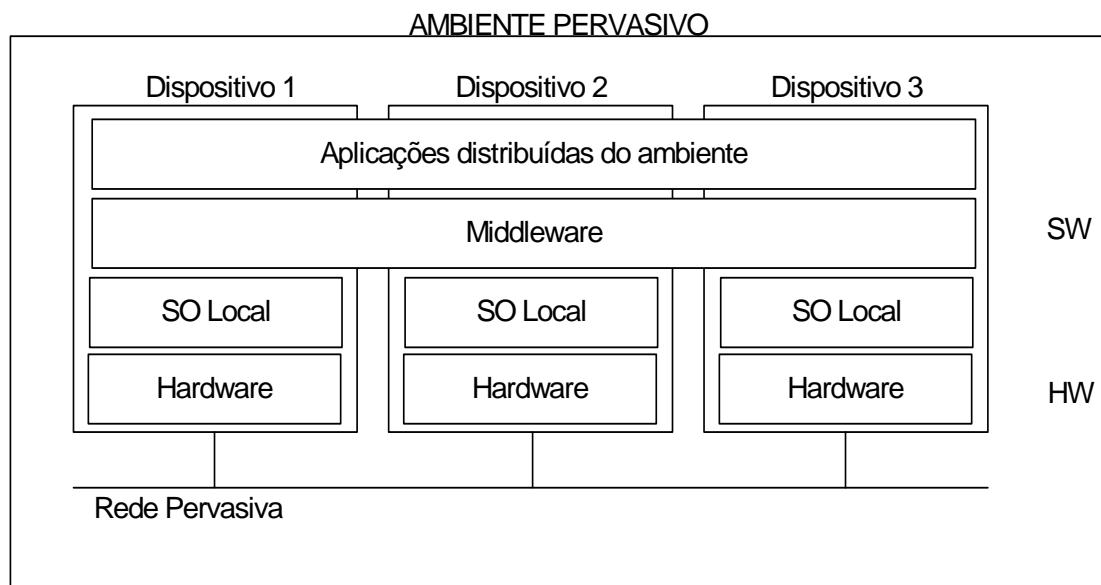


Figura 2.2: Estrutura de uma Arquitetura para um Ambiente Pervasivo.

A arquitetura pervasiva é baseada em sistemas distribuídos, mas essa arquitetura é voltada para atender os dispositivos pervasivos e os componentes (hardware e software) apresentam muitas modificações para permitir a execução das aplicações no ambiente pervasivo.

O hardware na computação pervasiva possui cada vez mais disponibilidade de recursos. Com a tendência de pequenos componentes possuírem grande poder computacional (SOC com diversos processadores, grande capacidade de memória e diversos periféricos e sensores), pequenos dispositivos, ambientes e objetos utilizados no dia-a-dia (como cadeiras, mesas e sofás) tendem a incluir "inteligência" para interagir com os usuários e com outros dispositivos, sem que o usuário perceba. Além disso, o uso cada vez maior de dispositivos com comunicação sem fio torna os dispositivos embarcados mais invisíveis e mais onipresentes para os usuários.

O software na computação pervasiva tende a tornar-se cada vez mais complexo. Com a heterogeneidade dos dispositivos e da rede, e os cuidados com limitação de memória, capacidade de processamento, energia e a latência da rede torna-se mais difícil projetar o software para a comunicação e distribuição de serviços e recursos entre os dispositivos. Entende-se por serviço um conjunto de funcionalidades e tarefas que um dispositivo é capaz de tratar. Recurso é um

¹Figura originada de [22]

componente físico ou virtual disponibilizado para os dispositivos que possuem limitações.

Para facilitar o desenvolvimento da aplicação, tendo em mente essas limitações e dificuldades, o software da computação pervasiva possui limitações e restrições também nas suas três camadas:

Sistema Operacional (SO): tem que ser projetado tendo em mente as limitações do dispositivo e do objetivo de uso do mesmo dispositivo. No entanto, o sistema operacional também deve ser flexível e configurável (para permitir novos tipos de serviços e aplicações) e ter capacidade de comunicação e disponibilização de recursos a outros dispositivos e ao usuário para atuar no ambiente pervasivo.

Middleware: para atender os princípios de invisibilidade e onipresença, os middlewares existentes na computação pervasiva devem fornecer os seguintes serviços: abstração da comunicação, adaptação, reconhecimento de contexto e descoberta de serviços (ou recursos).

Aplicação: devem ser projetadas levando em consideração a minimização da heterogeneidade dos dispositivos e a grande capacidade de adaptação. O primeiro fator é adquirido através da restrição de alguns aspectos durante a fase de projeto e implementação da aplicação pervasiva: limitação do hardware de alguns dispositivos e da comunicação que esses dispositivos podem proporcionar. A escolha da linguagem de programação, dos protocolos e padrões de comunicação e dos outros componentes SW é importante no projeto da aplicação para suportar as restrições existentes. O segundo fator proporciona ao sistema maior confiabilidade (evitar falhas), atualização do ambiente e de novas funcionalidades aos usuários.

2.5 Plataforma HW/SW

O conceito de plataforma HW/SW nesse trabalho é baseado em um dispositivo computacional genérico capaz de permitir o reuso de software. Uma plataforma é definida por um conjunto de componentes (hardware e software) integrados que são projetados para permitir o reuso de aplicações em software. Esse reuso permite a redução de custos de um projeto e fornece flexibilidade para atender um determinado conjunto de aplicações limitadas por determinadas restrições.

Um exemplo de plataforma é um dispositivo computacional utilizado para atender uma família de celulares. Muitos fabricantes produzem diferentes modelos de celulares com a mesma plataforma, modificando apenas a aplicação em software e o invólucro do celular. O reuso da plataforma para aplicações diferentes permite reduzir os custos de projeto para a produção de um novo modelo com a mesma plataforma.

A plataforma desse trabalho é definida para integrar diversos dispositivos em um ambiente pervasivo. A Figura 2.3 (a) mostra a fisionomia dessa plataforma e a Figura 2.3 (b) apresenta a estrutura da plataforma. A estrutura dessa plataforma é dividida em duas camadas: plataforma HW e plataforma SW.

A plataforma HW é constituída de três partes: Unidade de Entrada/Saída, que representa os componentes HW para controlar os dispositivos de entrada (E) e saída (S); Unidade de Processamento, que representa os componentes HW utilizados para processar os componentes SW, como os processadores e memórias; e Unidade de Rede, que representa os componentes HW para efetuar a comunicação com outros dispositivos. A escolha de cada componente HW em cada uma dessas unidades é importante para o projeto da plataforma, pois esses componentes delimitam as limitações do domínio das aplicações. Por exemplo, o desempenho do processador, o tamanho de uma memória ou o consumo de energia de um dispositivo é capaz de restringir o uso de muitas aplicações por essa plataforma.

Por outro lado, a plataforma HW por si não é capaz de obter o nível de reuso da aplicação software que se deseja. Para esse reuso, é necessário abstrair a plataforma a um nível mais alto, onde a aplicação SW tenha fácil acesso para utilizar os componentes HW. Por isso existe

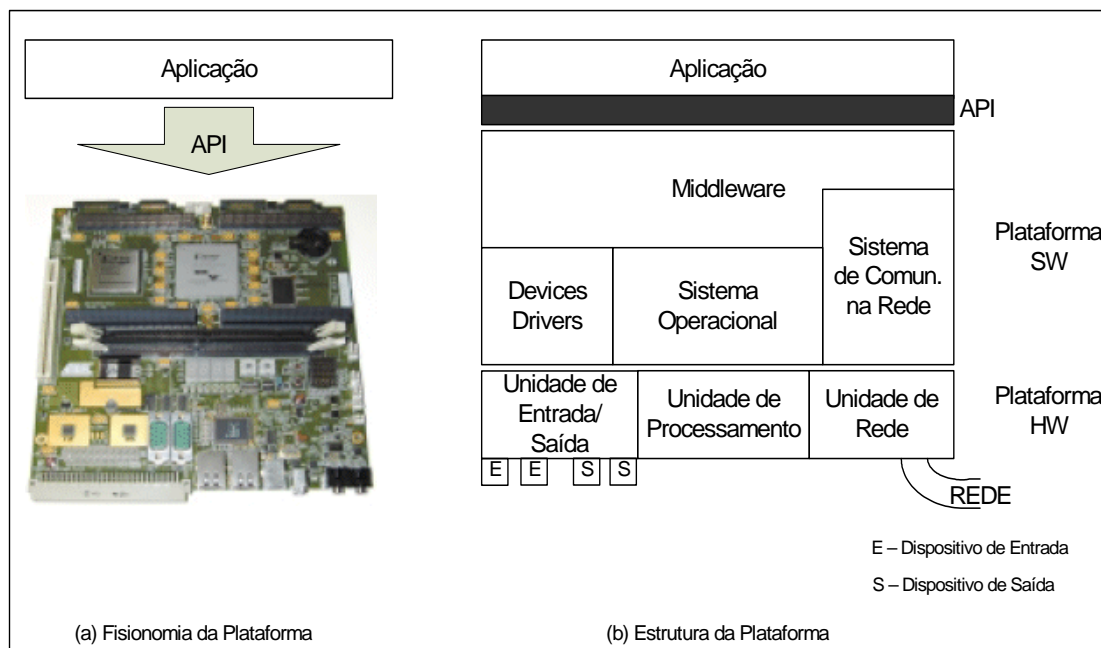


Figura 2.3: Fisionomia e Estrutura de uma Plataforma HW/SW.

uma camada que serve como interface da aplicação para os componentes HW. Essa camada é a plataforma SW e é constituída por componentes que controlam as unidades existentes na plataforma HW e abstraem a comunicação com: sistema operacional para controlar a execução das aplicações na unidade de processamento; os devices drivers para se comunicar com os dispositivos de entrada e saída e controlar a Unidade de Entrada/Saída; e o sistema de comunicação de rede que controla e estabelece os protocolos de comunicação para a Unidade de Rede. Outro componente importante na plataforma SW é o middleware. O middleware tem a tarefa de abstrair o sistema operacional para a aplicação e abstrair a comunicação com outros dispositivos heterogêneos. Para a aplicação fazer uso dos componentes HW e SW, existe uma interface que conecta a aplicação com a plataforma. Essa interface é definida como Interface de Programação da Aplicação (API).

2.6 Requisitos para Computação Pervasiva

A computação pervasiva é baseada em um ambiente computacional distribuído com grande diversificação de dispositivos que permite ao usuário o acesso computacional de modo invisível e onipresente. Com base nessas características, os requisitos definidos nesse trabalho necessários para a computação pervasiva são:

- **Suporte a Heterogeneidade.** A heterogeneidade em sistemas computacionais não desaparecerá no futuro [23]. Com base nessa premissa, o ambiente pervasivo deve permitir a conexão e a comunicação de dispositivos com diferentes plataformas que utilizam diferentes protocolos e meios de comunicação e diferentes sistemas operacionais.

- **Gerenciamento de Recursos.** Recursos e serviços se tornam disponíveis e indisponíveis à medida que o tempo transcorre e usuários se deslocam com seus dispositivos [24]. Um ambiente pervasivo deve ser capaz de buscar e descobrir recursos existentes na rede e fornecê-los aos dispositivos e às aplicações sempre que necessários.
- **Suporte a Conectividade.** Alguns dispositivos pervasivos (como por exemplo PDA), por consistirem de mobilidade, são capazes de conectar e desconectar constantemente da rede pervasiva. O ambiente pervasivo deve ser capaz de permitir essa conexão/desconexão dos dispositivos de acesso à informação sem que acarrete problemas de falhas e no Gerenciamento de Recursos.
- **Suporte à Mobilidade.** Os dispositivos móveis e os usuários se deslocam constantemente no ambiente pervasivo. Por isso, a computação pervasiva deve dar suporte a esse deslocamento e permitir a esses dispositivos e usuários todos os recursos que necessitarem. Além disso, o ambiente pervasivo deve ser capaz de capturar informações relacionadas a esse deslocamento tais como localização, contexto, etc.
- **Suporte a Falhas.** Um ambiente da computação pervasiva deve detectar falhas existentes no sistema. Quando detectada alguma falha, o sistema deve tratar dessa falha com base nas informações contidas no contexto do ambiente.
- **Suporte a Segurança.** Aplicações para sistemas pervasivos exigem mecanismos de segurança mais flexíveis e configuráveis que os tradicionais. O suporte à segurança possui dois requisitos específicos para executar aplicações pervasivas: controlar o acesso de sistemas heterogêneos e controlar o acesso a informações de forma que somente o dispositivo solicitante (autorizado ao acesso a informações) receba as informações requisitadas.
- **Suporte a Adaptação e a Reconfiguração Dinâmica.** Um ambiente pervasivo deve permitir suporte a adaptação para suprir a necessidade de um recurso e trazer maior flexibilidade para o sistema. Esses tipos de serviços abordam questões relativas à alteração da estrutura da aplicação e a inserção ou remoção de componentes (HW e SW) em tempo de execução e durante o processamento [25].
- **Suporte a Consciência de Contexto.** Um sistema pervasivo tem como uma de suas finalidades principais conhecer o estado atual do usuário e do ambiente, e modificar seu comportamento baseado nessas informações. Um contexto do usuário pode conter diversos requisitos consistindo de atributos como localização física, estado fisiológico (temperatura do corpo, por exemplo), estado emocional (fome, sede, sono), histórico pessoal, etc. Com as informações, o ambiente, com essa funcionalidade, deve antecipar as necessidades do usuário, mas sem inferir em momentos importunos.
- **Balanceamento entre Pró-atividade e Transparência.** Um ambiente pervasivo deve ser pró-ativo, ou seja, ele deve ser capaz de identificar a intenção de um usuário, através das informações de contexto, para tratar da invisibilidade. Muitas informações de contexto são baseadas na captura de informações do usuário e de contexto do ambiente. Ao mesmo tempo, o sistema deve ser transparente e permitir o usuário executar suas tarefas sem a distração do ambiente em busca de contexto. Por isso, um ambiente pervasivo deve permitir o balanceamento entre pró-atividade e transparência.
- **Descentralização.** As responsabilidades em um ambiente pervasivo são distribuídas entre os dispositivos pervasivos. Por isso, o sistema não possui um servidor que controla todo ambiente, mas esse controle é feito através de uma colaboração de todos dispositivos compartilhando recursos entre si para execução da aplicação.

Capítulo 3

Estado da Arte

Para elaborar o modelo de plataforma proposto no capítulo 1, foram estudados alguns trabalhos relacionados a arquiteturas pervasivas e a plataformas existentes na computação pervasiva. O objetivo do estudo das arquiteturas pervasivas é avaliar as necessidades que os ambientes pervasivos necessitam, em termos de serviços e recursos para a definição da arquitetura pervasiva apresentada no próximo capítulo desse trabalho. Essa arquitetura é utilizada com o propósito de capturar os serviços e alguns componentes necessários para a plataforma proposta.

Já o estudo das plataformas existentes na computação pervasiva mostra alguns trabalhos realizados para conectar dispositivos e objetos no ambiente pervasivo. Esse estudo ajuda na concepção do modelo de plataforma, avaliando os componentes HW e SW necessários para auxiliar no projeto de aplicações pervasivas.

3.1 Arquiteturas Pervasivas

A arquitetura pervasiva, mostrada no capítulo 4, é originada de um estudo realizado nessa seção. Esse estudo visa avaliar as necessidades das arquiteturas existentes na pesquisa acadêmica e é realizado sobre cinco arquiteturas pervasivas: AURA, GAIA, GTSH, Homeros e Pholo. No final dessa seção, uma breve análise é feita, caracterizando vantagens e desvantagens do uso de cada arquitetura.

3.1.1 AURA

O projeto Aura [26] é caracterizado por fornecer para cada usuário o conceito de invisibilidade de serviços de computação e sua localização. Os dois principais desafios do AURA são: maximizar o uso de recursos disponíveis e concentrar a atenção do usuário. A maior fonte de distração do usuário surge da necessidade de gerenciar seus recursos de computação em cada ambiente novo, e do fato que os recursos podem mudar dinamicamente e freqüentemente em um ambiente particular.

A arquitetura SW de AURA é dividida em cinco camadas. A camada do usuário (camada mais acima) é voltada para a inclusão das aplicações existentes em um determinado cenário em que AURA deverá executar. Essa camada é responsável pelo gerenciador das tarefas, chamado de Prisma, que é responsável por fornecer suporte às tarefas, detectar as atividades do usuário e fornecer os serviços de pró-atividade. A camada abaixo é a camada das tarefas que é composta pelas aplicações, por um componente que executa as tarefas remotas e por outro componente que dá suporte a essa execução.

A camada de serviços (uma camada abaixo) é caracterizada pelo tratamento de adaptação e acesso a informação remota. Nessa camada existem dois componentes: um sistema de gerenciamento de arquivos distribuídos que suporta arquivos remotos (Coda) e outro que fornece serviços de sistema operacional para adaptação e monitoramento de recursos (Odisséia). Na camada abaixo, encontra-se um kernel do sistema operacional para suportar as tarefas de um sistema operacional. E a última camada, apresenta uma rede inteligente com serviços de localização, pró-atividade, contexto e gerenciamento de energia no protocolo.

3.1.2 GAIA

A arquitetura SW de GAIA [10] é destinada para tratar um espaço ativo (Active Space). Espaço Ativo [10] é definido como um espaço físico coordenado por uma infra-estrutura em SW baseado no contexto que realça a mobilidade dos usuários para a interação e configura seu ambiente físico e digital. O objetivo de GAIA é coordenar as entidades em SW e os dispositivos de rede heterogênea contidos em um espaço ativo.

A arquitetura de GAIA é composta por três camadas: aplicações para o espaço ativo, framework de aplicações e o kernel de GAIA. As aplicações de GAIA são baseadas em componentes distribuídos e móveis que requerem suporte a execução e gerenciamento de componentes remotos. A camada framework de aplicações é responsável por mobilidade, adaptação e construção dinâmica de aplicações. Sua funcionalidade permite criar novas aplicações, assim como executá-las no espaço ativo.

E a última camada, o kernel de GAIA, é composta por serviços e sistemas que fornecem sincronização, armazenamento de informações e de contextos existentes, detecção de entidades físicas (pessoas e dispositivos) e digitais (serviços e aplicações) existentes no espaço ativo e adaptação conforme as necessidades do contexto.

3.1.3 GTSH

O Gator Tech Smart House (GTSH) [6] é um projeto voltado para o desenvolvimento de uma casa inteligente. Sua arquitetura é composta por seis camadas. A camada mais abaixo é a camada física e consiste de vários dispositivos e aparelhos elétricos que são utilizados para uma determinada funcionalidade da casa. Nessa camada, temos a televisão, a cafeteira, os sensores/atuadores, etc.

A camada acima é a camada da plataforma de sensores que tem como objetivo converter qualquer informação de um sensor ou atuador da camada física em um serviço em SW que pode ser programado ou composto por outros serviços. A camada de serviços que está um nível acima mantém o funcionamento dos serviços ativados. Existem dois tipos de serviços: serviços básicos que representam o mundo físico e são criados a partir do contexto do ambiente e serviços de composição que representam as aplicações e são criados pelos projetistas do sistema.

Já a camada de conhecimento é responsável por tratar das informações do sistema (conhecimento). Seus componentes são utilizados para registrar e descobrir novos serviços e recursos disponíveis. A camada de gerenciamento de contexto detecta e armazena os contextos do ambiente e deixa os desenvolvedores da aplicação criar e registrar os contextos de interesses. Cada contexto é armazenado em um grafo, onde facilmente pode ser recuperado pela camada de serviço para ativar ou restringir serviços no sistema. A camada mais acima é a camada de aplicação que consiste em um gerenciador de aplicação para habilitar serviços, construir um novo grafo que associa comportamento com contexto, registrar e compor novos serviços, assim como procurar e descobrir serviços. Além disso, essa camada é a responsável por efetuar monitoramento e auxiliar os desenvolvedores avaliar o funcionamento do sistema.

3.1.4 Homeros

O Homeros [27] é uma arquitetura SW voltado para atender sistemas heterogêneos móveis. Tem como objetivos gerenciar efetivamente seus recursos, contexto, localização e outros serviços. A arquitetura foi projetada para flexibilizar no suporte de novos serviços, requisições de usuários e novas aplicações.

A arquitetura do Homeros é composta por três camadas. A camada mais abaixo é composta por um gerenciador de componentes que consiste em gerenciar a reconfiguração de seus componentes e recursos. A camada central é composta por dois componentes: o gerenciador de eventos, cujo objetivo é distribuir os eventos através de um modelo de comunicação baseado em produtor e consumidor para estabilidade e confiabilidade; e o repositório de componentes que armazena informações e serviços existentes na aplicação.

A camada mais acima consiste de cinco serviços para aplicação pervasiva: um gerenciador de contexto, que fornece os serviços pró-ativos utilizando as informações do repositório de componentes; uma adaptação QoS que monitora as informações da rede; um gerenciador de configuração que é o responsável pela configuração de novos componentes; um gerenciador de recursos que permite a descoberta e a utilização de novos recursos e o gerenciador de segurança, que fornece soluções de autenticação e delegação usando agentes móveis.

3.1.5 PHOLO

A arquitetura Pholo [28] é destinada para dar suporte para o ambiente de execução do Holoparadigma. O Holoparadigma [29] é um modelo multiparadigma voltado para abstrair a modelagem de ambientes móveis e permitir representações mais similares ao mundo real. Sua unidade de modelagem, o ente, permite o desenvolvimento de sistemas tanto centralizados como descentralizados e possui grande capacidade de mobilidade.

A arquitetura Pholo é dividida em cinco camadas: camada de rede, sistema operacional, camada de serviços, camada de execução e aplicações.

A camada de execução e a camada de serviços são as responsáveis pelos serviços de computação pervasiva na arquitetura. A primeira camada é responsável pela execução dos programas do Holoparadigma. Essa camada é composta por dois serviços: HoloVM e o HNS. O HoloVM é uma máquina virtual que executa as aplicações, de forma distribuída, com suporte à comunicação entre entes, abstraindo toda a complexidade do desenvolvimento. Já o HNS tem o controle de todos os entes em execução, através de uma estrutura em forma de árvore. O objetivo é fornecer informações dos entes para as HoloVMs, como por exemplo localização de um ente.

A camada de serviços é responsável pelo fornecimento dos seguintes serviços: localização física, que informa a localização e aspectos de mobilidade dos entes; contexto para aquisição de informações sobre o mundo real; mobilidade, que consiste na movimentação de um dispositivo físico no espaço físico, na movimentação de um ente modificando o HNS e na migração de um ente (por exemplo, um serviço move-se de um dispositivo para outro); descoberta de serviços, que permite HoloVMs descobrir servidores HNS e descoberta de serviços desempenhados por entes de uma aplicação.

3.1.6 Avaliação

A arquitetura AURA é voltada à computação centrada no usuário, ou seja, a mobilidade física do usuário e a semântica de que a aplicação o segue é central na arquitetura AURA. Por isso, AURA trata dos requisitos necessários para atender ao usuário. No entanto, AURA fica restrita a mudanças do contexto nos ambientes. A arquitetura GAIA é capaz de atender os problemas

de contexto no ambiente. Ela é dirigida somente ao espaço ativo, ficando restrita somente às aplicações ligadas nesse espaço ativo e descartando o deslocamento global do usuário.

GTSH é voltada à computação centrada a situação, que integra o usuário e o ambiente. Ela se delimita aos dispositivos pela detecção do contexto do usuário e do ambiente. No entanto, fica restrita a mobilidade dos dispositivos e da aplicação, delimitando a aplicação somente ao ambiente em que se encontra. Homeros é uma arquitetura voltada na execução de seus serviços e possui uma mobilidade maior. No entanto, é uma arquitetura centralizada e as aplicações ficam restritas a um modelo cliente-servidor.

A arquitetura Pholo é voltada para o suporte à execução do modelo Holoparadigma. Essa arquitetura, através de seu modelo abstrato, disponibiliza a descentralização, em termos de serviços, e alta mobilidade do ambiente. No entanto, a arquitetura está em um estágio ainda imaturo (com relação a alguns serviços para a computação pervasiva) e por isso ainda não pode ser avaliado o tratamento de contexto e falhas. Além disso, o modelo ainda é centralizado para o controle e informações do ambiente (HNS).

3.2 Plataformas para Computação Pervasiva

Quando analisamos o mercado da computação pervasiva, encontramos diferentes tipos de plataformas dedicadas para uma ou mais funcionalidades. As plataformas podem ser voltadas para atender a disponibilização de serviços e recursos dos dispositivos entre diferentes redes, atender redes de sensores, para integrar dispositivos e objetos (artefatos [30]) a um ambiente pervasivo, ou até mesmo para auxiliar na construção de outras plataformas pervasivas.

Por isso nessa seção, estudamos diferentes tipos de plataformas para computação pervasiva. Com a finalidade de facilitar a visão desse trabalho, elaborou-se uma classificação para diferenciar as específicas funcionalidades das plataformas. Em cada subseção, veremos um tipo de plataforma, as suas funcionalidades para a computação pervasiva e especialmente alguns trabalhos relacionados a cada plataforma. No final da seção, uma breve análise é elaborada para avaliar as plataformas estudadas.

3.2.1 Gateways

O conceito relacionado a Gateway na computação pervasiva é um pouco diferente ao aplicado em redes de computadores. Na área de redes de computadores, um Gateway é utilizado apenas para interligar os dispositivos de um ambiente, estabelecendo a conexão e fazendo a conversão necessária (no caso de redes diferentes), tanto em termos de hardware quanto em software [31]. Na computação pervasiva, o Gateway possui serviços adicionais: descoberta de serviços para gerenciar a integração de novos dispositivos na rede e reconfiguração dinâmica do software para fornecer a instalação e atualização de módulos em software no gateway dinamicamente [32].

A plataforma estudada no trabalho relacionada ao Gateway é voltada para atender residências inteligentes e é chamada de RBA (*Room-Bridge Adapter*) [33].

O RBA é uma plataforma utilizada para conectar subredes pervasivas em residências inteligentes. A idéia no trabalho é que cada subdivisão de uma casa - cada peça da casa, tal como sala, quarto, banheiro, etc - seja representada por uma subrede, na qual é controlada por um gateway (RBA).

O RBA é responsável por fornecer os serviços adicionais de um gateway para os dispositivos de uma subdivisão da casa e é capaz de suportar diferentes tipos de redes locais - tais como TCP/IP, IEEE1394, LonTalk e outros - para a conexão dos dispositivos. Os RBAs são interligados através de uma rede principal homogênea (backbone) e um dos RBAs tem a função de gateway pervasivo

para essa rede. Além disso, esse RBA é o responsável pela comunicação externa à casa e por fornecer conexão à Internet.

Para fornecer essas funcionalidades, o software do RBA é composto por quatro camadas (aplicação, middleware core, middleware base e middleware device driver). Na camada device drivers estão os drivers instalados para os protocolos de comunicação relacionados à rede principal e à subrede do RBA. O middleware foi implementado utilizando CORBA [34]. Na camada middleware base, foi utilizado o GIOP (General Inter-ORB - funcionalidade do CORBA - para abstrair os detalhes de cada protocolo e permitir a interoperabilidade da comunicação. A camada middleware core é baseado em eventos e contém os seguintes serviços: registro e gerenciamento dos dispositivos e seus recursos (gerenciador do repositório); controle e comunicação dos dispositivos físicos (Gerenciador de Dispositivo); tratamento de eventos e comunicação entre os componentes SW do RBA (Serviço de Eventos); monitoramento do funcionamento dos dispositivos (Serviço de Gerenciamento). Por fim, a camada de aplicação que é baseada em agentes e é a responsável pelo desenvolvimento e controle dos serviços da residência inteligente (por exemplo agente para controlar a temperatura de uma subdivisão da casa).

3.2.2 Plataforma para objetos inteligentes

Os avanços na última década na microeletrônica e a comunicação permitiram o desenvolvimento de plataformas com baixo custo, menores e com maior interatividade. Com plataformas cada vez menores, a utilização de computação em objetos do dia-a-dia tem sido cada vez mais frequentes. A utilização dessas plataformas em objetos permite ambientes cada vez invisíveis e pervasivos. Esses objetos integrados a essa plataforma são chamados de objetos inteligentes.

Uma plataforma para objetos inteligentes normalmente é uma plataforma pequena e simples, com poucos recursos computacionais e baixo consumo de energia, voltada para o tratamento de apenas uma aplicação específica.

As plataformas para objetos inteligentes devem ser providas de interfaces físicas I/O para permitir interatividade com o usuário ou com o ambiente. Além disso, essas plataformas são constituídas de comunicação que permite a conectividade com outras plataformas do ambiente e de um software específico apenas para tratar as funcionalidades da aplicação.

No presente trabalho, realizou-se o estudo de três plataformas para objetos inteligentes: Smart-Its [30], BTNodes [35] e Motes [36].

Smart-Its [30] é uma plataforma em hardware voltada para permitir a customização de sensores, percepção e sensibilidade ao contexto, assim como saídas físicas para tipos específicos de objetos. Isso requer controle, configuração e integração de diferentes tipos de sensores. A comunicação da Smart-It é responsável por permitir objetos a conectar a outras plataformas para tornar-se um componente do ambiente. O software do Smart-It é composto por três bibliotecas, uma para tratar sensores e atuadores, uma para comunicação e outra para executar as tarefas específicas.

BTnode [35] é uma plataforma hardware/software com a finalidade de servir como protótipo de sistemas de redes de sensores em ambientes inteligentes. Redes de sensores é um conjunto de plataformas com sensores integrados com capacidade de comunicação e sensoriamento para permitir "experimentos" [35] de aplicações ou para verificar o contexto no ambiente [6].

A grande diferença da plataforma BTnode das outras plataformas para objetos inteligentes é que BTnode utiliza bluetooth como módulo de comunicação. Isso permite algumas desvantagens, como por exemplo, o aumento do custo da plataforma e maior consumo de energia. No entanto, a utilização do bluetooth como módulo de comunicação permite inúmeras vantagens: vantagens próprias das características do bluetooth (como comunicação orientada a conexão, maior interoperabilidade entre diferentes dispositivos, interface de desenvolvimento padronizada, segurança,

serviços de descoberta, alta largura de banda, abstração para comunicação, etc); fácil comunicação com dispositivos pervasivos (tais como celulares e PDAs para interação com usuários) e servidores (que permitem monitoramento da aplicação e armazenamento de informações); e servir como atuadores (notificar um usuário através de dispositivos bluetooth ou até agir com outro dispositivo como tirar foto se comunicando com uma câmera wireless).

Motes [36], assim como BTnode, é uma plataforma projetada para aplicações relacionadas a redes de sensores. A plataforma é construída sobre um processador com baixo consumo de energia e integra o padrão de comunicação Zigbee [37]. Possuem facilidades na integração com sensores e atuadores, e entre si (4 conectores de entrada/saída que permite uma plataforma se encaixar na outra sobre a forma de pilha). Além disso, a plataforma inclui diversas opções de entrada/saída: USB, 2 UART, áudio, GPIO, interface para câmera, entre outros. A plataforma foi projetada para permitir grande capacidade de processamento (alta velocidade de processamento e alta capacidade de armazenamento) e baixo consumo de energia (três modos de operação). Os componentes SW da plataforma Motes é composta por um sistema operacional específico para redes de sensores: TinyOS [38]. O Motes, com todos esses componentes, foi projetado para atender aplicações com grande complexidade, relacionadas a redes de sensores, para uso comercial: como em controles industriais e aplicações militares.

3.2.3 Plataforma de interação com o usuário

Nas pesquisas recentes, relacionadas ao estudo da interação homem máquina, muitos dispositivos diferentes relacionados à interface física com o usuário foram projetados (como é o caso do iDog, iLight e o IStylus [39]). Já outros dispositivos foram aperfeiçoados (como é o caso do headphone de AURA no vídeo de demonstração [40]). Esses dispositivos são relacionados à interação com o ambiente e às vezes são os grandes responsáveis por fornecer a invisibilidade necessária ao usuário: seja pela facilidade de uso ou seja por esconder a complexidade do usuário.

Nessa seção, as plataformas que integram esses tipos de dispositivos são estudadas. Essas plataformas diferenciam-se das plataformas da seção anterior porque possuem elevada interatividade com o usuário e muita comunicação com o ambiente para repassar dados fornecidos pelo usuário. Esses dados podem ser diretamente transferidos para o ambiente ou podem ser processados pela plataforma antes da transferência. As plataformas pesquisadas nessa seção são: Calder Toolkit e Phidgets.

Calder Toolkit [41] é uma plataforma de desenvolvimento para explorar e prototipar dispositivos físicos funcionais. Essa plataforma, como pode ser vista na Figura 3.1, é projetada para facilitar na criação de interfaces do usuário no domínio de interfaces físicas ¹. Elas fornecem um conjunto de componentes de entradas e saídas reutilizáveis (5-b) que permite diferentes formas para construir, junto com o restante (processamento e comunicação), a infra-estrutura da plataforma. Essa infra-estrutura é encaixada em blocos de isopores (5-a) para formar o dispositivo interativo (5-c). Para aumentar a configurabilidade da plataforma, existem dois tipos de comunicação para a mesma: com fio (cabeadas) e sem fio (wireless).

Além da alta reconfigurabilidade e flexibilidade que essa plataforma desempenha, ela possibilita diferentes formas de distribuição do processamento: compartilhada, mestre local e mestre global. Na primeira forma, os recursos computacionais e de energia são distribuídos entre todos componentes; na segunda forma, a fonte de energia e os recursos computacionais são fornecidos pouco a pouco por uma plataforma pervasiva central; na última forma, uma fonte externa (ex. PC) fornece os recursos computacionais e de energia em alta densidade.

Phidget [42] são plataformas que abstrai as informações repassadas na entrada do dispositivo interativo e as empacota enviando a um dispositivo central que executa o tratamento adequado

¹A Figura 3.1 da plataforma Calder Toolkit foi retirada de [41]

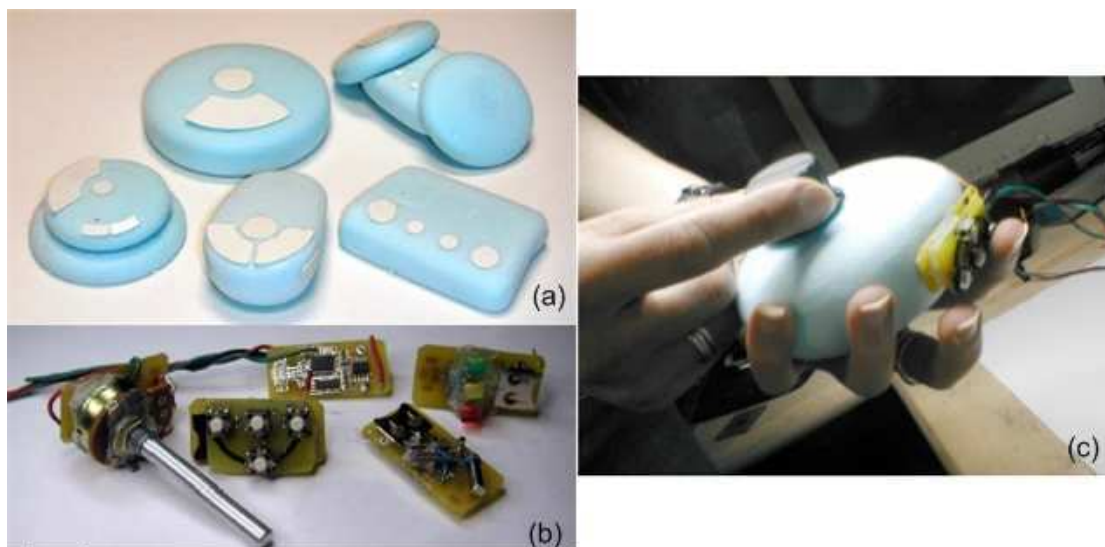


Figura 3.1: Formação de um dispositivo interativo: (a) Blocos de isopor para formar a infraestrutura do dispositivo; (b) Plataformas Calder Toolkit; (c) Dispositivo interativo.

para aquelas informações. O processo de saída funciona de maneira inversa: o dispositivo central repassa as informações empacotadas para o dispositivo interativo, no qual desempacota e envia para a saída adequada. O funcionamento desse sistema abstrai todo processo na plataforma de interação e deixa o processamento pesado para um dispositivo mestre. Por isso, o hardware da plataforma é composto apenas por três simples partes: componentes de entrada e saída para interação com o usuário; microcontrolador responsável por empacotar/desempacotar as informações; e o padrão de comunicação cabeado USB para interagir com o dispositivo central.

Phidget [42] são plataformas que abstrai as informações repassadas na entrada do dispositivo interativo e as empacota enviando a um dispositivo central que executa o tratamento adequado para aquelas informações. O processo de saída funciona de maneira inversa: o dispositivo central repassa as informações empacotadas para o dispositivo interativo, no qual desempacota e envia para a saída adequada. O funcionamento desse sistema abstrai todo processo na plataforma de interação e deixa o processamento pesado para um dispositivo mestre. Por isso, o hardware da plataforma é composto apenas por três simples partes: componentes de entrada e saída para interação com o usuário; microcontrolador responsável por empacotar/desempacotar as informações; e o padrão de comunicação cabeado USB para interagir com o dispositivo central.

3.2.4 Avaliação

Nessa seção foram mostrados alguns tipos de plataformas pervasivas encontradas na pesquisa acadêmica. Cada tipo de plataforma foi dedicado para uma ou mais finalidades específicas. Esse estudo é importante para modelar a plataforma proposta, pois permite avaliar serviços, componentes e restrições das plataformas existentes no mercado.

Para formalizar esse estudo, mencionamos na Tabela 3.1 algumas comparações realizadas entre os tipos de plataformas vistos anteriormente. As comparações realizadas foram relacionadas ao custo e à estrutura da plataforma assim como relacionados às restrições e aos componentes

para o projeto da plataforma proposta para a computação pervasiva.

Tabela 3.1: Análise das plataformas relacionadas ao trabalho

Propriedades	Gateway	Objetos Inteligentes	Interação com Usuário
Custo	Alto	Baixo	Médio - Alto
Tamanho	15 a 60 cm	1 a 8 cm	Variável
Consumo de Energia	Alto	Baixo	Baixo - Médio
Interatividade	Alta	Baixa - Média	Média - Alta
Invisibilidade	Baixa	Alta	Alta (íssima)
Mobilidade	Não	Sim	Sim
Gerenciamento de recursos	Sim	Não	Não
Escalabilidade	Média e difícil	Alta e fácil	Baixa e difícil
Configurabilidade	Sim	Não	Variável
Segurança	Alta	Baixa	Média
Suporte a Falhas	Sim	Não	Não
Suporte a Monitoramento	Médio - Alto	Baixo	Médio
Deteção de Contexto	Não	Sim	Sim
Tratamento de Contexto	Não	Não	Não

O gateway pervasivo é uma plataforma com o objetivo de interligar os dispositivos da rede, fornecendo-lhes acesso a outros serviços, segurança, suporte a falhas e conexão à Internet. Por ser utilizado apenas para interligar os dispositivos de um ambiente e ser fixo ao mesmo (normalmente), não há preocupação quanto à estrutura da plataforma (custo, tamanho e consumo de energia), mas sim com sua interatividade com outras plataformas (homogêneas e heterogêneas). Normalmente, os gateways pervasivos são dotados de serviços de atualização e de monitoramento do sistema, como é o caso do HomeGateway [32]. Não possui o serviço de tratamento de contexto, deixando a necessidade de um servidor para esse tipo de serviço. A escalabilidade foi definida como: média, pois não há a necessidade de grande quantidade de gateways no mesmo sistema; e difícil, porque a adição de um novo gateway no software da arquitetura pode ser de alta complexidade.

A plataforma destinada a objetos inteligentes é uma plataforma customizada, pequena, invisível, com baixo custo e baixo consumo de energia. No entanto, algumas plataformas estão aumentando o custo e o consumo de energia com a adição de meios de comunicação mais sofisticados, como é o caso do BTnode com bluetooth. Essa adição também afeta outros campos. Por exemplo, essa plataforma possui uma variável interatividade com outras diferentes plataformas - no caso BTnode com bluetooth, possui uma interatividade maior que Smart-It.

As plataformas para objetos inteligentes freqüentemente são destinadas para redes de sensores sem fio. Por isso, são altamente escaláveis, possuem mobilidade e deteção de contexto. Não possuem serviços de gerenciamento de recursos, de contexto, de segurança, de falhas ou de monitoramento, a menos que sejam projetados para plataforma alguns serviços adicionais.

As plataformas que interagem com usuários possuem grandes diferenças de umas com as outras. São dependentes da fisionomia e da especialidade que são projetadas e por isso, variam de custo, de tamanho, de consumo de energia e da interatividade com o sistema. São destinadas a fornecer invisibilidade ao usuário e detectar o contexto do ambiente. Sua escalabilidade é baixa e difícil, pois para adicionar uma diferente plataforma interativa, é necessária a construção de novos drivers para tratar sua especificação. Possui um variável grau de configurabilidade (no caso do Calder Toolkit alto e dos Phidgets baixo). Não é uma plataforma voltada para fornecer os serviços de computação pervasiva tais como gerenciamento de recursos, tratamento de falhas e detecção de contexto. E por fim, pode ser monitorado, se for adicionados recursos necessários para esse serviço.

A plataforma proposta nesse trabalho tem um objetivo semelhante ao gateway: integra os dispositivos e fornece os serviços e recursos do ambiente pervasivo. No entanto, o tratamento de contexto, um serviço importante na computação pervasiva, não é implementado pelo gateway.

Outra questão importante, comparando a plataforma proposta e o gateway, não encontrado na tabela, é a distribuição de serviços e recursos no ambiente. Um ambiente pervasivo com gateway é normalmente centralizado, onde um gateway principal trata das questões de segurança, do gerenciamento de serviços e recursos e do tratamento de falhas. Um ambiente pervasivo com a plataforma proposta nesse trabalho é totalmente distribuído, onde cada plataforma é responsável por um conjunto de dispositivos e por fornecer seus serviços à rede. Isso permite que a plataforma seja facilmente integrada a outros tipos de rede, como é o caso de redes ad-hoc.

As outras plataformas foram importantes para definir a interatividade com o usuário e com o ambiente. Uma plataforma deve fornecer suporte a interatividade e a invisibilidade dos dispositivos com o ambiente e com o usuário, sejam feitas através de sensores ou por dispositivos interativos. Por isso, uma outra preocupação da plataforma proposta é permitir a fácil integração com sensores, com interfaces físicas I/O e até com outras plataformas, fornecendo suporte a serviços e de recursos computacionais.

Capítulo 4

Definição de uma Nova Arquitetura

A definição da arquitetura é importante para a captura dos serviços e componentes necessários que a plataforma deve conter. O capítulo anterior permitiu a análise dos requisitos e estudos sobre algumas arquiteturas que podem auxiliar na concepção da arquitetura. Esse capítulo consiste na proposta de uma nova arquitetura com suporte a um ambiente pervasivo, auxiliando na implementação da plataforma alvo desse trabalho.

Para facilitar a visualização da proposta, essa seção foi dividida da seguinte maneira: a primeira seção descreve a arquitetura proposta através de uma estrutura em camadas; as quatro seções seguintes representam as quatro camadas desse modelo e mostram a importância de todos os componentes encontrados em cada camada.

4.1 Arquitetura Proposta

Algumas arquiteturas revisadas na seção anterior são voltadas para o ambiente em que ocorre a aplicação e delimitam-se somente a restrições aplicadas ao ambiente e aos usuários específicos que interagem naquela aplicação. Por isso, essas arquiteturas, são restritas na situação do contexto e na funcionalidade do dispositivo que interage na aplicação.

A proposta desse trabalho focaliza-se no ambiente, nos seus dispositivos e nas contribuições que estes podem proporcionar para o ambiente, o usuário e a situação na aplicação pervasiva. Seguindo o fato que os dispositivos são voltados para uma determinada funcionalidade, sendo essa genérica (ex. PDAs) ou específica (ex. sensor de temperatura), o foco do trabalho fica centrado no tratamento dos dispositivos para a situação da aplicação, do ambiente pervasivo onde ocorre a situação e das necessidades dos usuários que se encontram na situação.

O conceito de foco nos dispositivos é similar ao modelado nas arquiteturas Homeros e GTSH. No entanto, a arquitetura desse trabalho diferencia-se por tratar de diferentes tipos de ambientes em que os dispositivos atuarão (sensores, PDAs, e outros atuando em hospitais, ambientes de trabalho, casas inteligentes e outros), enquanto a arquitetura de GTSH atua somente em casas inteligentes.

Por isso, a arquitetura definida deve ser flexível, tanto ao tratamento do ambiente e do usuário, quanto das funcionalidades dos dispositivos e da situação que ocorre no ambiente. Ou seja, a arquitetura dos dispositivos deve se adaptar a diferentes tipos de ambientes e aplicações pervasivas, buscando interagir com estes e tratar da situação e das necessidades do usuário.

A Figura 4.1 apresenta a arquitetura definida neste trabalho baseado em uma estruturação em camadas. A arquitetura é dividida em quatro camadas: Camada Física, Camada SO, Middleware

e a Camada Aplicação. As camadas relacionadas e os componentes e funcionalidades existentes em cada camada são definidas nas seções seguintes.



Figura 4.1: Arquitetura do Trabalho Estruturada em Camadas.

4.2 Camada Física

A camada física visa o tratamento do meio físico e é composto pelos dispositivos físicos existentes no ambiente pervasivo e pelos meios de comunicação que interligam os mesmos. Esses dispositivos físicos ligados a esses meios de comunicação formam a infra-estrutura HW da arquitetura.

A infra-estrutura HW foi definida baseada em modelos com servidores locais. Esse modelo consiste em um ambiente particionado em subdivisões, onde cada subdivisão é composta por uma plataforma pervasiva. Essa plataforma possibilita a conexão e o controle de todos os dispositivos encontrados nessa subdivisão e fornece a esses dispositivos os serviços necessários para tratar da computação pervasiva: gerenciamento dos serviços, comunicação com outros dispositivos, segurança, controle de falhas, gerenciamento do seu contexto e monitoramento.

Os dispositivos podem ser conectados a essa plataforma diretamente (ponto-a-ponto) ou indiretamente (por uma rede) através de diferentes meios físicos de comunicação, possibilitando a heterogeneidade na comunicação. Por outro lado, as plataformas pervasivas (de cada subdivisão) se comunicam entre si através de uma rede homogênea. Essa comunicação permite a conexão e a distribuição de recursos de dispositivos entre si em diferentes subdivisões. Além disso, a rede homogênea pode possibilitar ainda o acesso à Internet para todas as plataformas pervasivas através de um gateway.

Outros componentes importantes para essa infra-estrutura HW são os servidores. Os servidores podem auxiliar as plataformas em prover os serviços pervasivos ou então serem utilizados pela aplicação. Esses servidores podem ser integrados na rede homogênea para auxiliar nos serviços das plataformas em conjunto (fornecer o tratamento de contexto) ou podem ser integrados em uma das plataformas.

A Figura 4.2 mostra um exemplo da infra-estrutura HW baseado no modelo servidor local aplicada a um ambiente de residência inteligente. Essa casa inteligente possui quatro subdivisões definidas como Sala1, Sala2, Suíte e Cozinha. Em cada subdivisão existem diversos dispositivos conectados a uma plataforma através de diferentes meios de conexão. Na Sala1, por exemplo, existem alguns objetos inteligentes conectados à plataforma através da rede Bluetooth, dois dispositivos fixos de ambiente (TV e DVD player) conectado à plataforma através da rede IEEE1394 e um PC conectado diretamente utilizando TCP/IP.

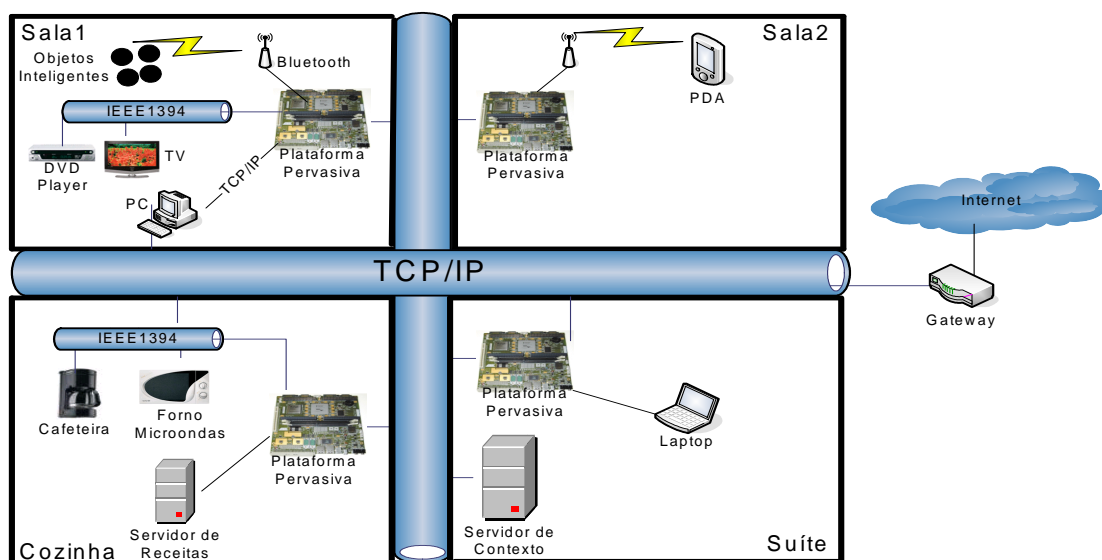


Figura 4.2: Infra-estrutura HW para Residência Inteligente.

As plataformas de cada subdivisão são conectadas a uma rede principal (no caso TCP/IP). Essa rede permite o acesso de recursos de um dispositivo de uma subseção a um outro dispositivo de outra seção. Isso permitiria, por exemplo, o PDA da Sala2 solicitar um recurso da cafeteira inteligente da Cozinha - como por exemplo fazer café. Os responsáveis por essa solicitação seriam as plataformas da Sala2 e da Cozinha.

Os dispositivos podem acessar à Internet através de um gateway. Por exemplo, o forno microondas pode acessar à Internet para buscar uma nova receita. Além disso, esse gateway permite o acesso externo a cada plataforma. Por exemplo, um usuário residente da casa pode solicitar a plataforma encontrada na Suíte um arquivo do laptop.

Já os servidores podem ser conectados à rede principal, como é o caso do servidor de contexto, para auxiliar a plataforma no serviço de tratamento de contexto. Os servidores também são capazes de disponibilizar recursos e auxiliar na aplicação em subdivisões - como é o caso do servidor de armazenamento de receitas da Cozinha.

4.3 Camada SO

A proliferação de inúmeros dispositivos pervasivos diferentes reacendeu o mercado para novos sistemas operacionais personalizados (ou específicos) para estes dispositivos [19]. Os sistemas operacionais são responsáveis por gerenciar os recursos (computacionais, energia e entrada/saída)

do dispositivo e por oferecer aos programadores e usuários, uma abstração que esconde a complexidade do hardware do dispositivo.

Uma arquitetura pervasiva consegue envolver uma variedade de diferentes sistemas operacionais. Cada sistema operacional pode ser utilizado para uma ou mais finalidades específicas ou genéricas. Por isso, esses sistemas operacionais podem ser classificados quanto ao seu objetivo de uso em sistemas operacionais de propósito geral, sistemas operacionais embarcados genéricos e sistemas operacionais embarcados dedicados [43]. Os sistemas operacionais de propósito geral são sistemas que não tem restrições nem limitações de seus dispositivos. Caracterizam-se por possuir inúmeros recursos como sistemas de arquivos, memória virtual e são voltados para aplicações maiores como aplicações de servidores. Exemplos desse tipo de sistema operacional são Windows 2000 e Linux (encontrados em [43]).

Os sistemas operacionais embarcados genéricos são sistemas que possuem determinadas restrições, como limitação de memória e bateria, mas são voltadas para aplicação de propósito geral. Normalmente são utilizados em sistemas embarcados com bastante recurso computacional, tais como PDA, telefones celulares, etc. Alguns exemplos mais usados desse tipo de sistemas operacionais são: Windows CE, uCLinux e Symbian (todos vistos em [43]).

Os sistemas operacionais embarcados dedicados caracterizam-se por serem específicos tanto pra executar em uma determinada arquitetura HW quanto para a execução de uma determinada aplicação. Por exemplo, o TinyOS [38] e o ACOS [44] são sistemas operacionais dedicados respectivamente para redes de sensores e para *smart cards*.

O número de sistemas operacionais para dispositivos pervasivos é cada vez maior e cada vez mais diversificado, devido à heterogeneidade dos mesmos dispositivos. A integração de diversos sistemas operacionais no ambiente pervasivo é necessária, assim como o suporte para a comunicação entre diferentes sistemas operacionais também. Nesse caminho, o grupo OMG [45] vem ampliando sua visão e diversificando alguns projetos para diferentes campos, como por exemplo, RT-CORBA e CORBA embarcado [?].

4.4 Middleware

A construção do middleware para a proposta do trabalho é feita com base nos requisitos descritos na seção 2.6 e nas arquiteturas avaliadas na seção 3.1. A proposta do middleware é importante para descrever os principais componentes para tornar a plataforma acessível aos ambientes pervasivos.

Os principais componentes do middleware são: Comunicação, Gerenciador de Recursos e Serviços, Adaptação e Reconfiguração, Gerenciador de Segurança, Gerenciador de Contexto e Monitor. Para facilitar a visão sobre cada componente, alguns exemplos de cenários são apresentados com base na infra-estrutura HW apresentada na Figura 4.2.

4.4.1 Comunicação

O componente comunicação tem como objetivo tratar da comunicação e permitir a distribuição de serviços e recursos entre os dispositivos na rede pervasiva, abstraindo a heterogeneidade dos dispositivos, da rede e dos componentes em SW que executam nos dispositivos. A comunicação pode ocorrer de duas formas: direta, se o outro dispositivo estiver na mesma subrede, e indireta, se estiver em outra subrede. No caso da comunicação direta, um dispositivo troca informações diretamente com outro dispositivo. Caso ocorra uma comunicação indireta, o dispositivo origem se comunica utilizando a plataforma pervasiva e este se encarrega por todo trabalho de abstração da comunicação.

Por exemplo, na Figura 4.2, se a TV e o DVD Player necessitarem de comunicação entre si, então ambos precisam apenas estabelecer uma conexão e iniciar a troca de mensagens (exceto se tiverem diferentes sistemas operacionais que não estabeleçam conexões; nesse caso o uso da plataforma seria necessário). No caso de uma comunicação entre a TV e o PC ou então entre a TV e a cafeteira, a comunicação não poderia ser direta, devido à heterogeneidade da plataforma ou ao longo caminho entre os dispositivos. Para esse caso, a arquitetura necessita das plataformas para permitir abstrair os protocolos e meios físicos de comunicação, assim como para permitir a facilidade na comunicação entre dispositivos distantes.

4.4.2 Gerenciador de Recursos e Serviços

O Gerenciador de Recursos e Serviços tem três finalidades na arquitetura: busca e descoberta de novos dispositivos (e seus serviços e recursos) no ambiente pervasivo; registrar o dispositivo e os serviços e recursos na rede pervasiva; permitir fácil acesso aos serviços e recursos (quando disponíveis e permitidos). Essas três finalidades são tratadas na arquitetura pela plataforma pervasiva.

A busca e descoberta de novos dispositivos (incluindo seus serviços e recursos) são feitas pelas plataformas de forma similar ao UPnP (exceto se o dispositivo estiver utilizando o padrão bluetooth; nesse caso, a plataforma identifica facilmente utilizando o serviço SDP [7]). O dispositivo que usa o UPnP, chamado de ponto de controle, faz a busca pela rede por dispositivos enviando uma mensagem de busca a todos dispositivos com um critério de busca. Quando um dispositivo tiver aquele mesmo critério, ele responde ao ponto de controle. Esse mecanismo é chamado de SSDP (Simple Service Discovery Protocol). No caso da plataforma, ela percorre toda sua subrede fazendo o papel de ponto de controle. Um exemplo disso aplicado a infra-estrutura da Figura 4.2 é a integração de aparelho de som na subrede IEEE1394 da Sala1. Em um dado momento, a plataforma pervasiva faz a busca pela subrede enviando a busca com o critério "meio de comunicação IEEE1394". Ao receber essa mensagem, todos dispositivos dessa subrede respondem a plataforma e essa identifica o novo dispositivo.

O registro do dispositivo é feito pela plataforma que descobriu o novo dispositivo. Quando a plataforma descobre um novo dispositivo, ela faz o registro localmente do dispositivo em sua memória e comunica a todas outras plataformas próximas da existência do dispositivo e de seus serviços. Continuando o exemplo anterior, a plataforma da Sala1 ao descobrir o aparelho de som, registra o dispositivo e seus serviços na sua base de dados. Após o registro local, a plataforma da Sala1 envia para todas plataformas próximas (no caso as plataformas da Sala2, Cozinha e Suite). Ao receber a mensagem, as plataformas fazem o registro do dispositivo e de seus serviços.

O acesso de uma plataforma a um recurso de um dispositivo inicia pela verificação local para encontrar o recurso. Caso a plataforma não encontre, ela envia uma mensagem para todas as outras plataformas próximas se alguma tem informação sobre o recurso e o dispositivo. Se chegar uma resposta antes de um longo timeout, a plataforma registra as informações do dispositivo e seus recursos. Ao encontrar o recurso, a plataforma solicitadora do recurso faz o pedido para a plataforma solicitada. A plataforma solicitada verifica se o recurso não está sendo utilizado e se é permitido o uso da plataforma solicitadora ao recurso solicitado. Caso seja, o recurso é alocado para a plataforma solicitadora. Seguindo o exemplo anterior, caso a plataforma da Sala2 tenha recebido o registro anterior com falhas, e necessite de um recurso X do aparelho de som, a plataforma da Sala2 vai enviar uma mensagem a todas plataformas próximas solicitando informações do recurso X. As três plataformas (Sala1, Cozinha e Suite), caso não ocorra falhas, informarão a localização do recurso X. Ao receber as informações, a plataforma Sala1 envia um pedido de acesso ao recurso X e a mesma informa a plataforma Sala2 que o recurso está disponível. Então a plataforma utiliza o recurso X.

4.4.3 Adaptação e Reconfiguração

O componente Adaptação e Reconfiguração é responsável pela adição e remoção de serviços e componentes de SW da plataforma para os dispositivos, cujo objetivo disso é a adaptação dos dispositivos a novos tipos de aplicações. Esse componente será necessário em duas situações: o usuário implementa através do framework (seção 4.5) novos serviços e atualizações para aplicações, disponibilizando em algum lugar para download; ou a aplicação (normalmente a AMA - seção 4.5) necessita de um serviço que não se encontra no ambiente.

O componente Adaptação e Reconfiguração é implementada de forma simples. Quando é solicitado, ele faz o download dos serviços em um repositório para armazenamento de serviços e componentes. No entanto, alguns serviços e componentes possuem dependências intercomponentes. Por isso, um arquivo XML contendo as informações (nome e localização) das dependências desses serviços e componentes. Futuramente, pretende-se estender esse modelo simples de consulta para um mecanismo que busca na Internet as dependências (semelhante ao projeto YUM).

4.4.4 Gerenciador de Segurança

O Gerenciador de Segurança deve ser capaz de tratar três objetivos gerais relacionados à segurança: integridade da informação contra adulteração dos dados; confidencialidade dos dados contra a exposição dos mesmos; e disponibilidade do sistema contra a recusa de serviço. Para a implantação desses objetivos, a criptografia, a autenticação e os firewalls ainda continuam sendo os métodos mais eficazes na computação pervasiva. A implementação desses serviços são feitos durante a solicitação dos serviços, verificando-se o acesso de um dispositivo, de um usuário e até mesmo de um ambiente a esses serviços solicitados. Para o caso de tentativas de invasão, alguns serviços de segurança podem ser implementados pelos dispositivos para assegurar proteção de memória, autenticação para acesso de serviços, desabilitação de serviços em caso de problemas, etc.

Inicialmente, apenas duas importantes abordagens sobre segurança são relacionadas na arquitetura: autenticação e controle de acesso.

A autenticação consiste na validação do acesso do usuário no ambiente. O seu objetivo é informar a identidade do usuário, permitindo ao usuário uma identificação no ambiente relacionado a tipos de acesso de serviços e recursos, dispositivos, aplicações, informações, etc. O usuário deixa de ser um usuário público perante o ambiente. Inicialmente, o tipo de autenticação do usuário no sistema é o mecanismo tradicional de login e senha. Pretende-se estender esse mecanismo para smartcards e RFID.

O mecanismo tradicional de login e senha é verificado quando o usuário está acessando uma informação de um terminal ou de um outro dispositivo que está presente no ambiente. Anexado a esse mecanismo, os dispositivos, os serviços e recursos, e as aplicações possuem um campo chamado de Credencial, que informa os usuários e os dispositivos permitidos para o acesso. Por exemplo, utilizando a infra-estrutura da Figura 4.2, um usuário resolve utilizar um PC para acessar a cafeteira. No entanto, o campo Credencial da cafeteira não contém informação do usuário e, portanto, restringe o usuário do seu uso. Existem outros meios de acesso do usuário ao ambiente, como dispositivos pessoais. No entanto, esse tipo de acesso está relacionado ao controle de acesso.

O serviço de controle de acesso permite restringir o acesso de dispositivos maliciosos a informações e recursos importantes na rede do ambiente pervasivo. A plataforma pervasiva possui interceptores (definidos em CORBA [?]) para o acesso desses dispositivos. Um interceptor é responsável por contactar o acesso de um dispositivo a um serviço. Esse interceptor pode interceptar o acesso do dispositivo na plataforma origem (onde o usuário se encontra) ou na plataforma destino (onde o serviço ou recurso se encontra). Por exemplo, na infra-estrutura da Figura 4.2, se

um usuário malicioso com celular de padrão bluetooth conseguir entrar no campo de cobertura do bluetooth da Sala1, e solicitar algum recurso do laptop da Suíte, o interceptor, tanto da plataforma da Sala1 quanto da Suíte, restringirá o acesso desse dispositivo no ambiente.

4.4.5 Gerenciador de Contexto

O Gerenciador de Contexto é o responsável por conhecer o estado atual do usuário, do ambiente e da rede, e adaptar seu comportamento baseando-se nessas informações para atender as necessidades dos usuários. O contexto representa todas as entidades do ambiente e seus atributos. Quatro entidades são definidas para o modelo de contexto: usuário, dispositivo, rede e a local. Cada entidade pode conter importantes informações: Usuário (perfil, preferências, identificador, estados físicos e emocionais, sensações térmicas, etc), dispositivo (tipo, identificação, recursos, serviços, etc), local (temperatura, intensidade da luz, umidade do ar, limites da área, etc) e a rede (largura de banda, latência, tipo de conexão, etc). AURA [46] relaciona essas entidades para formar novas entidades de contexto. As novas entidades e seus respectivos relacionamentos são: Interação Pessoa-Dispositivo, relacionamento entre usuário e dispositivo; Localização de Dispositivos, relação entre área e dispositivo; Integração do Dispositivo e Rede, relação entre dispositivo e rede; Localização de Pessoas, relação entre usuário e área; e finalmente, Localização de Rede, relação entre área e rede.

O funcionamento do Gerenciador de Contexto na arquitetura pode ser visto na Figura 4.3. Seu funcionamento é iniciado com a captura de uma nova informação pelos sensores (ou dispositivos de sensoriamento) ou com a aplicação cadastrando uma nova informação. Quando os sensores capturam uma nova informação, eles repassam para o Serviço de Coleta. A camada de aplicação também cadastra sua informação diretamente no Serviço de Coleta. O Serviço de Coleta transforma essa informação em contexto estático ou dinâmico. Contexto estático é definido nessa arquitetura como uma informação invariável ou uma informação que modifica em intervalos de tempos regulares. Um exemplo desse tipo de contexto seria a latência de uma rede ou localização de dispositivos fixos ao ambiente (que são removíveis raramente pelo usuário). Já o contexto dinâmico possui informações que estão frequentemente se modificando. Um exemplo de contexto dinâmico pode ser a localização de um usuário ou contextos com informações do ambiente (luminosidade, temperatura, etc).

4.4.6 Monitor

O último componente da Arquitetura Pervasiva é o Monitor. Ele disponibiliza a camada de aplicação informações do funcionamento das aplicações da rede, contextos e falhas ocorridas no ambiente. Permite a verificação e avaliação de todo ambiente e todas as plataformas do sistema. Fornece serviços que permite o acesso remoto para monitorar e gerenciar a plataforma pervasiva. Isso possibilita que administradores do ambiente possam avaliar e resolver problemas da plataforma e fazer testes e atualizações de novas versões de aplicações ou sistemas operacionais. Um fator restrito dos serviços de monitoramento remoto é preservar a privacidade dos usuários e das informações contidas no ambiente.

Um exemplo aplicado à residência inteligente da Figura 4.2 é a realização remota do conserto da plataforma pervasiva da Sala2. Por exemplo, suponha que um usuário alterou as configurações dessa plataforma. O componente Monitor disponibiliza serviços para o administrador do ambiente acessar essa plataforma através de um terminal externo, repassando informações de registros da rede e de configurações baixo-nível da plataforma.

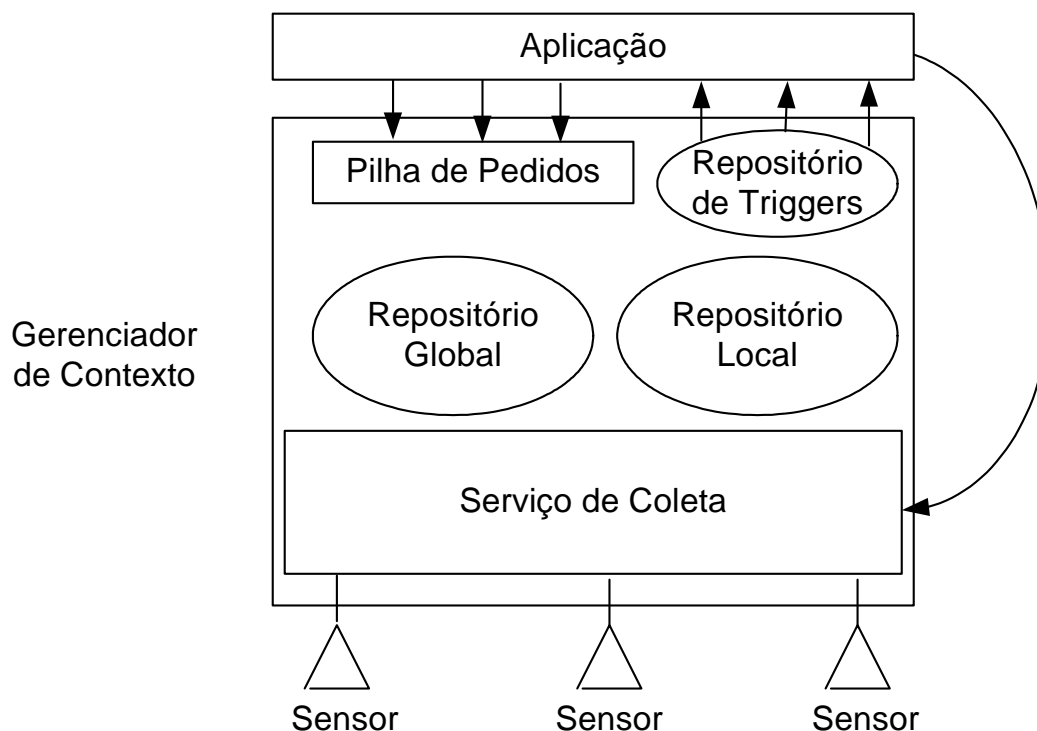


Figura 4.3: Funcionamento do Gerenciador de Contexto na Arquitetura.

4.5 Camada de Aplicação

A construção de uma determinada aplicação pervasiva implica em duas restrições: facilitar ou diminuir a heterogeneidade dos dispositivos e possuir capacidade de adaptação.

Esses problemas podem ser resolvidos facilmente pelo middleware descrito (seção 4.4). O primeiro problema pode ser tratado pelo componente Comunicação e o segundo problema é resolvido adaptando a aplicação facilmente pelo componente Adaptação e Reconfiguração.

Os componentes da Camada de Aplicação podem ser divididos em três tipos de componentes: as aplicações em execução, o Agente Móvel da Aplicação (AMA) e o ambiente de teste e desenvolvimento (framework).

As aplicações em execução são aquelas aplicações que estão sendo executadas nos dispositivos pervasivos com o objetivo de possibilitar ao usuário disponibilidade, invisibilidade e onipresença. Essas aplicações podem ser construídas através de duas formas: criadas pelo desenvolvedor ou usuário no ambiente de teste e desenvolvimento ou determinada pelo AMA, quando esse detectou a necessidade de uma nova aplicação para tratar um novo contexto.

O Agente Móvel da Aplicação (AMA) é uma aplicação inteligente capaz de fornecer serviços inteligentes para a aplicação pervasiva, com o objetivo de tratar contexto e detectar falhas. A detecção de falhas é implementada através da comunicação com o monitor e com o gerenciador de segurança, onde estes fornecem informações relacionadas às falhas para o AMA verificar a ocorrência de falhas.

No caso do tratamento de contexto, o AMA aguarda pelo pedido do gerenciador de contexto.

Nesse tratamento o AMA tem a missão de transformar o contexto em serviços ou em aplicações, através de três formas: perguntar ao usuário quais os serviços que tratam o contexto; descobrir de forma inteligente verificando todos os recursos e serviços; ou buscar por contextos similares.

Ambiente de teste e desenvolvimento ou framework são aplicações que possibilitam a um usuário experiente, projetista ou desenvolvedor construir novas aplicações, novos serviços ou novos recursos para o ambiente. Além disso, framework proporciona testes e avaliações sobre as aplicações, sobre os dispositivos, sobre as redes, sobre os roteadores e sobre os serviços existentes.

Capítulo 5

Modelagem e Implementação da Plataforma Proposta

Na última seção do capítulo 2, os requisitos necessários para desenvolver aplicações em ambientes pervasivos são descritos. Na seção 3.2, foram realizados estudo e avaliação sobre alguns tipos de plataformas para a computação pervasiva. Na avaliação, são comparadas três plataformas com requisitos na construção de uma plataforma HW/SW e com necessidades para os componentes da plataforma. No capítulo anterior, apresentamos a arquitetura pervasiva que serve como base para a captura dos serviços necessários para a construção da plataforma.

Nesse capítulo, a modelagem, o projeto, a implementação e a avaliação da plataforma proposta são descritas. Essa plataforma tem como objetivo integrar dispositivos eletro-eletrônicos em um ambiente pervasivo e fornecer os serviços existentes na rede pervasiva para os dispositivos integrados e fornecer os serviços desses dispositivos para a rede. Além disso, essa plataforma deve fornecer os serviços e recursos necessários para a computação pervasiva.

5.1 Modelagem da Plataforma

O modelo para a plataforma proposta foi projetado para permitir a integração de dispositivos eletro-eletrônicos heterogêneos na rede pervasiva. Esse modelo consiste de simples canais que permitem a fácil integração de dispositivos na plataforma e da plataforma na rede pervasiva.

Na Figura 5.1, o modelo de plataforma para a proposta desse trabalho é apresentado. Esse modelo é composto por dois canais de comunicação abstratos (identificados na Figura 5.1 como Canal de Comunicação A e Canal de Comunicação B). Esses canais permitem a abstração entre os meios físicos e facilitam a integração de heterogêneos dispositivos na rede.

A plataforma HW/SW é composta pelos recursos computacionais para efetuar a comunicação e integração dos dispositivos na rede. Disponibiliza todo o suporte em processamento para fornecer a troca de recursos entre os dispositivos entre si conectados na plataforma e entre os dispositivos na rede externa. Além disso, essa plataforma permite os serviços relacionados à computação pervasiva para os dispositivos integrados a ela.

Os Canais de Comunicação abstraem os tipos de comunicação integrados na plataforma. No modelo de Plataforma apresentado na Figura 5.1, o Canal de Comunicação A é responsável pela integração da plataforma com dispositivos e é responsável por abstrair vários meios de comunicação. Já o Canal de Comunicação B interliga a plataforma com a rede principal, onde esta não foi necessariamente definida na Figura 5.1.

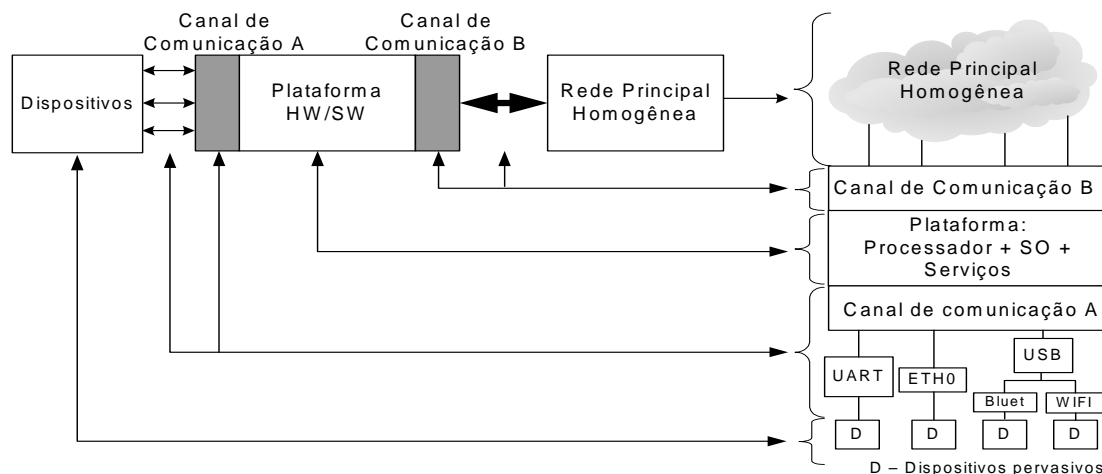


Figura 5.1: Modelo de Plataforma.

5.2 Projeto da Plataforma: Modelo em Camadas

A plataforma HW/SW desse trabalho tem como objetivo adaptar dispositivos eletrônicos a um ambiente pervasivo, onde essa plataforma é capaz de disponibilizar os serviços e recursos desses dispositivos para todos os outros dispositivos e servidores conectados a rede pervasiva. A plataforma HW/SW é composta por quatro camadas, conforme a Figura 5.2: camada física, camada SO, middleware e camada de aplicação.

A camada física visa o tratamento dos dispositivos físicos e é implementada em HW. Essa camada é dividida em quatro componentes: Unidade de Rede, utilizada para conectar a plataforma na rede pervasiva; Unidade de Funcionalidade Específica, utilizada para auxiliar e controlar a funcionalidade eletro-eletrônica do dispositivo; Interface E/S, utilizada para a comunicação e interface com o usuário, ambiente e dispositivos; e Unidade de Processamento, unidade responsável gerenciamento e execução das aplicações e das outras unidades.

A camada SO tem como objetivo tratar as funcionalidades de um sistema operacional para o sistema embarcado. É dividido em duas subcamadas: subcamada de Device Drivers que gerencia os componentes da camada física e subcamada do Sistema Operacional Embarcado que gerencia as tarefas da aplicação que executa no dispositivo e determina os serviços disponibilizados pelo mesmo, além de tratar as limitações do sistema.

O Middleware constitui um conjunto de componentes necessários para auxiliar a integração do dispositivo com a rede pervasiva e disponibilizar os serviços necessários que compõem o middleware da arquitetura definido na seção 4.4. Assim como na arquitetura, essa camada é formada por seis componentes básicos: Módulo de Comunicação, que integra o dispositivo na rede e gerencia a comunicação do dispositivo com outros dispositivos; Gerenciador de Serviços que fornece gerenciamento dos serviços e recursos desse dispositivo para a aplicação pervasiva; Unidade de Reconfiguração que auxilia a arquitetura SW na adaptação de novos componentes SW; Sistema de Autenticação e Controle de Segurança que ajuda na segurança do ambiente, fornecendo serviços de autenticação; Auxiliar de Contexto que auxilia na detecção de contexto a arquitetura; Monitor que fornece o monitoramento do dispositivo para a aplicação, informando erros e problemas.

A camada Aplicação é um fragmento das aplicações que executam no ambiente específica

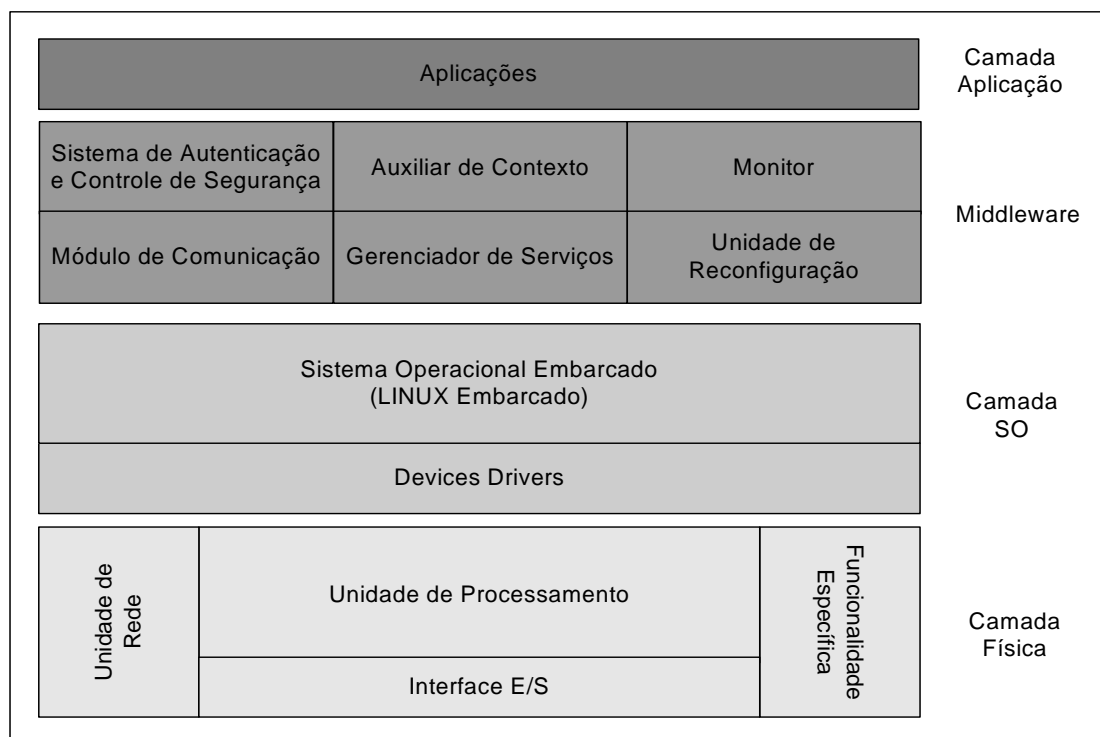


Figura 5.2: Plataforma do Trabalho Estruturada em Camadas.

para o dispositivo embarcado. Um exemplo de aplicação seria disponibilizar os serviços de uma cafeteira para o ambiente pervasivo.

5.3 Implementação da Plataforma

Na etapa de implementação da plataforma HW/SW proposta, analisou-se os recursos HW e SW que poderiam beneficiar a construção da plataforma. Foi decidido, devido ao tempo de projeto, utilizar uma plataforma HW e um sistema operacional, já existentes, para a plataforma proposta. Além disso, o desenvolvimento inicial do middleware foi projetado com poucas qualificações para permitir o primeiro protótipo de plataforma relacionado ao objetivo proposto. Nessa seção o motivo da escolha da plataforma HW e do Sistema Operacional é descrito assim como o desenvolvimento de alguns componentes para o middleware.

5.3.1 Plataforma HW

A etapa de implementação da plataforma HW/SW iniciou com a escolha ou a definição de uma plataforma que seja compatível com a definição da camada física do modelo proposto. Outro fator importante para a escolha da plataforma é que a Unidade de Rede da Camada Física realizasse duas formas de comunicação: cabeada e wireless.

Com base nessas premissas e no tempo de projeto, concluiu-se utilizar uma plataforma em HW já projetada para simular dispositivos semelhantes aos existentes na computação pervasiva.

A plataforma HW escolhida para a camada física foi a plataforma OMAP [47], da empresa Texas Instrument.

Assim a plataforma OMAP utilizada (Figura 5.3) foi utilizada com base nas unidades que compõe a camada física (subseção 5.2). Segue abaixo a relação de cada componente da camada física proposta com as especificações da plataforma:

- Unidade de Processamento: um processador ARM 9 e duas memórias (32Mb DDR-RAM e 32Mb Flash ROM);
- Unidade de Funcionalidade específica: DSP;
- Unidade de Rede: Ethernet;
- Interface E/S: Serial RS-232, entrada e saída de áudio e USB ¹.



Figura 5.3: Fisionomia da Plataforma OMAP.

¹Pretende-se posteriormente integrar um adaptador USB Bluetooth para permitir comunicação Wireless

5.3.2 Sistema Operacional

A camada SO também foi implementada a partir da reutilização de componentes já existentes. O Linux Embarcado pode satisfazer todas as tarefas e requisitos necessários para essa camada: um sistema robusto, com código livre, possui todos os drivers necessários para diversos periféricos e ele é facilmente portátil para outras plataformas.

Para a integração da camada SO com a camada física foi necessário adaptar um sistema linux alvo para o OMAP. A adaptação do sistema linux alvo para a plataforma OMAP é realizada em quatro estágios [48]:

1. Determinar os componentes do sistema Determinar os componentes do sistema consiste em elaborar uma lista de todos os itens necessários para o projeto. Esses componentes envolvem: a identificação do projeto, as informações sobre o hardware de execução (características do processador, memória, etc), ferramentas de desenvolvimento, características do kernel (versão, patches, etc), componentes do sistema de arquivos (informações de dispositivos ou periféricos, principais aplicações, etc), organização dos dispositivos de armazenamento (tamanho, formato, etc), configuração do bootloader (inicializador), serviços de rede, softwares adicionados e informações adicionais.

2. Configurar e desenvolver o kernel A configuração e o desenvolvimento do kernel consistem em selecionar os itens necessários para o sistema e compilar o kernel.

A seleção dos itens determina os componentes que serão incluídos no kernel. Os principais itens são: configuração do processador, tipos da memória, periféricos, rede, armazenamento, opções de rede, sistema de arquivos, segurança, entre outros.

Na fase de compilação do kernel, deve-se primeiramente instalar um conjunto de ferramentas para compilar o kernel. Esse conjunto de ferramentas é chamado de Toolchain. O Toolchain é composto por três ferramentas básicas: o Binutils, que serve para manipular os arquivos binários; o GCC, o compilador; e o Glibc, que é um pacote contendo as principais bibliotecas e pacotes para o Toolchain. O processo de instalação dessas ferramentas é feito da seguinte maneira: (i) seleciona-se a versão do kernel e os fontes; (ii) instala-se a ferramenta Binutil; (iii) instala-se o compilador (GCC); (iv) instala-se as bibliotecas necessárias (Glibc); (v) cria-se o compilador com as bibliotecas anexadas (GCC+GLIBC).

Após a instalação do Toolchain, a próxima etapa da configuração do kernel é configurá-lo. Com base na seleção dos itens feita anteriormente, configura-se o kernel com os principais componentes do projeto. Por fim, compila-se o kernel seguindo suas dependências (processador, compilador, etc) e instalam-se os módulos junto a ele.

3. Desenvolver o sistema de arquivos

O desenvolvimento do sistema de arquivos consiste em construir a estrutura básica do sistema de arquivos e na instalação de alguns componentes: bibliotecas, módulos do kernel, imagens e aplicações.

O desenvolvimento do sistema de arquivos pode ser feito através de uma ferramenta (modo mais fácil) ou desenvolver manualmente (modo mais customizado). Para o presente trabalho, utilizou-se uma ferramenta chamada BusyBox [49]. Essa ferramenta implementa muitos comandos e possui muitas facilidades na instalação dos componentes.

4. Configurar e instalar o software de inicialização

Essa etapa consiste em fazer a carga e a inicialização do Linux na plataforma alvo. É considerada uma das etapas mais difíceis no projeto e é realizada a partir de três fases: (i) manipulação do dispositivo de armazenamento; (ii) preparar o sistema de arquivos; (iii) configuração do bootloader.

A manipulação do dispositivo de armazenamento consiste na seleção do dispositivo de armazenamento, na partição do mesmo, na escolha por Swap (utilizado para memória virtual) e

na preparação do dispositivo de armazenamento para as outras fases.

A preparação do sistema de arquivos consiste na escolha do tipo de sistema de arquivos que sua plataforma deve ter. Essa etapa define: se o sistema de arquivo pode ser escrito; se o sistema de arquivos preserva as modificações ao reinicializar; se o sistema de arquivos recupera as alterações no caso de falha; se o conteúdo do sistema de arquivos montado pode ser comprimido a fim de reduzir seu tamanho; e o local que ficar armazenado o sistema de arquivos antes de ser montado.

A configuração do bootloader consiste na seleção do bootloader do sistema. O bootloader define a forma que vai ser inicializado o sistema e anexa as duas fases anteriores no sistema. Existem diversos bootloaders no mercado e a escolha de cada um é dependente do processador alvo e de suas funcionalidades.

Para o presente trabalho, foi decidido fazer a inicialização do sistema a partir da memória flash. O tipo de sistema de arquivos que foi utilizado foi o NFS [50], onde o sistema de arquivos é carregado para a memória Flash a partir de um servidor de arquivos. Esse processo agiliza o procedimento de testes da aplicação.

Para fazer o papel do bootloader no sistema, utilizou-se as ferramentas BOOTP/DHCP, TFTP e NFS no trabalho. Essa metodologia permite a utilização do NFS para o sistema de arquivos e é maneira mais fácil para fazer a inicialização do sistema.

5.3.3 Middleware

O middleware foi modelado e projetado para fornecer os serviços oferecidos pela computação pervasiva para a plataforma pervasiva. Todos os componentes já foram modelados. No entanto, alguns dos componentes, tais como Unidade de Reconfiguração, Auxiliar de Contexto e Monitor ainda não foram integrados na plataforma, mas outros já foram implementados: Módulo de Comunicação, Gerenciador de Serviços e Sistema de Autenticação e Controle de Acesso. A modelagem de cada um dos componentes do middleware pode ser visto abaixo.

Módulo de Comunicação. O módulo de comunicação tem a tarefa de integrar o dispositivo na rede e gerenciar a comunicação do dispositivo com outros dispositivos. Para poder realizar a comunicação em um ambiente pervasivo, o módulo de comunicação deve suportar heterogeneidade, mobilidade e segurança. Por isso, a idéia inicial do trabalho era implementar o módulo de comunicação utilizando CORBA [34].

No entanto, a dificuldade de adaptar CORBA para a plataforma é grande. Além disso, CORBA ainda não está disponível para a comunicação wireless Bluetooth. Por isso, a idéia inicial do trabalho foi implementar um Módulo de Comunicação simples, utilizando um mecanismo de comunicação orientado a mensagens [22], através de Sockets.

O diagrama de classes para a modelagem do Módulo de Comunicação pode ser visto na Figura 5.4. Esse diagrama possui uma interface que define métodos para conexão e comunicação (Connection) e uma interface que define o tratamento de uma mensagem (CommunicationListener). Seguindo a interface Connection, o Módulo de Comunicação define duas classes que representa a conexão por duas formas diferentes: Socket (SocketConnection) e Bluetooth(Bluetooth Connection).

Outra classe importante para o Módulo de Comunicação é a OpenConnection. Essa classe é responsável por criar a idéia de um servidor, disponibilizando solicitações de conexão e de serviços. Ela também é capaz de encerrar todas as conexões existentes (muito útil para dispositivos com mobilidade).

Gerenciador de Serviços. O gerenciador de serviços tem diversas tarefas para realizar e é responsável tanto pela requisição dos serviços quanto para o atendimento de uma solicitação externa. Para a requisição dos serviços, esse componente ainda deve fazer a busca e/ou descoberta

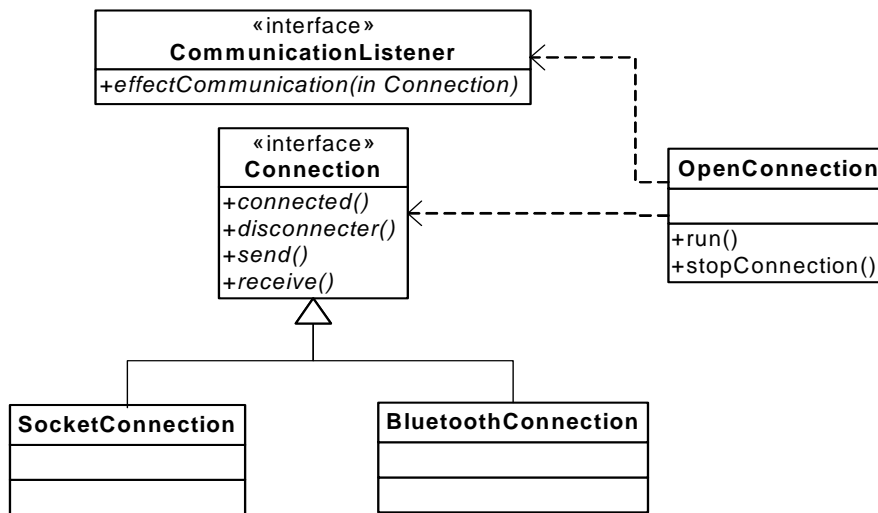


Figura 5.4: Diagrama de Classes do Módulo de Comunicação.

de serviços e a solicitação do serviço. Já para o atendimento de uma solicitação externa, o gerenciador de serviços fica aguardando por uma solicitação de serviço. Ao receber a solicitação do serviço, ele passa a solicitação para o módulo executor que busca o serviço em um repositório de serviço e faz a execução.

A Figura 5.5 mostra o diagrama de classes do Gerenciador de Serviços. Esse componente é composto por nove classes: Controller, Discovery, Executor, Register, Namespace, ResourceBroker, ServiceRepositoryManager, Gateway e Monitoring. A classe Controller é responsável pela comunicação da aplicação com esse módulo e controla as outras classes do sistema. A classe Discovery tem a função de descoberta de dispositivos e serviços na rede. Ao encontrar um novo dispositivo ou um novo serviço na rede, insere-o na classe Namespace, que é o responsável por disponibilizar informações sobre os dispositivos e serviços da rede.

A classe Register faz a solicitação de um pedido de registro do dispositivo no ambiente pervasivo. A classe ResourceBroker é a responsável por buscar e solicitar serviços na rede pervasiva.

A classe Gateway é a responsável por solicitar a comunicação na rede para disponibilizar ou buscar um serviço. A classe Monitoring aguarda por solicitações de serviços. A classe responsável por executar os serviços existentes no dispositivo é o Executor. E por fim, a classe ServiceRepositoryManager gerencia, controla, cria, remove e atualiza todos os serviços existentes no dispositivo alvo.

Unidade de Reconfiguração A Unidade de Reconfiguração foi projetada para adicionar, remover e adicionar serviços e aplicações para a plataforma. Seu objetivo é permitir adaptação da plataforma para suportar novos tipos de aplicações

A Figura 5.6 apresenta o diagrama de classe da Unidade de Reconfiguração. A adição de novos serviços funciona da seguinte maneira: a aplicação faz solicitação de novos serviços, seja através do framework ou do AMA. O ReconfigurationManager da plataforma faz a solicitação de download para o ServicesDownloader. Esse componente registra todas dependências em um arquivo e envia o mesmo junto com o serviço ou aplicação para o ReconfigurationManager. Quando esse componente recebe todos esses módulos, ele busca as dependências e adiciona tudo no ServiceRepositoryManager, emitindo uma solicitação de registro para o Gerenciador de Serviços.

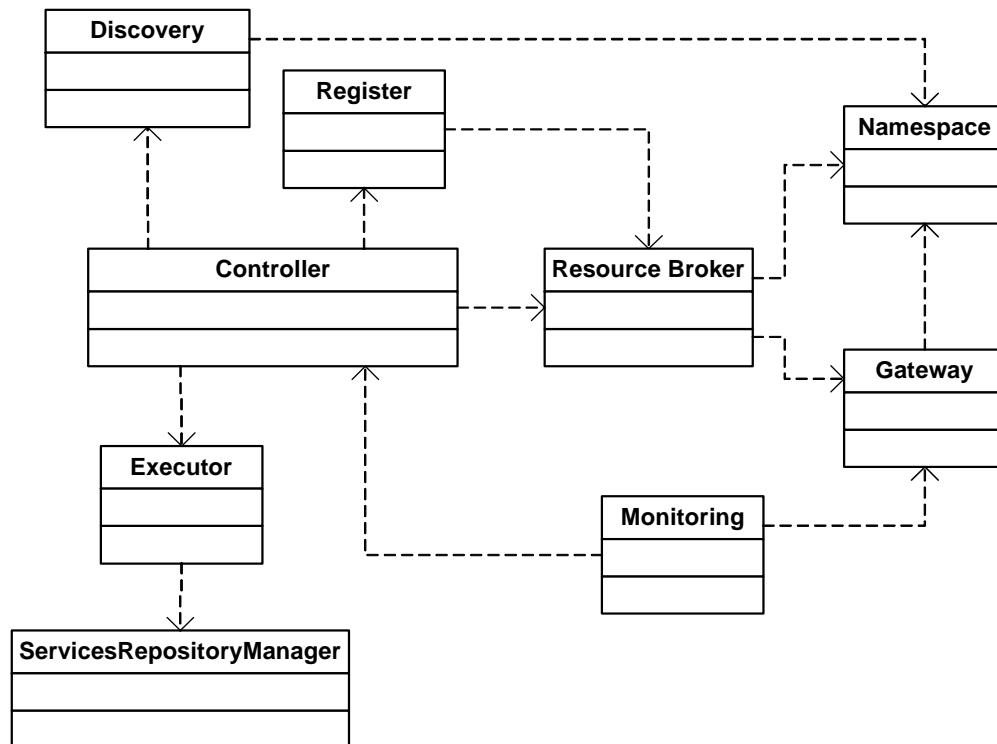


Figura 5.5: Diagrama de Classes do Gerenciador de Serviços.

O processo de atualização da aplicação é semelhante. No entanto, ao invés da aplicação solicitar a atualização, um recurso chamado `ApplicationUpdater` faz a solicitação automática. Esse serviço serve para facilitar o AMA e o framework para manter as aplicações do sistema atualizadas.

Sistema de Autenticação e Controle de Acesso.

O Sistema de Autenticação e Controle de Acesso é o responsável por tratar da segurança na plataforma, nos dispositivos e nos serviços realizados por ambos. Ele permite que usuários maliciosos tenham baixo ou nenhum acesso ao sistema. duas importantes abordagens sobre segurança são relacionadas na arquitetura: autenticação e controle de acesso.

A autenticação consiste na validação do acesso do usuário no ambiente. O seu objetivo é informar a identidade do usuário, permitindo ao usuário uma identificação no ambiente relacionado a tipos de acesso de serviços e recursos, dispositivos, aplicações, informações, etc. O usuário deixa de ser um usuário público perante o ambiente. Já o mecanismo de controle de acesso restringe as entidades do ambiente na utilização de determinados serviços e recursos. Essa restrição é feita através de um registro de informações chamado de `Credencial`. Um `Credencial` permite restringir, por exemplo, o acesso de um usuário a um serviço permitido apenas ao administrador do sistema.

O diagrama de classes da Figura 5.7 mostra as três classes existentes no Sistema de Autenticação e Controle de Acesso. Esse componente funciona da seguinte maneira: um usuário ou administrador está interagindo com uma determinada aplicação. Quando essa aplicação necessitar de um recurso ou serviço, solicita a classe `Logger` para a plataforma. Esse `Logger` verifica

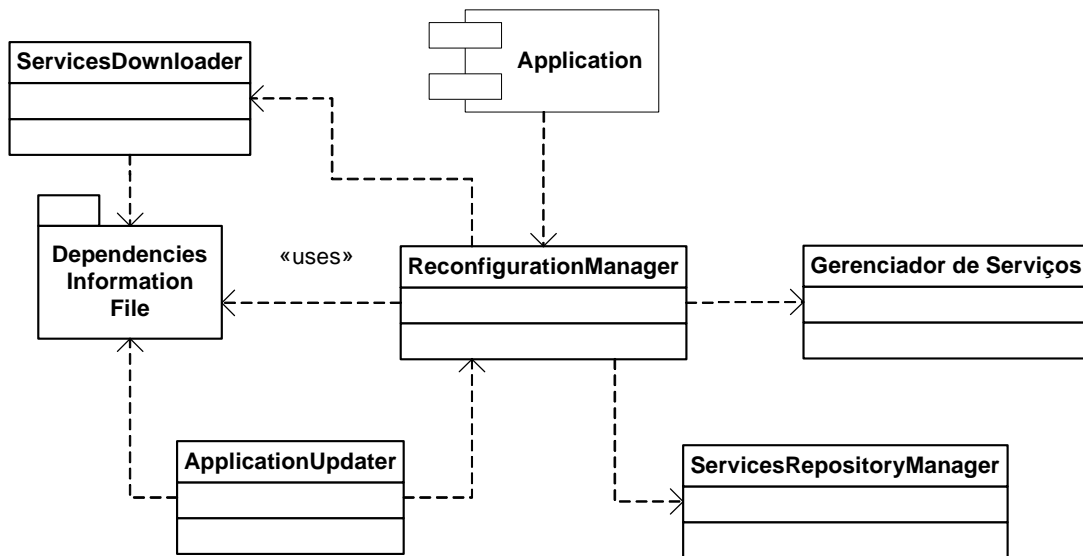


Figura 5.6: Diagrama de Classes da Unidade de Reconfiguração.

com o Authentication os dados do usuário e com o CredentialRepositoryManager, através do Credential específico, se é permitido o usuário acessar aquele serviço.

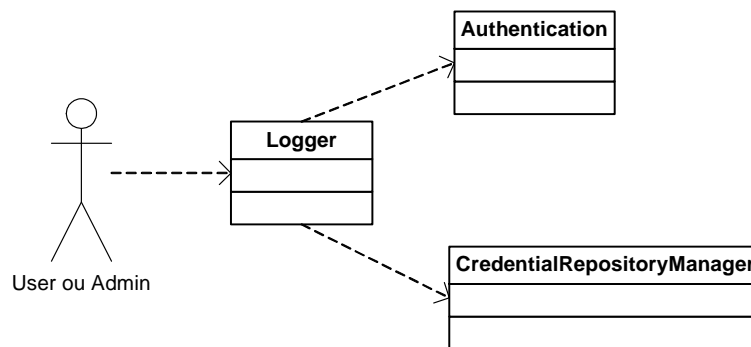


Figura 5.7: Diagrama de Classes do Sistema de Autenticação e Controle de Acesso.

Auxiliar de Contexto.

O Auxiliar de Contexto tem a finalidade de ajudar o gerenciamento de contexto no ambiente. Esse componente é responsável por fazer o tratamento local do gerenciador de contexto e fornecer auxílio para o acesso global de um determinado contexto. Dois tipos de contexto são tratados: contexto estático é aquele que tem suas informações modificadas raramente (ex. latência de uma rede ou localização de um dispositivo fixo); e contexto dinâmico é aquele que varia suas informações constantemente (ex. localização de um usuário).

Na Figura 5.8, o diagrama de classe do Auxiliar de Contexto é apresentado. Esse módulo é praticamente separado em dois submódulos: o primeiro submódulo é para tratar do contexto dinâmico, do contexto que se encontra localmente na plataforma. A detecção de contexto é feita através do Colector2 que coleta informações do ambiente (através de sensores e dispositivos) e armazena-as no ContextLocalRepositoryManager. O tratamento desse tipo de contexto é feito através de Triggers que executam imediatamente um método com a coleta de um determinado contexto. A aplicação tem o papel de construir esses Triggers e esses são armazenados no TriggerRepositoryManager.

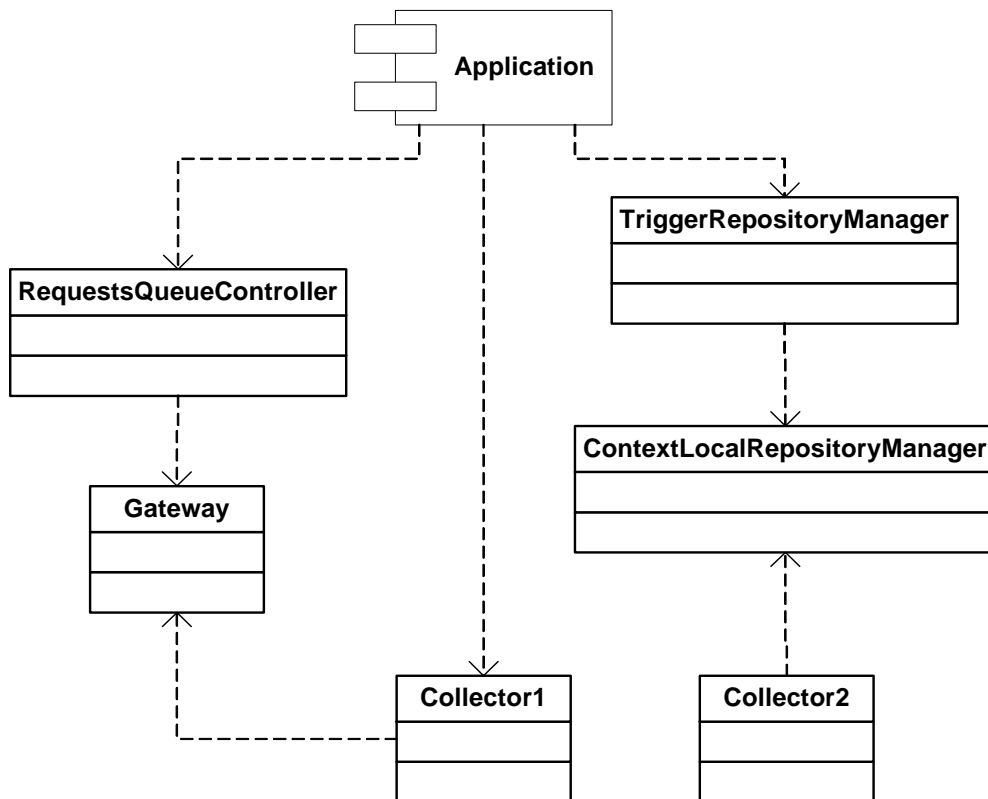


Figura 5.8: Diagrama de Classes do Auxiliar de Contexto.

O segundo módulo consiste apenas na inserção e no tratamento de um contexto estático. No caso da inserção de contexto estático, a aplicação solicita ao Collector1 a adição de novos contextos que repassa para o Gateway. Esse faz a solicitação de adição de contexto remotamente para os Servidores de Contexto ou para outras plataformas. No caso do tratamento de contexto,

o processo é inverso. A aplicação faz uma requisição de um determinado contexto para o RequestsQueueController. Esse empilha o pedido até estar pronto para atendê-lo. Quando estiver apto para o tratamento, o módulo RequestsQueueController faz uma consulta remota a base de dados do contexto. As informações do contexto são repassadas para a aplicação.

Monitor Por fim, o Monitor tem como objetivo fornecer recursos de monitoramento da plataforma a usuários e aplicações do ambiente. Esse módulo propicia o acesso a informações do funcionamento das entidades existentes no ambiente, mas prevê privacidade aos usuários. O monitor permite o acesso remoto de usuários distantes para monitorar e gerenciar serviços, recursos e dispositivos da plataforma. Isso possibilita a administradores do ambiente avaliar e resolver problemas da plataforma e fazer testes e atualizações de novas versões de aplicações ou de sistemas operacionais, mesmo estando longe do ambiente pervasivo.

O diagrama de classes do componente Monitor pode ser visto abaixo na Figura 5.9. O funcionamento é dado da seguinte maneira: um administrador faz acesso remoto na plataforma para solicitar o serviço de monitoramento. Para isso, o usuário conecta no WaitingConnection, verificando o controle de acesso do usuário a informação. No momento que foi autenticado, o usuário acessa o monitor. O monitor permite acesso aos arquivos de registro do sistema e a determinados serviços do ServiceRepositoryManager.

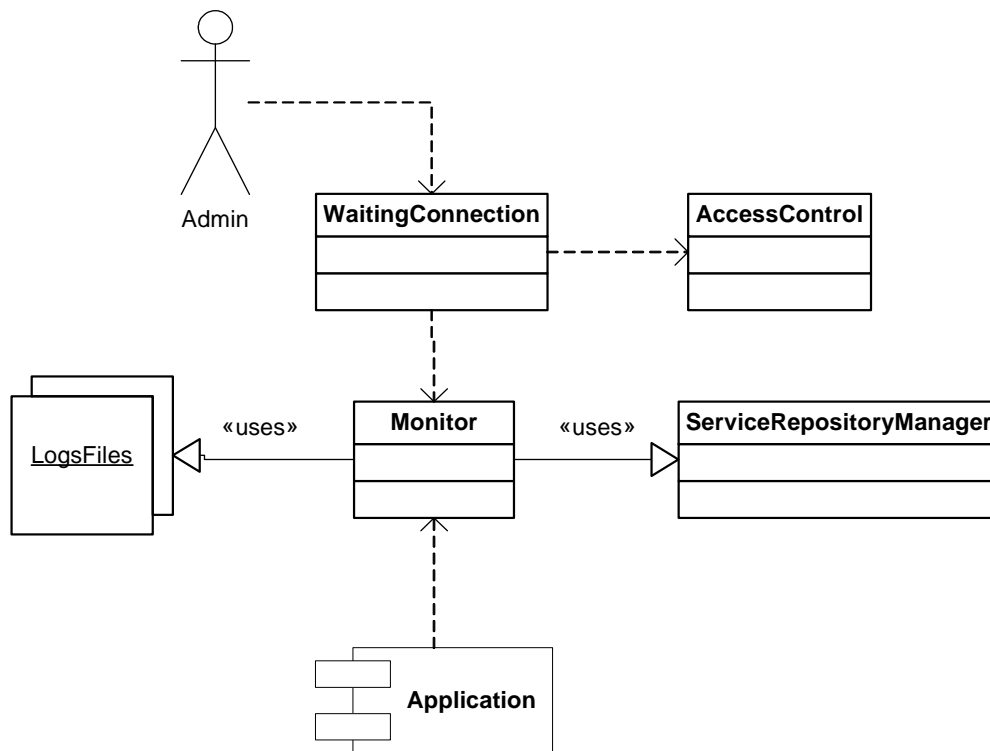


Figura 5.9: Diagrama de Classes do Monitor.

5.4 Avaliação da Plataforma

No final do capítulo 3, realizou-se um estudo comparativo relacionado a três tipos de plataformas voltadas para a computação pervasiva: gateways pervasivos, plataformas para objetos inteligentes e plataformas de interação com o usuário. Essa seção adiciona a plataforma pervasiva proposta na comparação, descrevendo uma pequena tabela semelhante à apresentada no estudo das plataformas.

Tabela 5.1: Avaliação da Plataforma Pervasiva Proposta.

Propriedades	Plataforma Pervasiva Proposta
Custo	Alto
Tamanho	14 cm
Consumo de Energia	Alto
Interatividade	Alta
Invisibilidade	Alta
Mobilidade	Permitida
Gerenciamento de recursos	Sim
Escalabilidade	Alta e média
Configurabilidade	Sim
Segurança	Alta
Suporte a Falhas	Sim
Suporte a Monitoramento	Médio - Alto
Deteção de Contexto	Sim
Tratamento de Contexto	Não

Conforme foi mencionado no estudo comparativo (subseção 3.2.4), a plataforma proposta nesse trabalho tem uma finalidade muito semelhante ao gateway. Por isso a plataforma possui muitas características e serviços iguais ao gateway: alto custo, tamanho e consumo de energia; Gerenciamento de Recursos; Segurança; Configurabilidade; Suporte a Monitoramento. No entanto, a plataforma pervasiva desse trabalho possui suporte a outros serviços relacionados a computação pervasiva, permitindo serviços inexistentes no gateway, como é o caso de invisibilidade, alta escalabilidade, deteção e tratamento de contexto e de mobilidade (as características da plataforma pode possuir suporte a mobilidade).

Capítulo 6

Aplicação Agenda Pervasiva e Avaliação da Plataforma

Esse capítulo tem como objetivo demonstrar a utilização da plataforma proposta para atender uma aplicação voltada a computação pervasiva. Essa aplicação, chamada de Agenda Pervasiva, é utilizada para tratar das tarefas e compromissos dos usuários em um ambiente distribuído e inteligente.

Inicialmente, é apresentada a descrição da aplicação, da infra-estrutura necessária para executá-la e mostramos um exemplo prático de funcionamento. Em seguida, alguns experimentos relacionados a essa aplicação são realizados com o objetivo de fazer uma avaliação funcional para verificar o comportamento do sistema. Por fim, a contribuição da plataforma para a aplicação Agenda Pervasiva e a utilização de outros recursos para auxiliar na "pervasividade" do ambiente são mostrados.

6.1 Aplicação Agenda Pervasiva

A aplicação Agenda Pervasiva é uma agenda eletrônica com novas funcionalidades utilizadas para computação pervasiva. A idéia repassada por essa aplicação é relacionada ao conceito "Siga-me". Esse conceito é um termo utilizado na pesquisa acadêmica para denominar a onipresença aos ambientes e às aplicações pervasivas.

Nessa seção, a descrição e o funcionamento da aplicação são apresentados. Em seguida, dois tipos de infra-estruturas aplicadas à Agenda Pervasiva são mostrados. Essas infra-estruturas são responsáveis por determinadas tarefas envolvidas na aplicação. Um exemplo prático relacionado a essa aplicação é apresentado para demonstrar o funcionamento do sistema.

6.1.1 Descrição da Aplicação

A Agenda Pervasiva não é uma aplicação para ser utilizada apenas em uma máquina ou em um dispositivo. Ela envolve um ambiente pervasivo e por isso, ela executa em conjunto com diversas máquinas, possui suporte para atender inúmeros usuários e utiliza diferentes tipos de recursos (ex. localização e tocador de música).

A Agenda Pervasiva é uma aplicação distribuída que necessita de muitos recursos para atender as suas funcionalidades e suas finalidades. Conforme mencionado anteriormente, uma das finalidades dessa aplicação é atender inúmeros usuários permitindo o cadastro de seus compro-

missos e suas tarefas. A segunda finalidade é permitir alta interatividade com esse usuário, disponibilizando diversos recursos para fazer o cadastro e a visualização de suas tarefas. Essa finalidade tem como proposta atender o requisito onipresença. E por fim, a última finalidade é fornecer o rastreamento e a localização do usuário, para auxiliar o usuário a relembrar de suas tarefas.

A agenda pervasiva funciona da seguinte maneira:

1. Um usuário cadastrado no sistema insere compromissos especificando data, horário, descrição, mensagem e sala. A data e o horário representam a fatia de tempo que a tarefa deve ser realizada. A descrição representa o próprio compromisso. A música representa uma mensagem de voz ou música que deve ser tocada no ambiente para lembrar o compromisso. A sala indica o local que o compromisso será realizado;
2. Um componente da aplicação chamado Gerente fica interagindo com esses terminais, efetuando o cadastro dos usuários e dos compromissos e atendendo todos pedidos dos usuários (ex. visualização de seus compromissos no ambiente). Esses cadastros e visualização são feitos através da interação do Gerente e do Servidor de Perfil, que contém os registros dos usuários e de seus compromissos;
3. Existe dois tipos de acesso aos terminais: ADMIN, que possibilita o usuário executar tarefas mais sofisticadas tais como cadastro de usuários, localização de um usuário, visualização das salas, do ambiente, dos usuários em uma sala, etc; DEFAULT, onde cada usuário acessa e modifica seus compromissos e tarefas assim como suas informações pessoais;
4. O Gerente controla, ainda, o tempo e todas as tarefas de todos usuários a fim de verificar quais os compromissos que estão sendo realizados e se estão sendo cumpridos pelos usuários;
5. Um tipo de localizador ou de servidor de localização adquire e contém informações relacionadas à localização de um determinado usuário (ou de seu dispositivo de uso);
6. Quando uma tarefa do usuário "estoura" o limite de tempo, o gerente faz uma solicitação para o servidor de localização pedindo a localização do usuário;
7. O Gerente verifica no Servidor de Perfil se o usuário está localizado na sala em que o compromisso deve ser realizado. Se o usuário estiver nessa sala, o compromisso é cumprido e uma mensagem sonora (cadastrada na etapa 1) é emitida no ambiente através de plataformas pervasivas existentes em cada sala.

6.1.2 Infra-estrutura do Ambiente para a Aplicação

Conforme foi visto na subseção anterior, a Agenda Pervasiva é composta por alguns dispositivos e componentes que realizam tarefas específicas para realizar o conceito de "pervasividade" para a aplicação. É o caso dos servidores de perfil e de localização, localizador, gerente e plataforma pervasiva.

A Figura 6.1 mostra uma infra-estrutura do ambiente que pode ser aplicado para a aplicação Agenda Pervasiva. Essa infra-estrutura teve a colaboração especial do Sistema de Localização Holo [51], apoiado pelo grupo de pesquisa Mobilab da UNISINOS.

Baseado na Figura 6.1, a infra-estrutura do ambiente possui os seguintes componentes:

- Gerente - tem controle sobre todo sistema e sobre a aplicação. É o responsável pela interação com todos os outros componentes (exceto as caixas de som e o localizador). É responsável por autenticar o usuário, cadastrar e buscar usuários e compromissos junto com os servidores e por controlar o registro para fornecer a funcionalidade da aplicação.

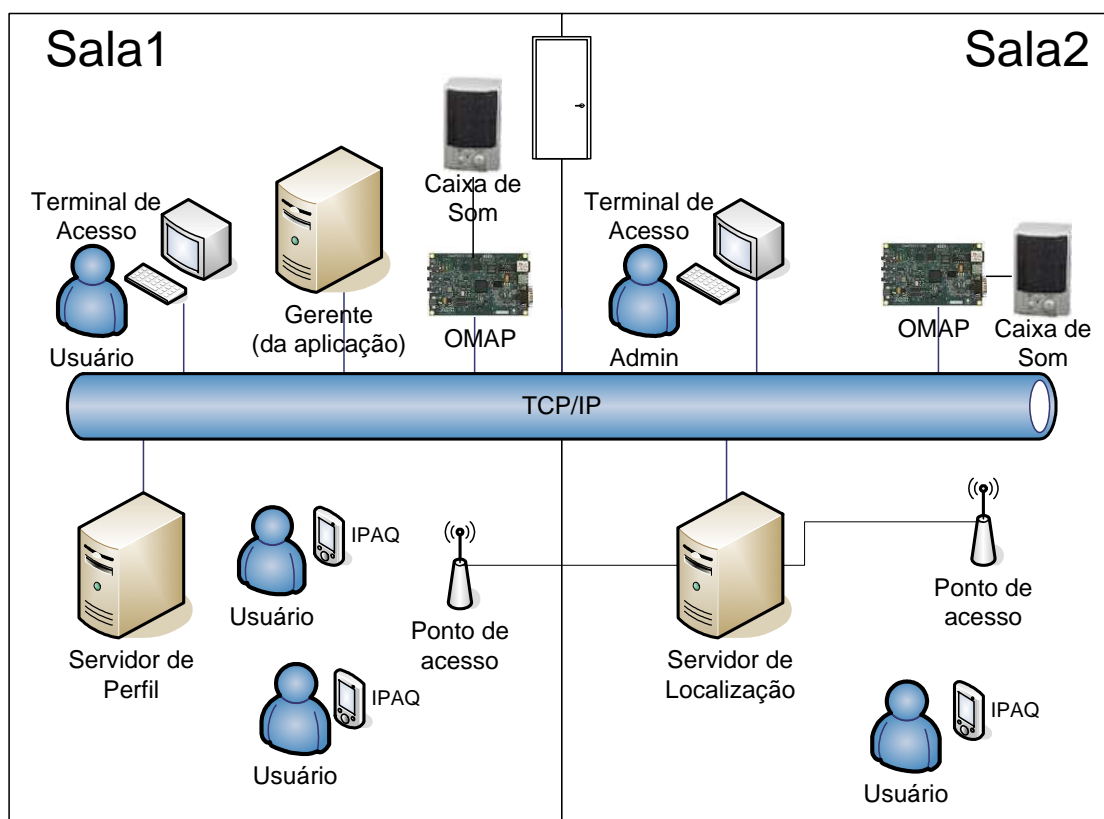


Figura 6.1: Infra-estrutura WIFI da aplicação Agenda Pervasiva.

- Servidor de Perfil - É um servidor com uma base de dados que cadastra, busca e acessa os usuários do ambiente, além de seus compromissos.
- Servidor de Localização - É o responsável por buscar, armazenar e fornecer informações relacionadas à localização de um usuário.
- Localizador - tem como localizar um usuário ou o seu dispositivo de uso
- OMAP - tem como funcionalidade permitir o acesso do dispositivo caixa de som em cada sala à rede pervasiva. Forneceu todo suporte de drivers para fornecer o serviço "Emitir mensagem sonora".
- Caixas de Som - dispositivo de interação com o usuário.
- Terminais de Acesso - terminais que permitem acesso ao usuário na rede pervasiva para, junto com o gerente, cadastrar novos usuários (no caso do administrador) ou novos compromisso (no caso de usuários cadastrados no sistema).

6.1.3 Exemplo de Funcionamento da Aplicação

A Agenda Pervasiva tem capacidade para muitos usuários com diversos compromissos cadastrados. Conforme foi visto anteriormente, o sistema é capaz de localizar qualquer usuário no

ambiente. Na Tabela 6.1, apresentamos alguns exemplos de usuários, com seus compromissos cadastrados e com a informação de sua localização.

Tabela 6.1: Exemplo de Funcionamento: Usuário, compromisso e localização.

Usuário	A				B			
Compromissos	Data	Hora	Música	Sala	Data	Hora	Música	Sala
C1	10/02	9hs	mus1	S1	10/02	9hs	mus2	S2
C2	10/02	10hs	mens1	S3	10/02	10hs	mus1	S1
C3	11/02	11hs	mus2	S1	10/02	11hs	mens3	S2
Localização:	Sala S1				Sala S2			

Suponha-se que os usuários cadastrados em um ambiente sejam os usuários da Tabela 6.1, juntos com seus compromissos e com sua localização. Os usuários somente serão informados por uma mensagem sonora provida pelo ambiente, se ele estiver realizado o seu compromisso no local e na hora exata. Por exemplo, se o usuário A não estiver na sala S1 (local destinado) às 9hs (horário destinado), a mensagem do compromisso C1 do usuário A não emitirá nenhum sinal sonoro a nenhuma sala.

Com relação ao acesso ao terminal, o usuário A possui acesso somente a seus compromissos e o usuário B a seus compromissos e assim sucessivamente. Nenhum usuário tem acesso a um compromisso de outro usuário. Outro fator importante relacionado ao acesso dos usuários é relacionado ao cadastro e adição de novos usuários. Somente o administrador da aplicação possui acesso restrito a esse tipo de informação.

6.1.4 Implementação da Aplicação

A Agenda Pervasiva foi implementada em C++ sobre a plataforma Linux em todos os PCs. A plataforma OMAP, conforme foi mencionado no capítulo anterior, utiliza o sistema operacional Linux Embarcado. A comunicação, também mencionada no capítulo anterior, foi realizada utilizando TCP/IP. A plataforma, primeiramente, foi conectada apenas a um dispositivo de áudio. O padrão utilizado na localização de um usuário é o WIFI. A localização realizada pelo grupo de pesquisa Mobilab da UNISINOS [51] é feito sobre um certo número de pontos de acesso, que efetuam um cálculo em conjunto para fornecer uma maior precisão na localização de um determinado dispositivo do usuário. O servidor de localização, baseado em WIFI, utiliza o padrão Web Services na sua comunicação. No sistema foi necessário o uso de um Proxy para poder fazer a interoperabilidade com o servidor.

6.2 Resultados Obtidos

O objetivo dessa seção é mostrar os serviços da aplicação, visualizar o comportamento do sistema e fornecer uma validação funcional, mostrando a realização do serviço através de telas capturadas no sistema.

Para alcançar esse objetivo, três experimentos relacionados a essa aplicação são avaliados: inicialização do ambiente e dos dispositivos; solicitação de um serviço pelo usuário; e a verificação dos compromissos dos usuários. Em cada experimento, vamos analisar todas as operações realizadas entre os dispositivos, onde em cada passo de um experimento, vamos mostrar a realização dos serviços em cada componente. Os experimentos foram elaborados com base na primeira infra-estrutura apresentada na Figura 6.1, utilizando o sistema de localização WIFI. A execução

realizada nas plataformas pervasivas definidas sobre o OMAP são acessadas através de terminais se comunicando via serial: no caso do OMAP1, foi utilizado o Kermit do Linux e no caso do OMAP2, foi acessada sobre o Hyperterminal do Windows.

6.2.1 Serviços Disponíveis no Sistema

A aplicação Agenda Pervasiva permite a capacidade de realizar serviços internos e externos ao ambiente. Isso é permitido, pois o Terminal de Acesso não é anexado ao Gerente da Aplicação. O Terminal de Acesso pode ser feita remotamente em um outro ambiente conectado à Internet.

A aplicação Agenda Pervasiva possui dois tipos de usuários, conforme sua permissão no ambiente: Administrador e DefaultUser. O Administrador possui serviços mais específicos relacionados a controlar e monitorar o ambiente. A Figura 6.2 mostra os serviços realizados pelo Administrador, numa consulta do help do sistema para o usuário Administrador.

```

Usuário ries deslogado
*****
Login: root
Password: ****
12345
ADMIN logado
:)|-> help
Help para ADMIN
createUser - cria um usuário
viewUsers - visualiza todos usuários pelo username
infoUser - mostra as informações de um usuário a partir do username
deleteUser - deleta um usuário
changePassword - modifica o password do usuário ADMIN
createSala - cria uma nova sala para o ambiente
deleteSala - deleta uma sala
localizeUser - informa a localização de um usuário em uma das salas
viewSalas - mostra todas as salas existentes no ambiente
viewUsersInSala - mostra todos usuários em uma sala
viewUsersInSala - mostra todos usuários em uma sala
loadPerfilXML - carrega o servidor de Perfil com o arquivo XML
loadLocalXML - carrega o servidor de Localização com o arquivo XML
help - ajuda
exit - sair
:)|->

```

Figura 6.2: Serviços específicos para administrador no Terminal de Acesso.

O usuário DefaultUser tem acesso somente a suas tarefas no sistema. Por isso, as únicas informações que esse pode visualizar e alterar são relacionadas a suas informações específicas ou a suas tarefas. A Figura 6.3 apresenta os serviços permitidos por um usuário comum no ambiente. Esses serviços são mostrados através de uma chamada por "help" de um usuário existente no ambiente.

```

Login: ries
Password: ***
123
USER ries logado
:)|-> help
Help para default user
help - ajuda
exit - sair
createTask - cria uma nova tarefa
viewAllTasks - mostra todas tarefas do usuário
viewTaskByTime - mostra todas tarefas do usuário em um determinado horario
viewTaskByDate - mostra todas tarefas do usuário em um determinado dia
viewTaskByDatetime - mostra todas tarefas do usuário em um determinado horario e
dia
viewTaskByDescricao - mostra todas tarefas do usuário com uma determinada descri
ção
viewTaskBySala - mostra todas tarefas do usuário em um determinado local
myInfo - mostra as informações do usuário
infoTask - mostra as informações de uma tarefa a partir do código
deleteTask - deleta uma tarefa
updateUser - atualiza as informações do usuário
changePassword - modifica o password do usuário
updateTask - atualiza as informações de uma determinada tarefa
:)|->

```

Figura 6.3: Serviços específicos para usuários default no Terminal de Acesso.

6.2.2 Experimento 1: Inicialização do ambiente para Agenda Pervasiva

O primeiro experimento realizado é relacionado a inicialização de todo o sistema, ou seja, de todos dispositivos e servidores no ambiente (exceto o Servidor de Localização Holo). Nesse experimento, os dispositivos e servidores se conectam entre si, fornecendo informações e registros de serviços uns aos outros.

Nessa seção, a inicialização do ambiente para agenda pervasiva foi elaborada em passos, onde cada passo pode ser realizado por um ou mais componentes do sistema¹. Para avaliar o sistema, capturou-se as telas de cada um dos componentes executando a aplicação. As Figuras 6.4 e 6.5 mostram as telas de execução das plataformas pervasivas do ambiente: OMAP1 e OMAP2. A Figura 6.6 mostra a execução do Gerente da Aplicação. A Figura 6.7 mostra a tela de execução do Servidor de Localização, enquanto a Figura 6.8 apresenta a execução do Servidor de Perfil. A Figura 6.9 mostra a execução do Terminal de Acesso. O funcionamento de cada componente mostrado nessas Figuras é apresentado pelos seguintes passos a seguir. Cada passo é representado nas Figuras dos componentes.

1- Servidor de Localização Proxy faz conexão no Servidor de Localização Holo. Servidor de Perfil e Servidor de Localização Proxy ficam esperando conexão de Gerente.

2- Gerente se conecta em ambos servidores e permite a conexão dos demais componentes, seja para as plataformas pervasivas registrarem-se, seja para os terminais de acesso para execução de serviços.

3- Gerente dispara seu relógio. Esse relógio possui um timer de 30 segundos que serve para verificar a existência de algum compromisso no momento. O relógio é o responsável pelo controle da execução da aplicação, pois é ele que verifica o compromisso, averigua se o usuário está cumprindo seu compromisso e envia a mensagem sonora para as plataformas pervasivas. Por isso

¹Componentes vistos na Figura 6.1. São eles: Gerente (da aplicação), Servidor de Perfil, Servidor de Localização, Terminal de acesso, OMAP da Sala1 (nomeado como OMAP1) e OMAP da Sala2 (nomeado como OMAP2).

```

ries@localhost:~/AgendaPervasiva/apl_omap - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help

[apl_omap]$ ./agenda_omap 2023
<DEBUG> Enviando para Gerente pedido de registro do dispositivo!
<DEBUG> Enviando nome: OMAP1
<DEBUG> Enviando sala: S1
<DEBUG> Enviando IP da plataforma: 192.168.1.2
<DEBUG> Registro realizado
<DEBUG> Criando novo serviço(nome,local,par):
-> (tocamusic,</bin/tocamusic>,1)
<DEBUG> Enviando para Gerente pedido de registro de serviço!
<DEBUG> Enviando plataforma: OMAP1
<DEBUG> Enviando serviço: tocamusic
<DEBUG> Registro de serviço realizado
<DEBUG> Iniciando monitoramento da conexão...
    
```

Passo 4

Passo 5

Passo 6

Figura 6.4: Inicialização da aplicação Agenda Pervasiva no componente OMAP1.

```

ries@localhost:~/AgendaPervasiva/apl_omap - Shell No. 6 - Konsole
Session Edit View Bookmarks Settings Help

[apl_omap]$ ./agenda_omap 2024
<DEBUG> Enviando para Gerente pedido de registro do dispositivo!
<DEBUG> Enviando nome: OMAP2
<DEBUG> Enviando sala: S2
<DEBUG> Enviando IP da plataforma: 192.168.1.3
<DEBUG> Registro realizado
<DEBUG> Criando novo serviço(nome,local,par):
-> (tocamusic,</bin/tocamusic>,1)
<DEBUG> Enviando para Gerente pedido de registro de serviço!
<DEBUG> Enviando plataforma: OMAP2
<DEBUG> Enviando serviço: tocamusic
<DEBUG> Registro de serviço realizado
<DEBUG> Iniciando monitoramento da conexão...
    
```

Passo 7

Figura 6.5: Inicialização da aplicação Agenda Pervasiva no componente OMAP2.

ele se conecta em ambos servidores.

4- OMAP1 solicita seu registro de dispositivo para Gerente. Gerente registra OMAP1 na sua lista de dispositivos.

5- OMAP1 cria o serviço "tocamusic" e faz pedido para Gerente para registrá-lo. Gerente registra o serviço tocamusic de OMAP1 na lista de dispositivos.

6- OMAP1 inicia monitoramento de sua conexão para verificar solicitação de serviços.


```

ries@localhost:~/AgendaPervasiva/Gerente - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
[Gerente]$ ./Gerente 2022
<DEBUG> Conectando-se ao Servidor Perfil
<DEBUG> Conectando-se ao Servidor Localização
<DEBUG> Conectado ao Servidor Perfil !!
<DEBUG> Conectado ao Servidor Localização!!
<DEBUG> Liberando conexões!!
<DEBUG-Relogio> Inicializando Thread do Relógio
<DEBUG-Relogio> Relógio conectado em Servidor Perfil
<DEBUG-Relogio> Relógio conectado em Servidor Local.
<DEBUG> Solicitação de Registro de dispositivos!
<DEBUG> OMAP1 registrado na lista de dispositivos!!
<DEBUG> Solicitação de Registro de Serviços!
<DEBUG> tocamusic registrado no dispositivo OMAP1
<DEBUG> Solicitação de Registro de dispositivos!
<DEBUG> OMAP2 registrado na lista de dispositivos!!
<DEBUG> Solicitação de Registro de Serviços!
<DEBUG> tocamusic registrado no dispositivo OMAP2
<DEBUG> Terminal conectando-se ao Gerente... OK!
<DEBUG> Solicitação de Login por Terminal de Acesso!
Recebido Login e senha..
Login: root
Senha privativa!
<DEBUG> Tipo de usuário: ADMIN
<DEBUG> Verificando senha do root no Servidor de Perfil...
<DEBUG> Usuário OK!!
<DEBUG> Senha OK!!
<DEBUG> ADMIN logado
<DEBUG> Restringido acesso somente a serviços ADMIN
<DEBUG> Aguardando por solicitação de serviço...
<DEBUG-Relogio> Consulta ao Servidor Perfil:
<DEBUG-Relogio> Date: 12/12/2006
<DEBUG-Relogio> Time: 23:10
<DEBUG-Relogio> Nenhum compromisso pendente!!

```

Passo 2

Passo 3

Passo 4

Passo 5

Passo 7

Passo 8

Passo 9

Passo 10

Passo 11

Passo 12

Passo 13

Figura 6.6: Inicialização da aplicação Agenda Pervasiva no componente Gerente.

7- OMAP2 registra a si e seu serviço, e inicia monitoramento de sua conexão assim como OMAP1 nos passos 4, 5 e 6.

8- Terminal de Acesso se conecta ao Gerente e permite tela de login para usuário digitar "login" e "senha".

9- Quando um usuário digitar "login" e "senha", Terminal de Acesso solicita serviço de login para gerente, informando "login" e "senha".

10- Gerente verifica o tipo de usuário: ADMIN ou DEFAULT.

11- Ao verificar que o tipo de usuário é ADMIN, Gerente contacta o Servidor de Perfil para verificar a senha. Servidor de Perfil informa que a senha confere no sistema.

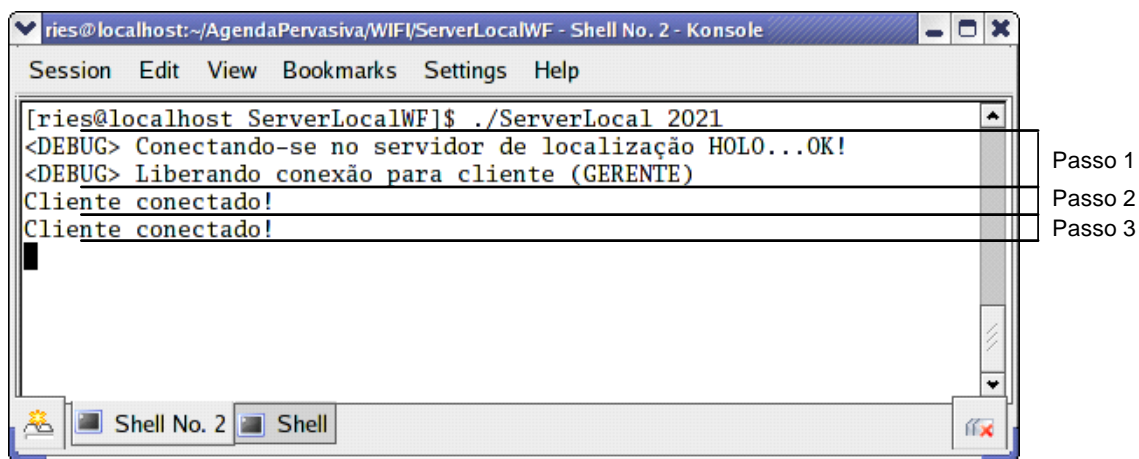


Figura 6.7: Inicialização da aplicação Agenda Pervasiva no componente Servidor de Localização Proxy.

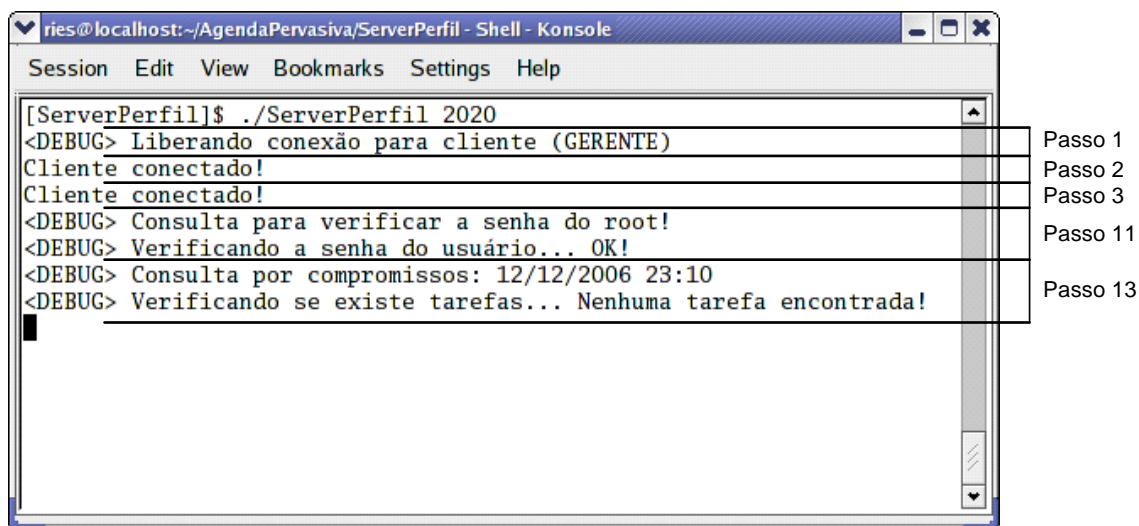


Figura 6.8: Inicialização da aplicação Agenda Pervasiva no componente Servidor de Perfil.

12- Gerente informa para Terminal de Acesso que o usuário é do tipo ADMIN. Isso restringe o usuário a utilizar serviços específicos para ADMIN. Após essa restrição, Gerente espera por solicitação de serviços e Terminal de Acesso espera por comandos do usuário.

13- Ao passar 30 segundos, módulo Relógio do componente Gerente dispara uma consulta ao Servidor de Perfil para verificar a existência de algum compromisso na data e hora atual. O Servidor de Perfil responde ao Relógio que não existe nenhum compromisso.

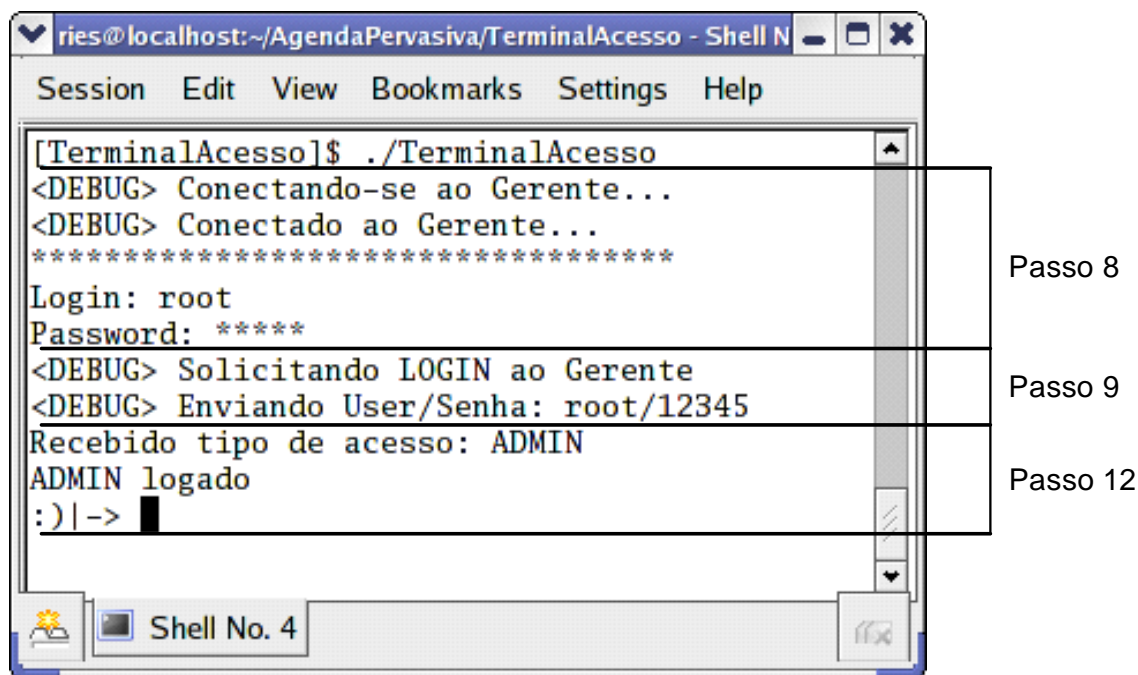


Figura 6.9: Inicialização da aplicação Agenda Pervasiva no componente Terminal de Acesso.

6.2.3 Experimento 2: Avaliação do Funcionamento dos Serviços

Esse experimento visa avaliar a execução de um serviço solicitado por um usuário. Um determinado usuário, no caso o administrador, entra no sistema e solicita a criação de um novo usuário no ambiente.

Assim como o experimento anterior, esse experimento é mostrado através de um conjunto de passos e das telas de execução dos componentes do sistema. As Figura 6.10 mostra a execução do componente Gerente de Aplicação, a Figura 6.11 mostra a execução do componente Servidor de Perfil e a Figura 6.12 mostra a execução do Terminal de Acesso. Essas Figuras são numeradas com passos apresentados a seguir.

- 1- Usuário administrador solicita serviço "createUser" na tela do Terminal de Acesso.
- 2- Terminal de Acesso solicita requisição para criar novo usuário. O envio dessa solicitação é feita utilizando um pacote com 2 campos separados por um token `|—|`. O primeiro campo é utilizado para autenticação do usuário e nesse trabalho seria representado apenas por uma senha "criptografada" do usuário (inversão da senha). O segundo campo representa a solicitação da criação do usuário.
- 3- Gerente, ao receber o pedido para criação do usuário, solicita algumas informações necessárias. O Gerente inicia solicitando o Username. Ao receber a solicitação do Gerente, Terminal de Acesso faz pedido na tela. Usuário digita o campo Username (ries). Terminal de Acesso novamente forma um pacote com 2 campos: o primeiro representa a senha "criptografada" e o campo Username.
- 4- Segue-se o passo 3 para solicitação do nome do usuário (Name).
- 5- Segue-se também o passo 3 para o endereço Bluetooth (ou celular). Como esse serviço ainda não foi implementado, o usuário deve digitar NO nesse campo.

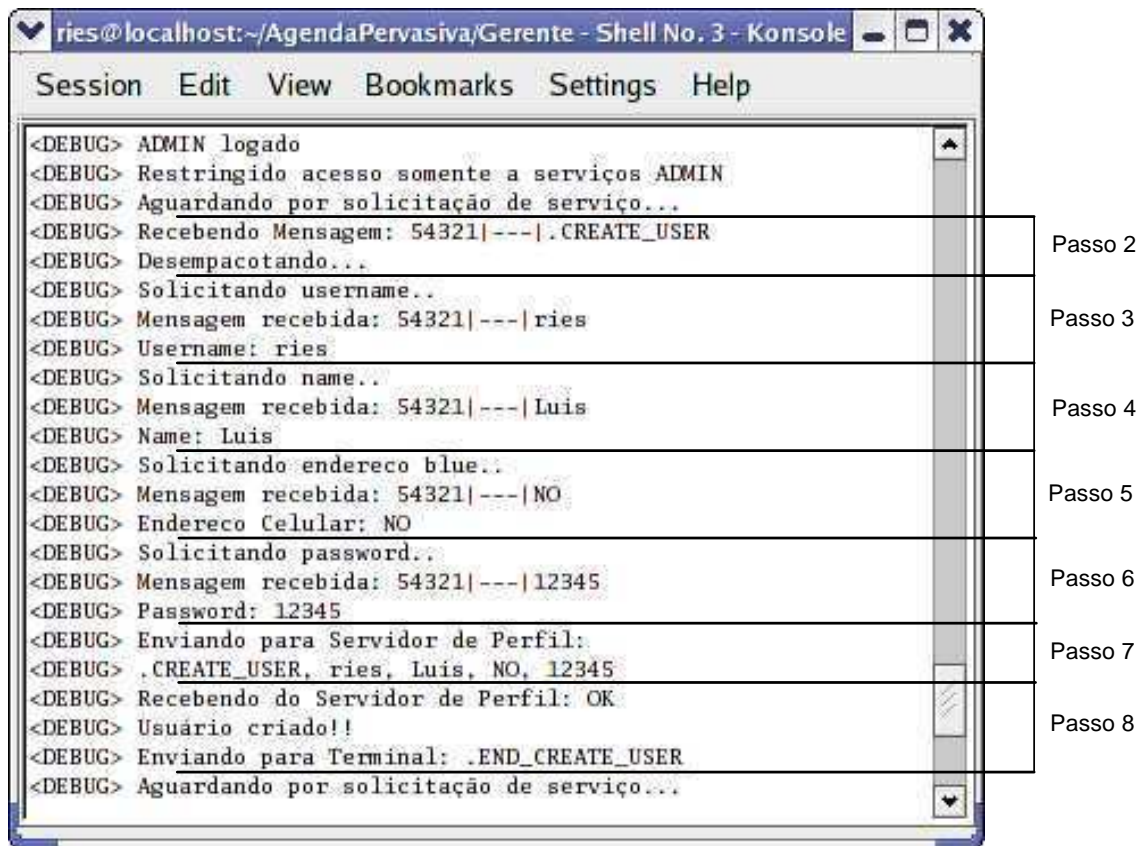


Figura 6.10: Contribuição do componente Gerente para a realização do serviço de criação do usuário.

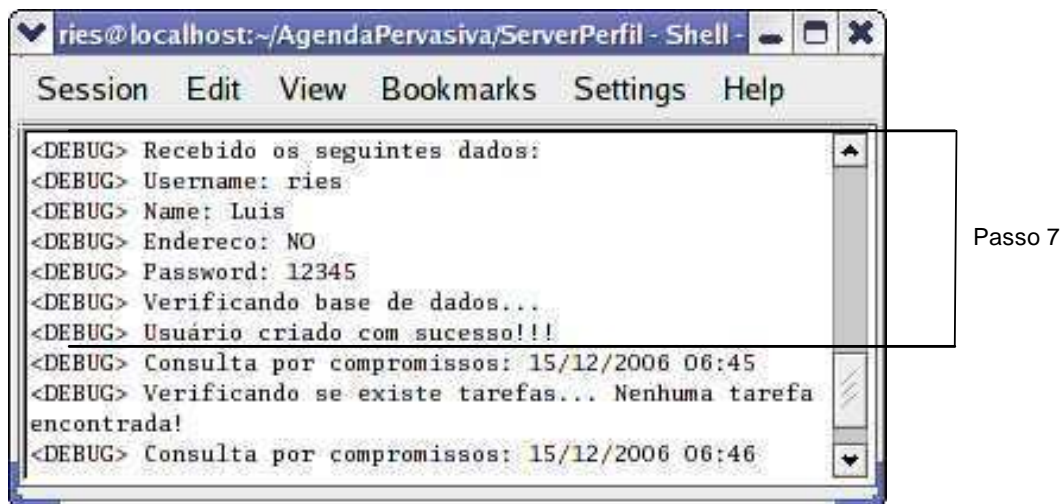


Figura 6.11: Contribuição do componente Servidor de Perfil para a realização do serviço de criação do usuário.

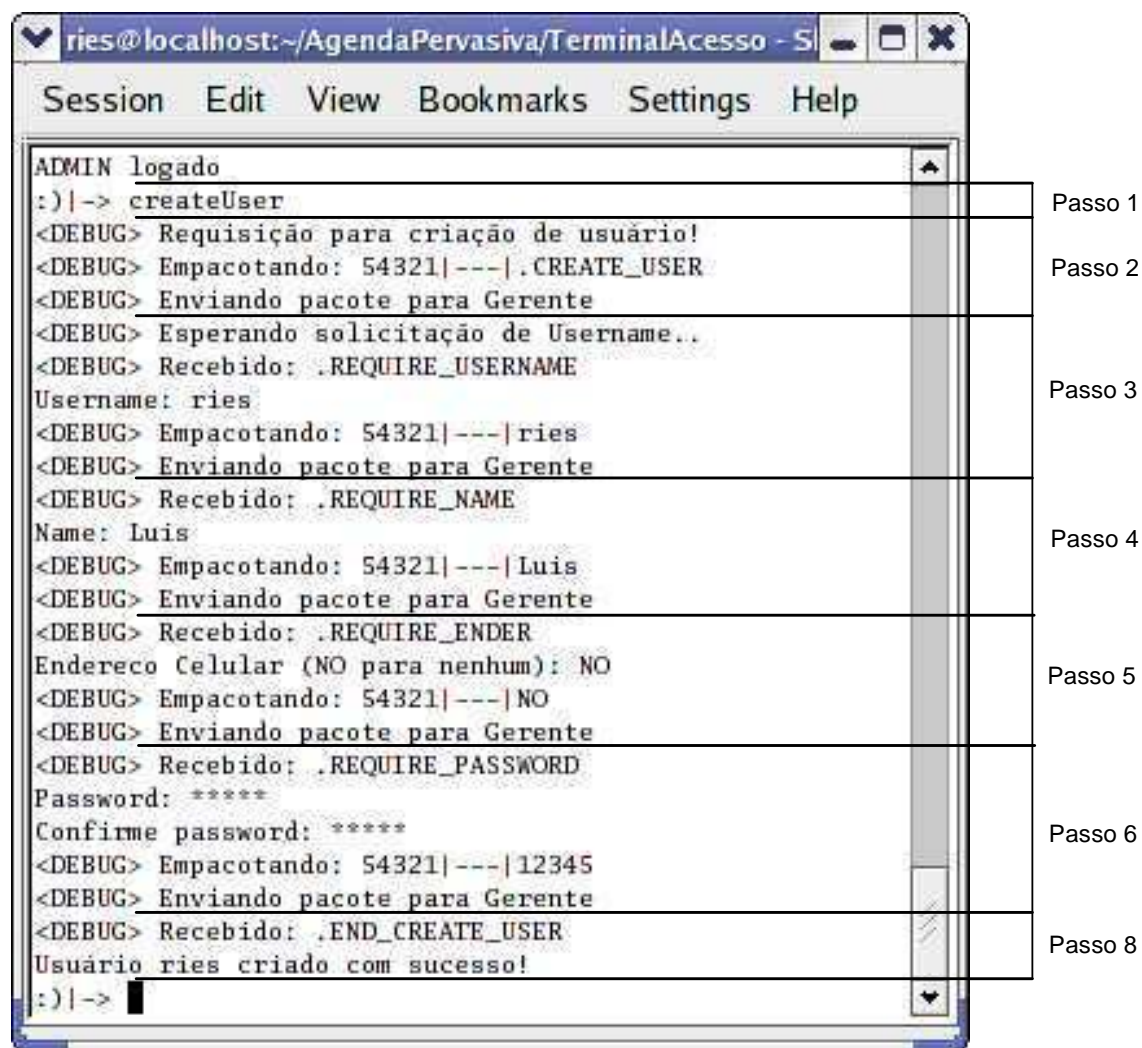


Figura 6.12: Contribuição do componente Terminal de Acesso para a realização do serviço de criação do usuário.

6- Segue-se, também para a senha (Password), o passo 3. No campo senha, é necessário que o usuário digite duas vezes a senha (a segunda para confirmação).

7- Gerente envia a solicitação de criação do usuário com todos os campos necessários para Servidor de Perfil. O Servidor de Perfil consulta a base de dados a fim de evitar redundância de usuários. Em seguida, o Servidor de Perfil cria o usuário na base de dados e envia confirmação ao Gerente

8- Por fim, o Gerente envia para Terminal de Acesso a confirmação da criação do usuário. Gerente novamente espera por solicitação de serviços e Terminal de Acesso espera por comandos do usuário.

6.2.4 Experimento 3: Avaliação do Funcionamento da aplicação Agenda Pervasiva

O último e mais importante experimento mostra o funcionamento da aplicação: a verificação dos horários dos compromissos; a constatação na realização das tarefas e a execução da plataforma pervasiva para o compromisso realizado. Assim como os demais experimentos, uma série de passos e as telas de execução dos componentes do sistema representam esse experimento.

As telas de execução dos componentes são mostrados nas Figuras 6.13, 6.14, 6.15, e 6.16. Os componentes necessários para a execução desse experimento foram: plataforma pervasiva que emite a mensagem sonora para execução da aplicação (Figura 6.13), Gerente da Aplicação (Figura 6.14), Servidor de Localização (Figura 6.15) e Servidor de Perfil (Figura 6.16). As figuras possuem uma numeração relacionada com passos apresentados a seguir.

```

ries@localhost:~/AgendaPervasiva/apl_omap - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help
<DEBUG> Enviando IP da plataforma: localhost
<DEBUG> Registro realizado
<DEBUG> Criando novo serviço(nome,local,par):
-> (tocamusic,</bin/tocamusic>,1)
<DEBUG> Enviando para Gerente pedido de registro de serviço!
<DEBUG> Enviando plataforma: OMAP1
<DEBUG> Enviando serviço: tocamusic
<DEBUG> Registro de serviço realizado
<DEBUG> Iniciando monitoramento da conexão...
<DEBUG> Recebendo solicitação para execução de serviço.
<DEBUG> Serviço: tocamusic - a.au
<DEBUG> Executando Serviço tocamusic
<DEBUG> Executando: /bin/tocamusic a.au
cp a.au /dev/dsp
<DEBUG> Serviço Finalizado
<DEBUG> Enviando confirmação de execução de serviço
<DEBUG> Iniciando monitoramento da conexão...
  
```

Passo 10

Passo 11

Passo 12

Figura 6.13: Funcionamento do componente OMAP1 para a execução da Agenda Pervasiva.

1- Relógio do Gerente verifica a cada 30 segundos se existe um novo compromisso. Para isso, Relógio compara o relógio do sistema com todos compromissos do Servidor de Perfil. Caso não exista, o Relógio recomeça a espera por 30 segundos.

2- Caso exista um compromisso (ex. 001) no horário exato e se o mesmo não foi realizado, o Relógio do Gerente solicita ao Servidor de Perfil o usuário relacionado a esse compromisso. O Servidor de Perfil faz a consulta e retorna o usuário (User1) ao relógio.

3- Em seguida, o Relógio do Gerente faz a solicitação ao Servidor de Perfil a Sala onde é realizada o compromisso. Servidor de Perfil consulta a base de dados e retorna ao Relógio a sala (no caso, Sala202).

4- Para verificar se o usuário (User1) está na sala desejada (no caso, do compromisso 001 ou Sala202), Relógio do Gerente solicita a localização do usuário para Servidor de Localização Proxy.

5- Servidor de Localização Proxy começa a empacotar pedido para conseguir comunicação com Servidor de Localização Holo. Então, Servidor de Localização Proxy faz a solicitação para

```

ries@localhost:~/AgendaPervasiva/Gerente - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
<DEBUG> Aguardando por solicitação de serviço...
<DEBUG-Relogio> Consulta ao Servidor Perfil:
<DEBUG-Relogio> Date: 30/11/2006
<DEBUG-Relogio> Time: 19:44
<DEBUG-Relogio> Verificando se existe algum compromisso pendente...
<DEBUG-Relogio> Nenhum compromisso pendente!!
<DEBUG-Relogio> Consulta ao Servidor Perfil:
<DEBUG-Relogio> Date: 30/11/2006
<DEBUG-Relogio> Time: 19:45
<DEBUG-Relogio> Verificando se existe algum compromisso pendente...
<DEBUG-Relogio> Compromisso pendente: 001
<DEBUG-Relogio> Solicitando usuário do compromisso 001 ao servidor perfil..
.
<DEBUG-Relogio> Usuário do compromisso: User1
<DEBUG-Relogio> Solicitando local do compromisso 001 ao servidor perfil...
<DEBUG-Relogio> Local do compromisso: Sala202
<DEBUG-Relogio> Verificando localização do usuário e comparando com sala 00
1
<DEBUG-Relogio> -> User1 está na sala Sala202
<DEBUG-Relogio> Tarefa 001 está sendo realizada
<DEBUG-Relogio> Música a ser tocada: a.au
<DEBUG-Relogio> Endereço IP da sala Sala202: localhost
<DEBUG-Relogio> Verificando serviço tocamusic na lista de dispositivos... O
K
<DEBUG-Relogio> Enviando solicitação de serviço para localhost: .EXECUTE_SE
RVICE|---|tocamusic|---|a.au
<DEBUG-Relogio> Abrindo conexão com OMAP.
<DEBUG-Relogio> Recebido confirmação - Serviço foi executado!
<DEBUG-Relogio> Fechando conexão com OMAP.
<DEBUG-Relogio> Consulta ao Servidor Perfil:
<DEBUG-Relogio> Date: 30/11/2006
<DEBUG-Relogio> Time: 19:46
<DEBUG-Relogio> Verificando se existe algum compromisso pendente...
<DEBUG-Relogio> Nenhum compromisso pendente!!

```

Passo 1

Passo 2

Passo 3

Passo 4

Passo 7

Passo 8

Passo 9

Passo 10

Passo 11

Passo 12

Figura 6.14: Funcionamento do componente Gerente para a execução da Agenda Pervasiva.

localizar o usuário (User1).

6- Servidor de Localização Proxy aguarda a resposta do Servidor de Localização Holo. Quando receber, ele desempacota retirando a informação sala (Sala202) do pacote.

7- A informação sala (Sala202) é enviada para o módulo Relógio. Esse módulo verifica se o usuário está realizando o compromisso, comparando a informação sala adquirida com o Servidor de Perfil com a informação sala adquirida com o Servidor de Localização.

8- Ao averiguar que o usuário está realizando o compromisso, o Relógio do Gerente busca qual a mensagem sonora que deve ser tocada na sala. Por isso, ele faz a solicitação ao Servidor de Perfil e este retorna a mensagem.

9- Com a finalidade de enviar a mensagem sonora para a sala desejada, o Relógio busca o endereço IP da plataforma pervasiva existente (ex. OMAP1) na sala através de sua lista de dispositivos, assim como o serviço a ser realizado.

10- O módulo Relógio solicita serviço ao OMAP1, informando o nome do serviço e os parâmetros desejados (no caso o nome do arquivo da mensagem).

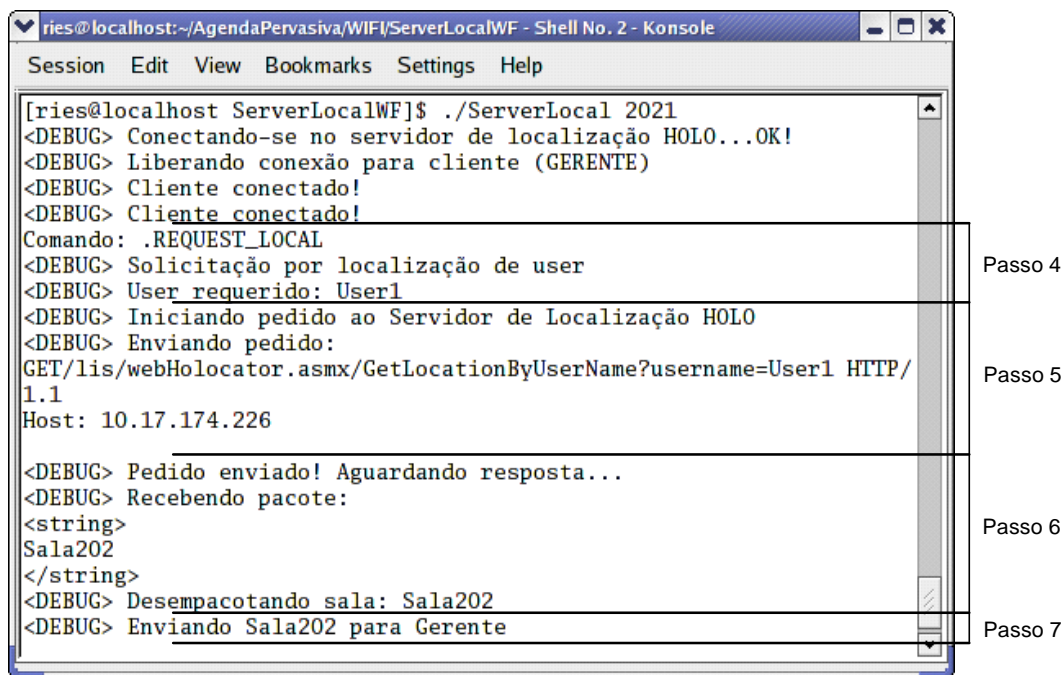


Figura 6.15: Funcionamento do componente Servidor de Localização para a execução da Agenda Pervasiva.

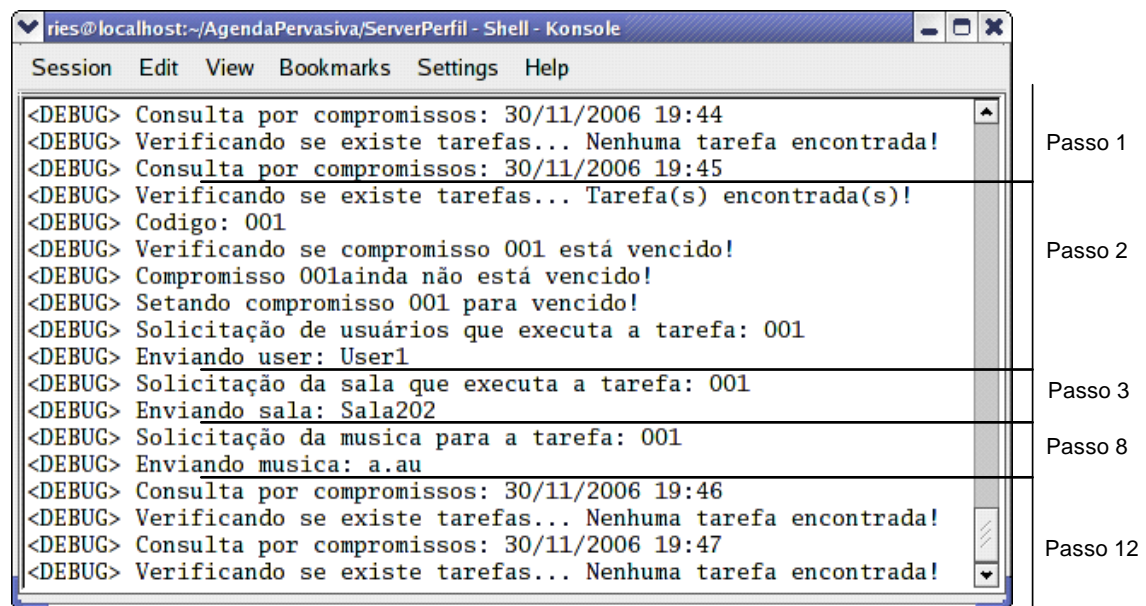


Figura 6.16: Funcionamento do componente Servidor de Perfil para a execução da Agenda Pervasiva.

11- Na solicitação, o Relógio abre conexão com o OMAP1 e faz o requerimento ao mesmo. O OMAP1 faz a execução do serviço, verificando o arquivo executável responsável pelo serviço e informando os parâmetros.

12- Ao finalizar o serviço, OMAP1 envia confirmação da execução ao Relógio e inicia novo monitoramento. Relógio fecha conexão com OMAP1 e novamente inicia a espera por 30 segundos.

6.3 A Contribuição da Plataforma para Aplicação e Utilização de outros Recursos

A aplicação Agenda Pervasiva tem como objetivo demonstrar o funcionamento da plataforma de modo prático, analisando o tratamento dos requisitos analisados no capítulo 2 e verificando as funcionalidades dos componentes do middleware. Além disso, é importante analisar o funcionamento da proposta de integrar um dispositivo na rede pervasiva e fornecer os recursos do mesmo para todos dispositivos da rede.

Por isso, essa seção faz uma breve análise das contribuições que a plataforma fornece para a aplicação pervasiva. Outros tópicos relacionados aos princípios e aos requisitos para a computação pervasiva são mencionados nessa seção.

6.3.1 Middleware

Na aplicação Agenda Pervasiva, permitiu a análise de cinco componentes do Middleware: Módulo de Comunicação, Gerenciador de Serviços, Sistema de Autenticação e Controle de Acesso, Auxiliar de Contexto e Monitor. A análise de cada um dos componentes segue abaixo:

- **Módulo de Comunicação:** esse módulo abstrai toda comunicação, permitindo simples APIs para os dispositivos externos a subrede se comunicarem com a plataforma pervasiva facilmente e vice-versa.
- **Gerenciador de Serviços:** esse módulo é importante para fornecer o serviço tocar a mensagem musical na sala. Ou seja, a plataforma pervasiva informa o seu serviço a toda rede junto com a localização. Além disso, a plataforma pervasiva permite a solicitação de um serviço facilmente. Por exemplo, a plataforma pervasiva é solicitada pelo Gerente da Aplicação para a execução de um serviço tocamusic e ela trata dos recursos, no caso caixa de som, necessários para a execução.
- **Sistema de Autenticação e Controle de Acesso:** permite somente usuários autorizados a cadastrar no sistema. Faz a distinção do usuário default do administrador. O administrador pode solicitar serviços para cadastrar novos usuários, novas salas e verificar o funcionamento do sistema, como por exemplo, localizar um usuário. Já o usuário default só é capaz de acessar informações relativas a seu respeito, como suas informações pessoais, suas tarefas e sua localização. A privacidade é um elemento importante na computação pervasiva. Por isso, o administrador e os outros usuários existentes não têm acesso a um determinado usuário.
- **Auxiliar de Contexto:** na aplicação, somente um tipo de contexto foi tratado: localização do usuário. No entanto, a detecção desse contexto (ou a localização) foi elaborada de duas maneiras (vistas anteriormente). A aplicação auxiliou a demonstrar como a plataforma trata os dois tipos de contexto: dinâmico e estático. Nós especificamos o contexto dinâmico como a localização por bluetooth, onde o OMAP é o responsável por detectar o contexto, armazenar as informações relativas e caso solicitado, enviar aos demais. Já a localização

por WIFI foi caracterizada por ser um tipo de contexto estático. Isso se deve, pois varia em uma determinada fatia de tempo e quando esse contexto é tratado em um servidor específico.

- Monitor: esse módulo é responsável por permitir ao administrado o acesso às informações contidas na aplicação e por fornecer serviços de testes e avaliação (como é o caso da inserção de dados nos servidores através de arquivos - loadPerfilXML).

6.3.2 Escalabilidade: adição de novas salas

A adição de novas salas para o ambiente da Agenda Pervasiva pode ser realizado de maneira fácil. Devido ao seu mecanismo de inicialização e ao módulo Discovery do Gerenciador de Serviços, a adição de novas salas no ambiente é possível apenas conectando uma plataforma pervasiva na rede principal. Ao conectar, a plataforma pervasiva se comunica com Gerente e faz seu registro, assim como o registro de seus serviços.

6.3.3 Proxy para tratar a heterogeneidade

O tratamento da heterogeneidade entre a aplicação e o sistema de localização da Mobilab foi feito de maneira simples: disponibilizamos um Proxy no servidor onde fica armazenado o Gerente de Aplicação. Esse Proxy, que é chamado de Servidor de Localização WIFI, funciona no sistema para permitir a comunicação da aplicação Agenda Pervasiva (TCP/IP, utilizando sockets) com o sistema de localização da Unisinos que utiliza Webservices para se comunicar.

Outra maneira, ainda não disponibilizada, é a utilização da plataforma para permitir a comunicação entre diferentes protocolos e funcionando como um gateway. Na implementação não foi adicionada a comunicação via Bluetooth. No entanto, se adicionássemos essa comunicação no sistema, poderíamos permitir a comunicação de dispositivos heterogêneos (telefone celular bluetooth com o restante do ambiente, por exemplo).

Capítulo 7

Conclusão e Trabalhos Futuros

Este capítulo revisa os quesitos de pesquisa considerados no desenvolvimento deste trabalho, ressalta as principais contribuições deste mesmo e caracteriza as oportunidades futuras propiciadas na evolução desse trabalho.

7.1 Conclusão

A computação pervasiva é o novo cenário computacional que vem sendo considerado o mais recente paradigma computacional desse século. A mobilidade de diversos grupos de pesquisa, tanto acadêmicos quanto industriais, para a computação pervasiva, vem crescendo em escala global. Esse novo cenário computacional contempla o desenvolvimento de aplicações com novas funcionalidades em ambientes inteligentes, como por exemplo, uma casa inteligente. A computação pervasiva disponibiliza ao usuário acesso computacional de modo invisível e onipresente.

A construção de aplicações e ambientes pervasivos, no entanto, proporciona muitos desafios. Um desses desafios é tratar da heterogeneidade de diversos dispositivos em uma rede. Por isso, o presente trabalho apresentou a implementação de uma plataforma que seja capaz de adaptar um ou mais dispositivos embarcados a um determinado ambiente pervasivo. Essa plataforma pode ser facilmente integrada em uma rede pervasiva e pode fornecer recursos e serviços de um ou mais dispositivos embarcados a outros dispositivos existentes no ambiente conectados a rede pervasiva.

Nesse trabalho, foram estudados alguns conceitos relacionados à computação pervasiva e ao trabalho realizado. O conceito de ambientes, aplicações, dispositivos e redes pervasivas foram mostrados para visualizar as diferenças com a computação tradicional. Dois conceitos foram importantes na definição do trabalho. O conceito utilizado para arquitetura pervasiva foi aplicado para todos os dispositivos pervasivos existentes no ambiente, enquanto o conceito de plataforma é baseado em um dispositivo computacional genérico capaz de permitir o reuso de aplicações em software.

Outro aspecto importante para o trabalho foi a captura dos requisitos para a computação pervasiva. Os requisitos necessários para o desenvolvimento de um ambiente pervasivo foram estudados a fim de auxiliar na concepção da plataforma pervasiva proposta no trabalho. Esses requisitos também foram utilizados para o estudo relacionado a arquiteturas e plataformas pervasivas. O estudo de arquiteturas pervasivas foi importante para avaliar as necessidades que os ambientes inteligentes necessitam, em termos de serviços e recursos. O estudo relacionado às plataformas da computação pervasiva auxilia na concepção do modelo da plataforma proposta, fornecendo uma visão geral dos componentes HW e SW necessários.

Com base nas pesquisas realizadas sobre o estudo das arquiteturas pervasivas, foi definido uma arquitetura para o trabalho. Essa arquitetura pervasiva foi utilizada para a captura dos serviços e componentes necessários para a modelagem da plataforma. Essa arquitetura foi baseada em um modelo em camadas, onde em cada camada foram abordados os componentes e sua importância no ambiente pervasivo.

Analisando os requisitos capturados, os estudos realizados sobre plataformas pervasivas e as camadas da arquitetura pervasiva, definiu-se o modelo de plataforma pervasiva para integrar dispositivos eletro-eletrônicos no ambiente pervasivo. Primeiramente realizou-se um modelo da plataforma, definindo o funcionamento no sistema como um todo. Em seguida, foi realizado o projeto da plataforma, com base no modelo em camadas, definindo os principais componentes para a mesma plataforma. Com base nessa modelagem e nesse projeto, a plataforma foi implementada e as principais camadas foram definidas detalhadamente.

A plataforma proposta foi avaliada utilizando uma aplicação voltada para um ambiente inteligente. A aplicação Agenda Pervasiva consiste em uma agenda eletrônica distribuída no sistema capaz de localizar o usuário e verificar se os compromissos do mesmo foram realizados com sucesso. Esse ambiente foi executado com base em dois tipos de infra-estrutura de localização: WIFI, onde efetua uma localização precisa de um dispositivo WIFI do usuário (desenvolvida por [51]) e Bluetooth, que busca identificar o usuário através de um dispositivo Bluetooth. Para avaliar os resultados obtidos, a execução do sistema foi capturada e analisada passo a passo. Por fim, foi apresentada a contribuição da plataforma e de outros recursos para a aplicação pervasiva.

7.2 Contribuições de Pesquisa

Esse trabalho contribui principalmente para a proposição de uma plataforma pervasiva que facilita a integração de um ou mais dispositivos ou periféricos em uma rede pervasiva. Essa plataforma é capaz de diferentes tipos de aplicações em diferentes ambientes e cenários pervasivos. Outras atividades secundárias nesse trabalho foram:

- Conceitos gerais e aplicados à computação pervasiva;
- Análise de requisitos básicos para desenvolver aplicações pervasivas;
- Estudo relacionado sobre arquiteturas pervasivas, assim como sua contribuição para o trabalho;
- Estudo sobre plataformas pervasivas, definindo sua especificação, suas contribuições e suas limitações com relação a esse trabalho;
- Projeto de uma arquitetura pervasiva, apresentando um modelo em camadas e definindo cada componente dessa arquitetura;
- Modelagem de uma plataforma pervasiva que facilita a integração de diversos dispositivos heterogêneos em uma rede pervasiva;
- Definição e projeto de componentes voltados para o tratamento de requisitos relacionados à computação pervasiva;
- Desenvolvimento de uma aplicação prática para ambientes pervasivos, relacionando alguns requisitos a computação pervasiva. No entanto essa aplicação encontra-se em estágio inicial;

7.3 Trabalhos Futuros

A plataforma pervasiva proposta no trabalho propiciou uma certa variedade de trabalhos futuros. São eles:

- Estudar, analisar e desenvolver componentes e serviços mais específicos que permitam um tratamento mais eficaz para a computação pervasiva, tais como tratamento de falhas, métodos mais eficientes para segurança, comunicação entre dispositivos totalmente heterogêneos, etc;
- Construir frameworks e agentes de aplicação para facilitar o desenvolvimento de aplicações na plataforma pervasiva;
- Proporcionar serviços de roteamento para a plataforma pervasiva, com o objetivo de estender a proposta do trabalho;
- Testar, avaliar e analisar a plataforma para verificar suas funcionalidades pervasivas e a flexibilidade próximo trabalho;

Referências Bibliográficas

- [1] J. Wu and I. Stojmenovic, "Guest editors' introduction: Ad hoc networks," *IEEE Computer Society*, vol. 37, no. 2, pp. 29–31, February 2004.
- [2] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Personal Communication*, vol. 8, no. 4, pp. 10–17, August 2001.
- [3] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *IEEE Computer Society*, vol. 36, no. 3, pp. 25–31, March 2003.
- [4] J. Krikke, "T-engine: Japan's ubiquitous computing architecture is ready for prime time," *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 4–9, April-June 2005.
- [5] M. Weiser, "The computer for the twenty-first century." *Scientific American*, vol. 265, no. 3, pp. 94–104, September 1991.
- [6] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, "The gator tech smarthouse: A programable pervasive space," *IEEE Computer Society*, vol. 38, no. 3, pp. 50–60, March 2005.
- [7] M. Miller, *Descobrimdo bluetooth*. Rio de Janeiro: Campus, 2001, 289 p.
- [8] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, September 1997.
- [9] U. Varshney, "Vehicular mobile commerce," *IEEE Computer Society*, vol. 37, no. 12, pp. 116–118, December 2004.
- [10] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "A middleware infrastructure for active spaces," *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 74–83, October-December 2002.
- [11] Y. Shi, W. Xie, G. Xu, R. Shi, E. Chen, Y. Mao, and F. Liu, "The smart classroom: Merging technologies for seamless tele-education," *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 47–55, April-June 2003.
- [12] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, 1999, pp. 304–307.
- [13] S. Voidsa, E. Mynatt, B. MacIntyre, and G. Corso, "Integrating virtual and physical context to support knowledge workers," *IEEE Pervasive Computing*, vol. 1, no. 3, pp. 73–79, July-September 2002.

- [14] B. Johanson, A. Fox, and T. Winograd, "The interactive workspaces project: Experiences with Ubiquitous Computing Rooms," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 67–74, April-June 2002.
- [15] M. Fleck, M. Frid, T. Kindberg, E. O'Brien-Strain, R. Rajani, and M. Spasojevic, "From informing to remembering: Ubiquitous systems in interactive museums," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 13–21, April-June 2002.
- [16] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "Easyliving: Technologies for intelligent environments," in *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, 2000, pp. 12–29.
- [17] P. Kourouthanassis and G. Roussos, "Developing consumer-friendly pervasive retail systems," *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 32–39, April-June 2003.
- [18] J. E. Bardram, H. B. Christensen, and A. K. Olsen, "Activity-driven computing infrastructure-pervasive computing in healthcare," Center for Pervasive Computing, University of Aarhus, Aarhus, Denmark, Tech. Rep., October 2004, cfPC-2004-PB-24.
- [19] U. Hansmann, L. Merk, M. Nicklous, and T. Stober, *Pervasive computing handbook*. Berlin: Springer, 2001, 409 p.
- [20] D. Anderson, *Firewire(R) system architecture: IEEE 1394A*, 2nd ed. Boston: Addison-Wesley, 1999, 544p.
- [21] Echelon Corp., "The lonstalk protocol," 2002, capturado em <http://www.echelon.com/developers/lonworks/protocol/default.htm>, Novembro 2005.
- [22] A. Tanenbaum and M. Steen, *Distributed Systems: Principles and Paradigms*, 1st ed. New Jersey: Prentice Hall, 2002, 803 p.
- [23] K. Henriksen, J. Indulska, and A. Rakotonirainy, "Infrastructure for pervasive computing: Challenges," in *GI Jahrestagung (1)*, 2001, pp. 214–222.
- [24] A. C. Yamin, "Arquitetura para um ambiente de grade computacional direcionado às aplicações móveis, distribuídas e conscientes do contexto da computação pervasiva," Tese de Doutorado, UFRGS, 2004, 194p.
- [25] R. Litiu and A. Prakash, "Dacia: a mobile component framework for building adaptive distributed applications," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 2, pp. 31–42, April 2001.
- [26] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, "Project aura: Toward distraction-free pervasive computing," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22–31, April-June 2002.
- [27] S. W. Han, Y. B. Yoon, H. Y. Youn, and W. Cho, "A new middleware architecture for ubiquitous computing environment," in *2nd IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS 2004)*, 2004, pp. 117–121.
- [28] D. Bonatto, J. Barbosa, G. Cavalheiro, and J. Ramos, "Pholo: uma arquitetura para computação pervasiva utilizando o holoparadigma," in *VI Workshop em Sistemas de Alto Desempenho (WSCAD)*, 2005, capturado em http://www.inf.unisinos.br/~barbosa/textos/WSCAD_2005.pdf, Junho 2006, 8p.

- [29] J. L. Barbosa, “Holoparadigma: um modelo multiparadigma orientado ao desenvolvimento do software distribuído,” Tese de Doutorado, UFRGS, 2002, 213p.
- [30] H. Gellersen, G. Kortuem, A. Schmidt, and M. Beigl, “Physical prototyping with smart-its,” *IEEE Pervasive Computing*, vol. 3, no. 3, pp. 74–82, July-September 2004.
- [31] A. Tanenbaum, *Redes de computadores*. Rio de Janeiro: Campus, 1997, 923 p.
- [32] D. Valtchev and I. Frankov, “Service gateway architecture for a smart home,” *IEEE Communications Magazine*, vol. 40, no. 4, pp. 126–132, April 2002.
- [33] J. H. Park, M. J. Lee, and S. J. Kang, “Corba-based distributed and replicated resource repository architecture for hierarchically configurable home network,” *Journal of Systems Architecture*, vol. 51, no. 2, pp. 125–142, February 2005.
- [34] M. Henning and S. Vinoski, *Advanced CORBA® Programming with C++*. Boston: Addison Wesley, 1999, 1120 p.
- [35] J. Beutel, O. Kasten, F. Mattern, K. Römer, F. Siegemund, and L. Thiele, “Prototyping wireless sensor network applications with bnodes,” in *First European Workshop on Wireless Sensor Networks (EWSN 2004)*, 2004, pp. 323–338.
- [36] T. Knott, “Smart surrogates,” *BP Frontiers magazine*, vol. 1, no. 9, pp. 6–10, April 2004.
- [37] Global Inventions Inc., “Zigbee alliance home page,” 2004, capturado em: <http://www.zigbee.org>, Outubro 2006.
- [38] D. Gay, P. Levis, and D. Culler, “Software design patterns for tinyOS,” in *Proceedings of the 2005 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools For Embedded Systems (LCTES '05)*, 2005, pp. 40–49.
- [39] R. Ballagas, M. Ringel, M. Stone, and J. Borchers, “Istuff: a physical user interface toolkit for ubiquitous computing environments,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*, 2003, pp. 537–544.
- [40] M. Satyanarayanan, “Home page of aura,” 2002, capturado em <http://www.cs.cmu.edu/aura/auravideo.mpg>, April 2006.
- [41] J. C. Lee, D. Avrahami, S. E. Hudson, J. Forlizzi, P. H. Dietz, and D. Leigh, “The calder toolkit: wired and wireless components for rapidly prototyping interactive devices,” in *Proceedings of the 2004 Conference on Designing interactive Systems: Processes, Practices, Methods, and Techniques (DIS '04)*, 2004.
- [42] S. Greenberg and C. Fitchett, “Phidgets: easy development of physical interfaces through physical widgets,” in *Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology (UIST '01)*, 2001, pp. 209–218.
- [43] L. H. Ries and R. Michelin, “Geração de um sistema operacional dedicado para sistemas embarcados,” Trabalho de Conclusão de Curso, PUCRS, 2004, 86p.
- [44] A. Technologies, “Smartcard OS,” 2005, capturado em: http://www.acmet.com/html/smartcard_os.html.
- [45] OMG Inc., “Specialized corba specifications,” 2006, capturado em http://www.omg.org/technology/documents/specialized_corba.htm, Julho 2006.

- [46] G. Judd and P. Steenkiste, "Providing contextual information to pervasive computing applications," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., July 2002, cMU-CS-02-154.
- [47] T. Instruments, "Wireless solutions - omap platform," 2005, capturado em <http://www.ti.com/omap>, Fevereiro 2006.
- [48] K. Yaghmour, *Building Embedded linux systems*. Beijing: O'Reilly, 2003, 391 p.
- [49] R. Landley, "Busybox," 1999, capturado em <http://www.busybox.net>, Julho 2006.
- [50] C. Smith, "Linux nfs-howto," 2006, capturado em <http://nfs.sourceforge.net/nfs-howto>, Julho 2006.
- [51] C. Rolim, "HLS: um modelo para suporte à sistemas de localização no holoparadigma," Dissertação de Mestrado, UNISINOS, 2007, 87p.