

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ**

Eduardo Germano da Silva

**DESENVOLVIMENTO DE UM SISTEMA PARA
ADAPTAÇÃO DE CONTEXTO EM AUTOMAÇÃO
RESIDENCIAL**

Araranguá, julho de 2013.

Eduardo Germano da Silva

**DESENVOLVIMENTO DE UM SISTEMA PARA
ADAPTAÇÃO DE CONTEXTO EM AUTOMAÇÃO
RESIDENCIAL**

**Trabalho de Conclusão de
Curso submetido à Universi-
dade Federal de Santa Cata-
rina, como parte dos requisitos
necessários para a obtenção do
Grau de Bacharel em Tecno-
logias da Informação e Comu-
nicção.**

Araranguá, julho de 2013.

Eduardo Germano da Silva

**DESENVOLVIMENTO DE UM SISTEMA PARA
ADAPTAÇÃO DE CONTEXTO EM AUTOMAÇÃO
RESIDENCIAL**

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de “Bacharel em Tecnologias da Informação e Comunicação”, e aprovado em sua forma final pela Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, julho de 2013.

Prof. Dr. Vilson Gruber
Coordenador do Curso

Banca Examinadora:

Prof. Dr. Anderson Luiz Fernandes Perez
Orientador

Prof. Dr. Alexandre Leopoldo Gonçalves

Prof. Dr. Ricardo Alexandre Reinaldo de Moraes

Aos meus pais, Edson Fortes e Valdete Germano.

AGRADECIMENTOS

Ao meu orientador, o professor Anderson Luiz Fernandes Perez, que como um pai foi paciente e me apoiou, acreditando em mim até nos momentos que nem eu mesmo acreditava. À minha namorada Tatiana, que mesmo nos momentos de adversidade compreendeu minhas escolhas e não me abandonou. À Universidade Federal de Santa Catarina. Aos parceiros do LARM que de alguma forma participaram dessa jornada, especialmente ao Elder Tramontin e Renan Darós, que foram essenciais para a conclusão deste trabalho. À minha família e amigos que se fizeram presentes nos momentos mais íngremes. Aos professores da banca pelas contribuições.

*O Filho da empregada também vai virar
Doutor.*

Eduardo Germano

RESUMO

A automação residencial é uma tendência em novos empreendimentos imobiliários, pois traz benefícios como segurança, acessibilidade e eficiência energética. Na automação residencial tradicional o processo de automação se dá de maneira reativa, ou seja, é realizada quando uma pessoa, ocupante da residência, aciona um determinado comando. Entretanto, a automação sensível ao contexto aumenta o grau de acessibilidade de uma residência, pois permite que o ambiente seja um agente proativo e aja de maneira inteligente, se antecipando as diversas situações com soluções que beneficiam seus usuários. Este trabalho descreve o desenvolvimento de um sistema distribuído para monitorar os ambientes de uma residência, configurando, gerenciando e adequando o contexto de acordo com as preferências do usuário. O sistema permite que o usuário controle totalmente a residência, por meio de interfaces de comunicação locais e remotas, é baseado em *hardware* de dimensões reduzidas, permitindo que este execute de modo pervasivo. A eficiência e a confiabilidade do sistema foram comprovadas em três diferentes experimentos realizados em bancada.

Palavras-chave: automação residencial, computação ubíqua, adaptação de contexto, *hardware* livre.

ABSTRACT

The home automation is a trend in new buildings, it brings benefits such as security, accessibility and energy efficiency. Traditional home automation process happens in a reactive, that is performed when a person occupying the residence, triggers a particular command. However, the context-sensitive automation increases the degree of accessibility of a residence, because it enables the environment to be a proactive agent and act smartly, anticipating the various situations with solutions that benefit its users. This work describes the development of a distributed system for monitoring environments of a residence, configuring, managing and adapting context according to user preferences. The system allows the user to fully control the residence, through local and remote communication interfaces, hardware is based with reduced dimensions, allowing it to run in a pervasive way. The efficiency and reliability of the system have been proven in three different experiments.

Keywords: home automation, ubiquitous computing, context adaptation, open-source hardware.

LISTA DE FIGURAS

Figura 1	Evolução da modernização dos aparelhos celulares.....	33
Figura 2	Relação entre computação ubíqua, móvel e pervasiva. . .	34
Figura 3	Princípios da computação móvel.	35
Figura 4	Sony <i>SmartWatch</i> ®.....	36
Figura 5	Áreas que compõem a computação pervasiva.	36
Figura 6	Microsoft Kinect®.....	39
Figura 7	Google Glass®.	39
Figura 8	Sala de reuniões da Xerox PARC.....	40
Figura 9	Aware Home.....	41
Figura 10	Adaptive House.....	42
Figura 11	Arquitetura do projeto Gaia.	43
Figura 12	Arquitetura do One.World.	45
Figura 13	Áreas que compõem a domótica.	48
Figura 14	Sensor de umidade e temperatura.	49
Figura 15	IHC.	50
Figura 16	Samsung Galaxy S3®.	51
Figura 17	Arquitetura ABC.	52
Figura 18	Exemplo de uma aplicação X-10.	54
Figura 19	Exemplo de um módulo controlador X-10.	55
Figura 20	Hierarquia de uma rede EIB / KNX.	57
Figura 21	Ilustração de uma típica rede LonWorks.....	58
Figura 22	Exemplo de rede ZigBee.	59
Figura 23	Esquema simplificado da comunicação em malha.	60
Figura 24	Tamanho do <i>chipset</i> Z-Wave comparado com uma moeda de um quarto de dólar.	61
Figura 25	Arquitetura de <i>hardware</i> do sistema.....	63
Figura 26	Raspberry PI modelo B.....	64
Figura 27	Placa de prototipação Arduino UNO.....	66
Figura 28	Arduino UNO com um Ethernet Shield.....	66
Figura 29	Arquitetura de <i>software</i> do sistema.	67
Figura 30	Formato da mensagem de cumprimento.	69
Figura 31	Formato da mensagem de contexto.....	69

Figura 32	Esquemático do circuito de controle do experimento 1..	71
Figura 33	Imagem do primeiro experimento montado no <i>protoboard</i> .	72
Figura 34	Imagem da interface de saída do controle central durante o primeiro experimento.	72
Figura 35	Esquemático do circuito de controle do experimento 2..	73
Figura 36	Imagem do segundo experimento montado no <i>protoboard</i> .	74
Figura 37	Imagem da interface de saída do controle central durante o segundo experimento.	74
Figura 38	Esquemático do circuito de controle do primeiro ambiente.	75
Figura 39	Esquemático do circuito de controle do segundo ambiente.	76
Figura 40	Imagem do primeiro cômodo do terceiro experimento montado no <i>protoboard</i>	76
Figura 41	Imagem do segundo cômodo do terceiro experimento montado no <i>protoboard</i>	77
Figura 42	Imagem da interface de saída do controle central.	77

LISTA DE TABELAS

Tabela 1	Evolução da computação em termos usuário x computador.....	32
Tabela 2	Dimensões da computação ubíqua.....	35
Tabela 3	Lista de canais disponíveis para o experimento.....	74
Tabela 4	Disposição das lâmpadas nos ambientes.....	75

LISTA DE ABREVIATURAS E SIGLAS

LARM	Laboratório de Automação e Robótica Móvel.....	28
UbiComp	<i>Ubiquitous Computing</i>	31
PC	<i>Personal Computer</i>	31
ROM	<i>Read-Only Memory</i>	33
4G	<i>Fourth Generation</i>	35
Wi-Fi	<i>Wireless Fidelity</i>	35
PDA	<i>Personal Digital Assistant</i>	36
AHRI	<i>Aware Home Research Initiative</i>	41
IHC	<i>Intelligent Home Control</i>	50
ABA	Arquitetura Baseada em Automação	51
ABC	Automação Baseada em Comportamento	51
IA	Inteligência Artificial	52
ID3	<i>Iterative Dichotomiser 3</i>	52
TAC	Tecnologias de Automação e Controle	53
TIC	Tecnologias da Informação e Comunicação	53
PLC	<i>Powerline Communication</i>	53
bps	<i>bits por segundo</i>	55
EHS	<i>European Home System</i>	55
EIB	<i>European Installation Bus</i>	55
KNX	Konnex	55
EIBA	<i>European Installation Bus Association</i>	56
ANSI	<i>American National Standards Institute</i>	57
ISO	<i>International Organization for Standardization</i>	57
OSI	<i>Open Systems Interconnection</i>	57
Hz	<i>Hertz</i>	59
RAM	<i>Random Access Memory</i>	64
RCA	<i>Radio Corporation of America</i>	64
SD	<i>Secure Digital</i>	64
USB	<i>Universal Serial Bus</i>	64
HD	<i>Hard Disk</i>	65
HDMI	<i>High-Definition Multimedia Interface</i>	65
SAC	Sistema de Adaptação de Contexto	67

SQL	<i>Structured Query Language</i>	68
PWM	<i>Pulse Width Modulation</i>	72
LED	<i>Light-emitting diode</i>	76
3D	<i>Three Dimensional</i>	80

SUMÁRIO

1 INTRODUÇÃO	25
1.1 OBJETIVOS	26
1.2 JUSTIFICATIVA E MOTIVAÇÃO	26
1.3 METODOLOGIA	28
1.4 ORGANIZAÇÃO DO TRABALHO	28
2 COMPUTAÇÃO UBÍQUA	31
2.1 DEFINIÇÕES E OBJETIVOS	31
2.2 COMPUTAÇÃO MÓVEL	34
2.3 COMPUTAÇÃO PERVASIVA	36
2.4 INTERFACES UBÍQUAS DE COMUNICAÇÃO	37
2.5 RESIDÊNCIAS INTELIGENTES	39
2.6 MIDDLEWARES	43
2.7 DESAFIOS DA COMPUTAÇÃO UBÍQUA	45
3 DOMÓTICA	47
3.1 DEFINIÇÃO E OBJETIVOS	47
3.2 NÍVEIS DE AUTOMAÇÃO RESIDENCIAL	48
3.2.1 Sistemas Autônomos	49
3.2.2 Sistemas Integrados - Central de Automação	50
3.2.3 Sistemas Complexos - Residência Inteligente	50
3.3 ARQUITETURAS PARA AUTOMAÇÃO RESIDENCIAL ..	51
3.3.1 ABA - Arquitetura baseada em Automação	51
3.3.2 ABC - Automação baseada em Comportamento ...	52
3.4 TECNOLOGIAS DE AUTOMAÇÃO E CONTROLE	53
3.4.1 TACs com meio de comunicação definido	53
3.4.2 TACs sem fio	58
4 SISTEMA PARA ADAPTAÇÃO DE CONTEXTO EM AUTOMAÇÃO RESIDENCIAL	63
4.1 ARQUITETURA DE HARDWARE DO SISTEMA DE ADAPTAÇÃO DE CONTEXTO RESIDENCIAL	63
4.1.1 Unidade de Controle Central	64
4.1.2 Unidade de Controle Local	65
4.2 ARQUITETURA DE SOFTWARE DO SISTEMA DE ADAPTAÇÃO DE CONTEXTO RESIDENCIAL	66
4.2.1 Controle Central	67
4.2.2 Controle Local	69
5 AVALIAÇÃO DO SISTEMA	71
5.1 EXPERIMENTO 1 - CONTROLE DE UM VENTILADOR ..	71

5.2 EXPERIMENTO 2 - CONTROLE DE UM TELEVISOR . . .	73
5.3 EXPERIMENTO 3 - CONTROLE DE LUMINOSIDADE . . .	75
6 CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS	79
6.1 PROPOSTAS PARA TRABALHOS FUTUROS	80
REFERÊNCIAS	81

1 INTRODUÇÃO

Com o passar dos anos e o rápido avanço tecnológico é possível perceber que a computação vem evoluindo a ponto dos dispositivos computacionais reduzirem de tamanho e ainda assim aumentarem a capacidade de processamento. Esse avanço tecnológico possibilita o desenvolvimento de inúmeras aplicações para os mais diversos propósitos.

A automação residencial se beneficia deste avanço tecnológico, pois utiliza mecanismos computadorizados que executam de forma automática tarefas que antes eram realizadas pelos usuário.

Essa área da automação traz benefícios para o cotidiano dos usuários em aspectos como segurança, conveniência, acessibilidade e eficiência energética. Essa característica faz com que atualmente a automação residencial seja tendência em novos empreendimentos imobiliários e alvo de estudos em diversos centros de pesquisa do mundo inteiro.

Periféricos como sensores e câmeras permitem monitorar os usuários de uma residência, e assim mapear informações como os hábitos, características e necessidades das pessoas. Essas informações permitem o desenvolvimento de aplicações que fazem uso das características de um ambiente à favor de um usuário, e são denominadas aplicações sensíveis ao contexto.

Na automação residencial tradicional o processo de automação se dá de maneira reativa, ou seja, a automação só é realizada quando uma pessoa, ocupante da residência aciona um determinado comando, entretanto as aplicações sensíveis ao contexto aumentam o grau de acessibilidade de uma residência, pois permitem que o ambiente seja um agente proativo e aja de maneira inteligente, se antecipando as diversas situações com soluções que beneficiam seus usuários.

Residências que utilizam aplicações sensíveis ao contexto são denominadas *smart-homes*, e não visam somente ampliar o conforto e a segurança de seus usuários, mas sim a qualidade de vida em todos os aspectos. Essa característica, aliada com o aumento da expectativa de vida das pessoas, faz com que as *smart-homes* sejam ícones importantes em cenários contemporâneos e futurísticos.

Permitir que uma residência reconheça seus usuários e adapte seus ambientes internos de acordo com as preferências destes, movendo com ele todo seu contexto computacional, cômodo por cômodo, por meio de um sistema eficiente e confiável, é o principal objetivo e contribuição deste trabalho.

1.1 OBJETIVOS

Esta sessão apresenta o objetivo geral e os objetivos específicos deste trabalho.

- **Geral**

Desenvolver e avaliar um sistema distribuído para prover a adaptação de contexto em uma residência conforme as preferências dos usuários.

- **Específicos**

1. Pesquisar o estado da arte da computação ubíqua e da automação residencial.
2. Estudar as plataformas open *hardware* Raspberry PI e Arduino e as linguagens de programação Python e Wiring.
3. Especificar um sistema distribuído para adaptação de contexto residencial que considere, além das informações do usuário, características dos vários cômodos existentes em uma residência.
4. Projetar um sistema distribuído que atenda os requisitos definidos em (3).
5. Desenvolver o sistema distribuído projetado em (4).
6. Testar o sistema desenvolvido em (5) em bancada e avaliar os resultados obtidos.

1.2 JUSTIFICATIVA E MOTIVAÇÃO

Com o passar dos anos houve um aumento significativo na expectativa de vida das pessoas, e essa condição leva ao projeto de residências que ampliem o grau de acessibilidade de seus usuários. Para que isso aconteça, residências devem adotar sistemas automatizados que não somente ampliem o conforto e a segurança, mas sim que melhorem a qualidade de vida dos usuários em todos os aspectos, permitindo que pessoas vivam de maneira mais independente.

Somente automatizar uma residência não resolve o problema, é preciso transformar as moradias automatizadas, de agentes reativos para agentes proativos, em outras palavras, não programar uma casa

para as pessoas de modo que ela simplesmente responda as ações previamente programadas, mas sim dotar a residência de inteligência, a ponto de atender as necessidades, os estilos e os costumes dos usuários.

Prover inteligência a uma residência exige a colaboração de vários recursos, que coordenados por um sistema computacional coletam informações sobre o ambiente, como o grau de luminosidade, a temperatura, a umidade, ou até mesmo o nível de ventilação de um cômodo, e ações de seus habitantes, como por exemplo, a permanência ou não de uma pessoa em um determinado cômodo da residência.

Partindo do princípio de que a residência conhece quem reside em seu interior, a mesma poderia inferir soluções para eventuais situações emergenciais, como cortar o fornecimento de gás de um fogão ao perceber que um usuário se ausentou do imóvel e que tenha acidentalmente o esquecido ligado, ou informar à parentes e/ou corpo de bombeiros, que uma pessoa de idade mais avançada encontra-se em estado emergencial de saúde.

Fazer com que a residência perceba o que ocorre em seu interior para prevenir situações catastróficas é uma subárea da automação residencial denominada *Home Care* (GIROUX; PIGOT, 2005), recomendada para pessoas que necessitam de cuidados contínuos e possuem dificuldade para efetuar as atividades do cotidiano.

Uma aplicação sensível ao contexto tem implicações positivas em uma residência, pois é responsável por distribuir as devidas funções e configurações para seus respectivos dispositivos finais.

Um exemplo é o caso onde uma criança decide ir à cozinha para brincar com as bocas do fogão ou com produtos de limpeza. É possível a residência impedir o acesso da criança a esses recursos? E também informar a algum usuário responsável sobre o que a criança está fazendo?

Um segundo exemplo pode ser, ao chegar do trabalho cansado, um usuário que liga a televisão todos os dias para assistir o jornal local e com o anoitecer, configura a luminosidade a ponto de lhe proporcionar melhor conforto. Conhecendo essas preferências, a residência poderia automaticamente configurar a televisão às necessidades do usuário, quando este apenas se sentar na poltrona em frente ao televisor.

Mas, e se o usuário em questão resolve assistir o noticiário em seu dormitório? Em uma residência tradicional, ele deveria desligar todos os aparelhos da sala, rumar em direção ao quarto, ligar a televisão e ajustar a luminosidade novamente. Nesse caso surge um questionamento, será que a residência não poderia prover toda essa adaptação de contexto de forma automática para este usuário? Isso aumentaria

o conforto, pois a pessoa precisaria somente se levantar da poltrona, ir para o dormitório e deitar em sua cama, toda a configuração de áudio, vídeo e luminosidade seria de responsabilidade da residência, e não mais do usuário.

Permitir que uma residência reconheça seus usuários e adapte seus ambientes internos de acordo com as preferências destes, movendo com ele todo seu contexto computacional, cômodo por cômodo, por meio de um sistema eficiente e confiável, é o principal objetivo e contribuição desse trabalho.

1.3 METODOLOGIA

Este trabalho é uma pesquisa tecnológica que visa gerar um processo de automação residencial inteligente e possui a finalidade de melhorar a qualidade de vida das pessoas que possam vir a utilizar este sistema.

O sistema foi desenvolvido de maneira distribuída, com um servidor e vários clientes, no qual o servidor executa em um Raspberry PI, desenvolvido na linguagem de programação Python, e os clientes executam na plataforma Arduino, desenvolvidos em Wiring. A comunicação entre o servidor e os clientes se dá via Ethernet.

Os testes foram realizados em bancada, no LARM - Laboratório de Automação e Robótica Móvel, onde foram simulados vários cenários, onde cada dispositivo gerava um log do sistema. Os resultados foram analisados para avaliar o desempenho do sistema.

1.4 ORGANIZAÇÃO DO TRABALHO

Este documento, além desta introdução, está organizado em mais 5 (cinco) capítulos que abordam os seguintes conteúdos:

O **Capítulo 2** descreve algumas definições sobre computação ubíqua, móvel e pervasiva, além de aplicabilidade, conceitos e arquiteturas da computação inserida no dia-a-dia dos usuários.

O **Capítulo 3** discorre sobre a domótica e automação residencial por meio de conceitos, arquiteturas de implementação, produtos e tecnologias de automação e controle residencial.

O **Capítulo 4** descreve o sistema de adaptação de contexto residencial. Para um melhor entendimento, as arquiteturas de *hardware* e *software* serão descritas separadamente.

O **Capítulo 5** apresenta os resultados obtidos na avaliação do sistema desenvolvido. Para avaliar o sistema foram elaborados três experimentos, sendo: um experimento onde o usuário controlará o acionamento de um ventilador; um segundo experimento onde o usuário controlará o estado de um televisor, e por fim, o terceiro experimento onde o usuário controlará o sistema de luminosidade de dois cômodos.

O **Capítulo 6** apresenta as considerações finais e algumas propostas para trabalhos futuros.

2 COMPUTAÇÃO UBÍQUA

Neste capítulo são descritas algumas definições sobre computação ubíqua, móvel e pervasiva, além de aplicabilidade, conceitos e arquiteturas da computação inserida no dia-a-dia dos usuários.

2.1 DEFINIÇÕES E OBJETIVOS

O termo computação ubíqua foi utilizado pela primeira vez pelo cientista Mark Weiser, chefe de pesquisa da Xerox Palo Alto *Research Center* (PARC), no artigo intitulado *The Computer for the 21st Century* (WEISER, 1991), de sua autoria, escrito em 1988. Essa ideia surgiu através da observação de Weiser no aumento das funcionalidades e da predominância de dispositivos computacionais portáteis, oferecendo melhores serviços à usuários finais.

Outros termos pelos quais a computação ubíqua também é conhecida são: UbiComp (*Ubiquitous Computing*), *Everyware* (*Hardware em tudo*) e até *Things That Think* (Coisas que Pensam), porém esse trabalho utilizará apenas o termo computação ubíqua. Ubiquidade se refere ao ato de “ser onipresente”, ou “estar em toda parte” (IBRAHIM, 2009).

A computação ubíqua tem como proposta tornar simples, comum e natural a utilização de computadores, de modo que usuários não percebam as interações com tecnologias computacionais, em outras palavras, a computação ubíqua busca tornar implícita a integração entre dispositivos computacionais e usuários, por meio de um caminho natural, semelhante à outras tecnologias, como escrita, eletricidade, entre outras, que estão incorporadas ao cotidiano das pessoas (JUNIOR, 2011).

Nos primórdios da computação, ou Era do Mainframe, o cenário era caracterizado por um supercomputador atendendo a vários usuários. Na fase subsequente, iniciou-se a Era do Computador Pessoal (PC), onde essa relação entre usuário e computador se aproximava de um para um.

Observando a evolução no número de dispositivos computacionais e na massificação de dispositivos móveis, Weiser vislumbrou um cenário diferente ao da realidade da época, onde vários computadores serviriam vários usuários (CACERES; FRIDAY, 2012), a Era da Computação Ubíqua, realizável através da evolução da computação dis-

Tabela 1 – Evolução da computação em termos usuário x computador.

PC por Pessoa	
Era Mainframe	Um supercomputador para várias pessoas.
Era PC	Um computador para uma pessoa.
Era UbiComp	Vários computadores servem várias pessoas.

tribuída e da popularização da rede mundial de computadores. Um comparativo entre o número de computadores por pessoa é descrito na Tabela 1.

A prova dessa teoria é que atualmente o ser humano se encontra na Era da Computação Ubíqua, onde cada computador está inserido naturalmente no mundo real, um ambiente onde as pessoas podem usar um dispositivo computacional e acessar a internet facilmente (DONG; ZHANG; DONG, 2010). De acordo com Santana et al. (2009), a computação ubíqua é o oposto da realidade virtual, pois faz com que o mundo virtual penetre o mundo físico do usuário, enquanto a realidade virtual imerge o usuário em ambientes virtuais.

É comum usuários possuírem vários computadores como por exemplo um *desktop* no trabalho, um *desktop* pessoal, um *notebook*, um celular *smartphone* e/ou um *tablet*, e estes dispositivos compartilham funções similares para auxiliar em tarefas distintas. Essa era a principal ideia de Weiser, que na Era UbiComp, não só o número de aparelhos computacionais aumentasse, mas que também as suas tarefas, funções e formas fossem semelhantes, porém, de acordo com Araujo (2003), a ubiquidade computacional não significa um computador em cada mesa, mas sim, agregar à qualquer objeto, capacidade de processamento e comunicação, seja um interruptor, uma fechadura, ou um sapato.

Supondo que em uma sala de aula sejam substituídos os elementos clássicos, como por exemplo, quadros, folhas de papel, livros e cadernos, por aparelhos individuais como telas digitais, *tablets* e outros dispositivos com telas sensíveis ao toque (*touch screen*); os quadros poderiam conter uma interface para desenhar, escrever, ou demonstrar exemplos de aula, os livros digitais poderiam possuir ferramentas de busca, com significado de palavras, pesquisar por ideias relacionadas na web, e acessarem conteúdos de multimídia, e afins, ou quem sabe as canetas também poderiam ter acoplado um pequeno dispositivo de armazenamento que guardaria o que foi escrito, agruparia e enviaria à base de dados do ambiente virtual de aprendizagem da instituição de ensino que possui esses equipamentos. Isso dá uma grande ideia de o

que a computação em toda parte pode fazer.

Outro ponto da computação ubíqua defendido por Weiser era que os computadores estariam em toda parte, e por sua onipresença e naturalidade nos ambientes, acabariam imperceptíveis aos usuários (COULOURIS; DOLLIMORE; KINDBERG, 2007). De fato, atualmente há vários eletrodomésticos, automóveis, e sistemas de autoatendimento que possuem sistemas embarcados com microprocessadores capazes de gerenciar e automatizar estes aparelhos (FREITAS, 2013).

Todas essas possibilidades surgiram a partir de uma maior mobilidade computacional, além da miniaturização de dispositivos e do aumento da conectividade sem fio. Mediante tais condições, a computação ubíqua explora cada vez mais a integração de dispositivos computacionais com o mundo físico. Como estes dispositivos móveis diminuem em escala de tamanho com maior capacidade de processamento, fica mais fácil incorporá-los ao mundo físico e conectá-los a computadores pessoais, e/ou servidores convencionais usando redes sem fio.

A Figura 1 exemplifica a miniaturização dos aparelhos celulares, desde o DynaTAC, lançado pela Motorola em 1983 e mais à esquerda da imagem, até os tempos atuais, com o exemplar de um *smartphone* à direita.



Figura 1 – Evolução da modernização dos aparelhos celulares.
Extraído de (PIVOT, 2013)

Em um dispositivo ubíquo a quantidade de memória é bem limitada, por isso é necessária a execução de programas e o armazenamento de dados diretos de sua localização, na memória ROM (*Read-Only Memory*) do dispositivo, por exemplo. Essa característica de recursos limitados, força o uso de programas mais eficientes. De acordo com Araujo (2003) essas aplicações requerem um sistema operacional estável, pois não devem ser reiniciadas e nem atualizadas durante sua vida útil.

Juntamente com a computação móvel, a computação ubíqua é responsável por várias subáreas como a computação trajável (*Wearable Computing*) (BORONOWSKY et al., 2006), a computação de mão (*Handheld Computing*) (HAGEN; CONNORS; PELLON, 2003) e a computação sensível ao contexto (*Context-Aware Computing*) (MOSTEFAOUI; PASQUIER-ROCHA; BREZILLON, 2004).

Conforme ilustrado na Figura 2, a computação ubíqua está entre a computação móvel e a computação pervasiva.



Figura 2 – Relação entre computação ubíqua, móvel e pervasiva.
Adaptado de (ARAÚJO, 2003)

Diferentemente da computação tradicional, que é caracterizada pelo baixo grau de integração com o ambiente e por nenhuma mobilidade, onde *mainframes*, *workstations* e *desktops* executam tarefas por meio de interfaces tradicionais, compostas de dispositivos de entrada e saída, como teclado, mouse e monitor (JUNIOR, 2011), a computação ubíqua utiliza um modelo computacional que permite que usuários se movimentem, e que serviços e recursos localizem usuários, além de uns aos outros (CIRILO, 2010).

Com a integração de sistemas móveis e pervasivos, na qual tem se tornado cada vez menos perceptível pela sua presença no cotidiano, sistemas ubíquos estão cada vez mais eficientes, e beneficiam o usuário final em conectividade e desempenho. A Tabela 2 apresenta um comparativo entre a mobilidade e o grau de embarcamento da computação ubíqua, móvel e pervasiva.

2.2 COMPUTAÇÃO MÓVEL

A computação móvel se baseia em sistemas computacionais distribuídos em diferentes dispositivos, no qual há uma intercomunicação por redes sem fio, o que permite a mobilidade desses aparelhos (ADELSTEIN et al. apud CIRILO, 2010). Surgiu com a necessidade de usuários

Tabela 2 – Dimensões da computação ubíqua.

	Mobilidade	Grau de “Embarcamento”
Computação Tradicional	Baixa	Baixo
Computação Pervasiva	Baixa	Alto
Computação Móvel	Alta	Baixo
Computação Ubíqua	Alta	Alto

Adaptado de (ARAUJO, 2003)

carregarem seus computadores pessoais, mantendo uma conectividade com outros dispositivos. A Figura 3 ilustra o conceito da computação móvel, que é responsável por agregar mobilidade à dispositivos computacionais.



Figura 3 – Princípios da computação móvel.

Na década de 80, os computadores pessoais ficaram mais leves, e por consequência, mais fáceis de carregar; havia ainda a possibilidade de se interligarem através de linhas telefônicas, por meio de um *modem*, que alterava o sinal elétrico e tornava possível a conexão com a Internet. Atualmente, pode-se observar que com a vinda de novas gerações de conexões de dispositivos móveis, surgiram novas tecnologias em comunicações como ZigBee, Wi-Fi (*Wireless Fidelity*) Direct, Bluetooth Low Energy, Z-Wave e 4G (*Fourth Generation*).

Com os avanços da tecnologia móvel foi possível o surgimento da computação de mão (*Handheld Computing*), aparelhos que cabem na palma da mão, como os PDAs (*Personal Digital Assistants*), *smartphones*, *smartwatches*. Os *smartwatches* são relógios de pulso inteligentes, com sistema operacional embarcado, *touch screen*, além de várias fun-

cionalidades, como efetivação de chamadas telefônicas, recebimento e envio de e-mails, consulta à meteorologia, entre outras (SONY, 2013). A Figura 4 ilustra alguns *smartwatches* fabricados pela multinacional japonesa Sony.



Figura 4 – Sony *SmartWatch*®.
Extraído de (SONY, 2013)

2.3 COMPUTAÇÃO PERSASIVA

A computação pervasiva é responsável por embutir dispositivos computacionais ao ambiente, essa integração pode ser perceptível ou não do ponto de vista do usuário (KAHL; FLORIANO, 2011). Esse tipo de computação habilita que dispositivos computacionais colem informações sobre o seu entorno, e se ajuste, para se integrar ao ambiente (CIRILO, 2010). A Figura 5 ilustra a computação pervasiva, que é o ponto de intersecção entre dispositivos computacionais e o ambiente.



Figura 5 – Áreas que compõem a computação pervasiva.

A computação pervasiva se assemelha muito à computação ubíqua, porém, no contexto computacional, a pervasividade é somente uma parte da ubiquidade, pois possui um baixo grau de mobilidade, o que

torna sistemas pervasivos “engessados e estáticos”.

A pervasividade computacional implica que dispositivos não sejam apenas aparelhos posicionados de maneira estratégica, mas sim que esteja distribuídos no ambiente, e indica a integração de computadores com outros elementos, tais como sensores, isolados ou em rede, e interfaces presentes em objetos de mobília e vestuário (JUNIOR, 2011).

Tais características permitem que sistemas pervasivos sejam capazes de identificar e coletar dados e informações por meio de interfaces de comunicação, essas informações podem ser variações do ambiente ou ações dos usuários, possibilitando a geração de modelos computacionais automáticos, configurados conforme a necessidade do usuário, da aplicação e dos dispositivos da rede. Com essa interação é possível que cada membro do sistema seja capaz de identificar a presença de outros elementos no ambiente, tanto usuários, como outros dispositivos eletrônicos, sendo assim possível a interação automática entre os elementos, onde pode ser construído um contexto inteligente, que potencializa o uso de sistemas pervasivos.

2.4 INTERFACES UBÍQUAS DE COMUNICAÇÃO

São necessárias inovações tecnológicas para facilitar o processo de comunicação entre usuários e computadores, que é fundamental para proporcionar avanços significativos nas tecnologias ubíquas e suas diversas aplicações na realidade humana, ou seja, a computação ubíqua estimula que tecnologias se desenvolvam para dispositivos que vão além do computador de mesa, criando uma integração entre usuário e computador diferente da que existe atualmente, por meio de dispositivos de entrada e/ou saída de dados.

A computação ubíqua carece de tecnologias que se adaptem aos requisitos humanos de usabilidade e ergonomia, por isso necessita de interfaces que aceitem diversas opções de manipulação e que operem através de equipamentos acessíveis (OVIATT apud JUNIOR, 2011), em outras palavras, o desafio é aumentar a integração entre usuário e computador, por meio de interfaces capazes de compreender gestos, expressões regionais, variações de sotaques, e enviar os comandos necessários para um computador, que aprende com as experiências registradas anteriormente, e as relacionem aos dados recebidos dos sensores, que fazem parte da rede, representando mais que uma simples tradução de comandos para linguagem de máquina. Com um melhor desempenho de sistemas ubíquos, o usuário poderá se concentrar no processo

de comunicação, troca e compartilhamento de dados.

As ferramentas que tornam isso possível são aplicações que fazem uso de computação cooperativa (*Computer Supported Cooperative Work*), que são aplicativos que auxiliam no processo de construção e compartilhamento de informação e conhecimento de maneira cooperativa (BEHAR; COSTA, 1996). Todo esse conteúdo é armazenado em um repositório, e disponibilizado para todos os usuários da aplicação.

Um expoente da computação cooperativa é a Wikipedia¹, a maior enciclopédia do mundo, oriunda da colaboração de seus usuários, e ambientes inteligentes, que por meio de dispositivos e tecnologias trabalhando de maneira cooperativa, aperfeiçoam alternativas que tornam desnecessário que alguém esteja na frente de um computador ou de dispositivos computacionais para a manipulação do sistema, e detectam alterações de contexto no ambiente, como a entrada de um novo dispositivo, ou a saída de um usuário (DOMINGUES, 2008).

Essa característica dos ambientes inteligentes permite que eles funcionem à distância, de maneira autônoma, sem a intervenção do usuário. Em ambientes inteligentes observa-se os seguintes elementos:

- **Hands-Free:** são dispositivos ou aplicações que reconhecem instruções ou dados de seus usuários, sem que os mesmos necessitem utilizar as mãos para efetivar a interação (MANFIO, 2012). Atualmente é possível ver este tipo de tecnologia em programas de reconhecimento de voz, e outros dispositivos que permitem, por exemplo, que a interação entre o dispositivo e o usuário continuem, enquanto a pessoa se move em uma sala de aula, como *liveboards*, *tablets* e Kinect. A Figura 6 apresenta um sensor Kinect® , um dispositivo desenvolvido pela Microsoft, capaz de interagir com jogos eletrônicos sem a necessidade de *joystick*. O Kinect® utiliza um projetor, uma câmera e um sensor de infravermelho.
- **Noção de Contexto:** trabalha através de sensores instalados no ambiente interno, e tem como função detectar e informar ao ambiente inteligente o que acontece ao redor; as informações podem ser ações de usuários, configurações dos dispositivos dispostos ao meio, e dados do ambiente como temperatura, luminosidade, ventilação, entre outros. O ambiente ao receber esses dados, deve atuar de acordo com a necessidade do usuário, tomando as decisões corretas para cada cenário específico (DOMINGUES, 2008).

¹Disponível em: <http://www.wikipedia.org/>

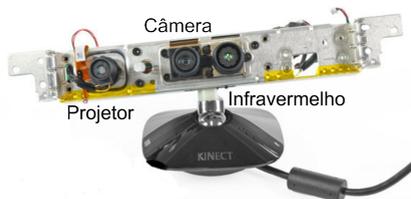


Figura 6 – Microsoft Kinect®.
Extraído de (SMISEK; JANCOSEK; PAJDLA, 2011)

2.5 RESIDÊNCIAS INTELIGENTES

Dentro dos princípios da computação ubíqua, os computadores são embarcados em todo lugar no ambiente, projetados para perceber a presença dos usuários e agir de acordo com determinadas situações (WEISER, 1991).

Para interagir tanto com o usuário quanto com ambientes digitais, foram projetados dispositivos móveis como celulares *smartphones*, PDAs, *tablets*, que proporcionam ao usuário uma maior eficiência em várias tarefas, interagindo com ambientes inteligentes, e seguindo a presença discreta destes dispositivos, como se fossem invisíveis, já que a computação ubíqua está em segundo plano.



Figura 7 – Google Glass®.
Adaptado de (STARNER, 2013)

Algumas aplicações que operam em sistemas ubíquos são responsáveis por fazer a associação e interação entre os componentes de *software* quando estes interagem entre si para acompanhar entidades que se movem, falham ou surgem espontaneamente em seus ambientes, e explorar a sua integração com o mundo físico, usando meios perceptivos e de reconhecimento de contexto.

Por meio do avanço tecnológico, atualmente a computação ubíqua já é realidade em alguns casos, como:

- *Wearable Computing*: são dispositivos computacionais trajáveis, como *smartwatches*, *smartshirts*, ou por exemplo o Google Glass. A Figura 7 ilustra o Google Glass®, um moderno dispositivo computacional, que pode registrar imagens e sons vivenciados pelo usuário, além de permitir a conexão com a Internet, e enviar conteúdo multimídia em tempo real.
- Automóveis: exemplos de ubiquidade computacional, os carros atualmente saem de fábrica dotados de vários sensores e sistemas embarcados que melhoram a experiência de conduzir um veículo, além de informar falhas e auxiliar o condutor em manobras específicas.

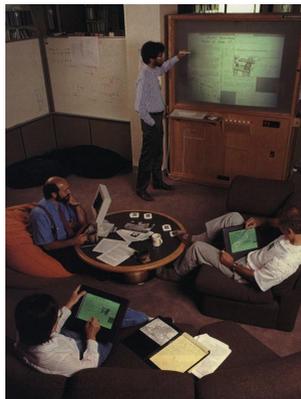


Figura 8 – Sala de reuniões da Xerox PARC.
Extraído de (WEISER, 1991)

- Ambiente de Aprendizagem Modernizado: salas de aula na qual há uma conectividade entre uma lousa digital, projetor e dispositivos móveis, por exemplo.

- Sala de Reuniões Sensível ao Contexto: salas desenvolvidas com elevado número de dispositivos tecnológicos, que auxiliam na tomada de decisões em ambientes organizacionais cooperativos e que potencializam a criatividade dos usuários. A Figura 8 demonstra uma reunião da Xerox PARC, com Mark Weiser (mais à esquerda) e mais três cientistas, e prova que salas de reuniões inteligentes são alvos de estudos sobre computação ubíqua desde a criação do termo, em meados dos anos 80.



Figura 9 – Aware Home.
Extraído de (INSTITUTE, 1999)

Além dos casos citados anteriormente, as residências merecem uma atenção especial, pois são alvos constantes de pesquisa para o desenvolvimento de aplicações ubíquas. Nesse ambiente é possível conhecer e mapear as atividades, os costumes e as necessidades dos usuários de uma casa (ARAUJO, 2003). Para fornecer serviços que proporcionem um aumento na qualidade de vida dos usuários, existem projetos de grandes organizações, dentre os quais se destacam o Aware Home, EasyLiving e a Adaptive House.

O Aware Home Research Initiative (AHRI) (INITIATIVE, 2000), é um projeto do Grupo de Pesquisas Tecnológicas do Instituto americano de Tecnologia da Georgia. Iniciado em 2000, o projeto é pioneiro no gênero “laboratório residencial”. O AHRI foi desenvolvido para proporcionar um ambiente familiar autêntico, de maneira flexível, que permita o avanço de pesquisas em diversas áreas, como por exemplo sensoriamento, infraestrutura de controle, fornecimento de dados e saúde e bem-estar.

Essa residência inteligente encontra-se implantada em um terreno de cinco mil metros quadrados e possui dois andares idênticos,

com cômodos normais de uma residência e outros cômodos especiais, utilizados para experimentações com o sistema (KIDD et al., 1999). Uma imagem da Aware Home é ilustrada na Figura 9.

O EasyLiving é um projeto do Grupo de Pesquisas da Microsoft voltado para a criação de arquiteturas e tecnologias funcionais para ambientes inteligentes, que auxiliem a interatividade entre as pessoas e dispositivos computacionais (SHAFER et al., 1998).

O projeto possui soluções para a maioria dos cômodos de uma residência, porém o que prende a atenção é a sala, dotada de dispositivos como computadores, *home-theater*, poltronas, e serviços que fornecem um controle automático de luminosidade, acústica baseada na localização do usuário, e transferência de conteúdos entre as telas dispostas no ambiente (ARAUJO, 2003).

E por fim, desenvolvido pela Universidade do Colorado, a Adaptive House, ilustrada na Figura 10, surge com o conceito de não ser uma casa programada, mas sim uma casa que se programa. Nesse contexto, a residência é quem busca se adaptar aos estilos e costumes dos usuários, monitorando o ambiente e se antecipando as necessidades das pessoas, inclusive sob o aspecto de eficiência energética (MOZER, 2005).



Figura 10 – Adaptive House.
Adaptado de (MOZER, 2005)

A residência inteligente opera por meio de um *middleware*, responsável por repassar para o *software* de gerenciamento os dados coletados por sensores dispostos na residência, e atua no abastecimento de água e nos sistemas de ventilação, aquecimento e iluminação.

2.6 MIDDLEWARES

*Middleware*s são frameworks de integração, responsáveis pela integração de aplicações distribuídas em uma única arquitetura (HUGUES; PAUTET; KORDON, 2003), e podem ser vistos como uma camada de *software* intermediária, que provê à camada superior uma visão abstrata sobre os serviços que as camadas inferiores disponibilizam.

Middlewares facilitam o desenvolvimento de aplicações distribuídas, e a integração de sistemas que não foram previamente projetados para trabalharem de maneira integrada (IBRAHIM, 2009). Em aplicações ubíquas, *middlewares* são responsáveis pelo transporte dos dados oriundos de sensores para a camada superior, onde se encontra a aplicação ubíqua. Dentre os *middlewares* mais conhecidos para aplicações ubíquas, serão citados Gaia e One.World.

- **Gaia:**

O Gaia foi desenvolvido pela Universidade de Illinois em Urbana-Champaign, e é um *middleware* sistema operacional, ou seja, abstrai o acesso à recursos físicos de um computador, e define uma estrutura interna de *software* distribuída para o desenvolvimento e compartilhamento de aplicações ubíquas (ROMAN et al., 2002).

Oferece serviços de localização, eventos de contexto e repositórios de informações através de uma camada, denominada espaço ativo, que provê a usuários móveis uma maior interação e configuração de seus espaços físicos e digitais (CORREA, 2006). A Figura 11 ilustra a arquitetura do projeto Gaia.

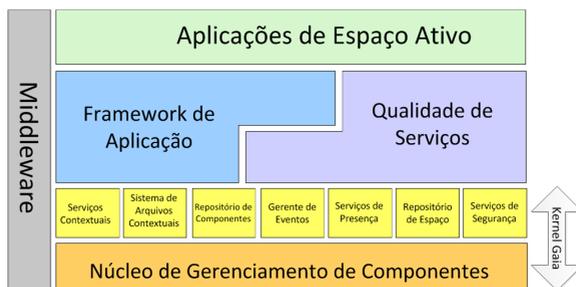


Figura 11 – Arquitetura do projeto Gaia.
Adaptado de (GOMES, 2007)

O kernel² do Gaia é responsável por definir todos os serviços essenciais para a inicialização básica do sistema, esses serviços básicos estão localizados nas duas camadas mais inferiores do *middleware*.

A linguagem LuaOrb, baseada em Lua, e semelhante à linguagem BASH, disponibiliza todo o aparato para testes, extensão de componentes, prototipação rápida de novas aplicações e configuração dos serviços de inicialização do sistema (GOMES, 2007).

- **One.World:**

O projeto One.World é um *middleware* para aplicações ubíquas, e foi originalmente criado pela Universidade de Washington, porém, atualmente está em desenvolvimento na Universidade de Nova York (COSTA, 2006). Esse *middleware* é estruturado em dois níveis: o nível de usuário, composto pela camada de espaço do usuário, que fornece serviços dinâmicos, que permitem a mobilidade de aplicações, com tolerância a falhas e *checkpoint*, e o nível de kernel, que é dividido em duas camadas, serviços fundamentais e serviços de sistema (CORREA, 2006). A Figura 12 ilustra a arquitetura do projeto One.World.

Os serviços fundamentais do One.World são compostos por quatro blocos: a máquina virtual, que provê um ambiente de execução similar ao restante da estrutura do *middleware*; tuplas, é o modelo de representação universal dos dados do *middleware*, que simplifica o compartilhamento de informações (GRIMM, 2004); eventos assíncronos, responsáveis por notificar mudanças no contexto de execução; e por fim, ambientes, que isolam as aplicações e armazenam as que se encontrem em execução (COSTA, 2006).

Subsequente à camada de serviços fundamentais, a camada de serviços do sistema auxilia no desenvolvimento de aplicações, e conta com: serviço de gerenciamento de dados, que permite ao *middleware* consultar, armazenar e trocar informações na forma de tuplas, e é formado pelo trabalho em conjunto da máquina de busca e o serviço de armazenamento de tuplas; modelo de comunicação, baseado no serviço de eventos, responsável por direcionar eventos para serviços e locais remotos; serviço de descoberta, que disponibiliza uma interface dinâmica para tratamento de eventos; e serviço de checkpoint, responsável por capturar o estado

²Kernel - Núcleo de um sistema operacional, responsável pelo interfaceamento com o *hardware* para prover recursos do sistema para aplicações de usuário.

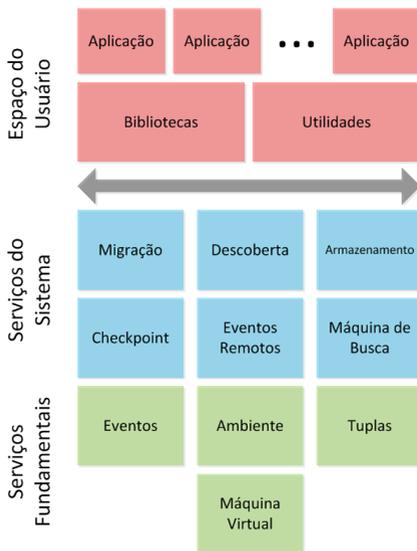


Figura 12 – Arquitetura do One.World.
Adaptado de (COSTA, 2006)

de execução do ambiente, e o armazenar em forma de tupla, para posterior carregamento desse estado do sistema (CORREA, 2006).

2.7 DESAFIOS DA COMPUTAÇÃO UBÍQUA

Mesmo com o aumento do número de microcontroladores em sensores e atuadores, a computação ubíqua ainda enfrenta barreiras que precisam ser superadas para que haja uma aceitação significativa entre as pessoas. Para superar esse desafios é necessário desenvolver soluções acessíveis e eficazes.

A computação ubíqua é computacionalmente distribuída, utiliza sensores, e a rede de computadores, por isso, faz-se necessária o desenvolvimento de novas arquiteturas de implementação que permitam a coleta, o processamento e a disseminação de dados contextuais de usuários para tratamento de contexto, sem abrir mão da segurança da informação (YEE, 2005), pois, quando qualquer dispositivo acessar uma rede, ele também pode, dependendo das configurações, extrair in-

formações importantes dos usuários e do sistema.

Outro ponto importante, é a criação de redes sem fio padronizadas, “amigáveis” e seguras, que tornem a configuração de dispositivos na rede menos árdua, que por consequência atingiria um público-alvo maior. Por exemplo, em redes bluetooth, existem serviços de localização de dispositivos, que permitem a descoberta de nomes e endereços de dispositivos presentes em um ambiente, e também a verificação de serviços, onde o usuário acaba ignorando qual dispositivo fornece uma aplicação, em prol das vantagens do serviço oferecido.

Com a modernização dos *hardwares*, *softwares* e das redes de comunicação, a computação ubíqua não está distante da realidade da maioria das pessoas, porém para seu sucesso, é necessário oferecer suporte social, satisfação e auxílio para seus usuários, participando da vida dos demais, próximos ou à distância, mas sem substituir os vínculos que existe entre as pessoas.

3 DOMÓTICA

Este capítulo discorre sobre a domótica e automação residencial por meio de conceitos, arquiteturas de implementação, produtos e tecnologias de automação e controle residencial.

3.1 DEFINIÇÃO E OBJETIVOS

A evolução dos processos automatizados na indústria, trouxeram padrões, uniformidade e flexibilização nos meios de produção. Com o intuito de melhorar a qualidade de vida das pessoas, a automação vem se estendendo do ambiente industrial para o ambiente doméstico (FERREIRA, 2010).

Domótica é um termo que nasceu da junção da palavra *domus*, que significa residência em latim, e robótica (CHAMUSCA, 2006). Também pode ser definida como a utilização da eletricidade, eletrônica e das tecnologias da informação de maneira simultânea, para realizar a gestão, seja ela local ou remota, de uma residência e disponibilizar aplicações que aumentem os meios de comunicação internos, e ampliem a eficiência energética, a segurança e o conforto dos residentes (MARIOTONI; JR., 2002).

Apesar da similaridade entre automação residencial e domótica, existem diferenças, como por exemplo, a automação residencial é composta por um ou mais dispositivos atuando de maneira singular sem qualquer comunicação entre os mesmos, ao passo que a domótica descreve a integração entre todos os dispositivos fazendo com que eles atuem em conjunto para uma determinada função especificada no projeto (FERREIRA, 2010).

A domótica acopla tecnologia em residências objetivando melhorar a qualidade de vida dos residentes, priorizando a segurança, eficiência energética, o conforto e o uso dos meios de comunicação, viabilizando o uso racional dos recursos para seus habitantes, como ilustra a Figura 13.

Muratori e Bo (2011), baseados em pesquisas realizadas nos Estados Unidos, afirmam que:

- 84 % dos construtores norte-americanos admitem que é um diferencial agregar tecnologia nas residências.
- sistemas automatizados que empregam sustentabilidade, eficiência

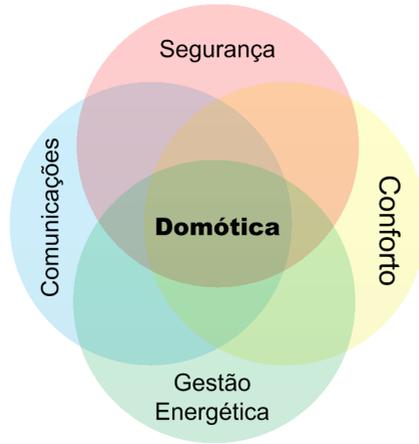


Figura 13 – Áreas que compõem a domótica.
Adaptado de (FERREIRA, 2010)

energética e preservação de recursos naturais são cada vez mais requisitados.

- as tendências atuais que devem atingir um grande crescimento nos próximos anos são os *media centers*, o monitoramento à distância e o controle de iluminação.

A domótica pode ser um fator decisivo para atingir consumidores com necessidades específicas. Ainda de acordo com Muratori e Bo (2011), a utilização de diferenciais nos produtos imobiliários será fator imperativo para atrair uma base maior de consumidores.

3.2 NÍVEIS DE AUTOMAÇÃO RESIDENCIAL

Sistemas de automação residencial necessitam de dispositivos de *hardware*, como sensores e atuadores, que são fundamentais para a coleta de informações e acionamentos eletroeletrônicos.

Os sensores necessitam estar estrategicamente acoplados ao ambiente, para obter maior precisão na coleta de dados (TEJANI; AL-KUWARI; POTDAR, 2011), e são divididos em duas categorias: sensores de presença, que são projetados para atuar em ambientes internos, e são responsáveis por detectar a presença de um usuário em seu raio de

alcance, e repassar essa informação para seu sistema de gerenciamento; e sensores de ambiente, que coletam dados sobre o ambiente, como por exemplo o nível de luminosidade, a temperatura, o grau de abertura das janelas, a velocidade do vento que entra pelos dutos de ventilação, e até mesmo se uma eventual chuva está invadindo o cômodo de uma residência. A Figura 14 ilustra um sensor de umidade e temperatura.



Figura 14 – Sensor de umidade e temperatura.
Extraído de (ARDUINO, 2013)

Atuadores são responsáveis pela realização dos processos aos quais é aplicada a automação. Atuadores podem ser de acionamento magnético, pneumático, elétrico, ou misto.

Além do uso destes dispositivos, os sistemas de automação residencial são divididos em três classes, de acordo com o nível de integração, sendo a primeira classe composta por sistemas autônomos, a segunda e a terceira por sistemas integrados e sistemas complexos, respectivamente (PINHEIRO, 2004).

3.2.1 Sistemas Autônomos

Sistemas Autônomos são os que empregam menor nível de complexidade de implantação e também possuem um baixo custo, como desvantagem, não oferecem grandes funcionalidades, apenas alguns comandos simples, como por exemplo, ligar e desligar um dispositivo (SILVA; CARVALHO, 2011). Este tipo de implementação não permite a integração entre os dispositivos, ou outros sistemas de automação implantados na residência. A configuração dos dispositivos é definida previamente, por meio de um controlador.

Como este sistema opera de forma independente, não há a possibilidade de inserir controles mais complexos, pois cada dispositivo possui seu próprio controle.

3.2.2 Sistemas Integrados - Central de Automação

Sistemas integrados de automação acoplam vários sistemas a um único controlador central, esse dispositivo é denominado Central de Automação (PINHEIRO, 2004), e é responsável pela interação entre o usuário e o sistema, e também por detalhar todas as informações que a residência oferece ao sistema automatizado. Essas funcionalidades podem ser disponibilizadas por um aparelho IHC (*Intelligent Home Control*), ilustrado na Figura 15, inserido em um ponto estratégico da residência.



Figura 15 – IHC.
Adaptado de (TELMOTICA, 2013)

Este tipo de sistema difere dos sistemas autônomos, pois permite a comunicação entre os dispositivos e assim o controle e o gerenciamento de toda a residência.

Apesar das funcionalidades, a ampla gama de integração e os benefícios aos usuários que esses sistemas oferecem, eles estão limitados a operar na forma pretendida por seus respectivos fabricantes, como vantagem, há um melhor aproveitamento dos recursos utilizados (NETO; MENON, apud DA SILVA; DE CARVALHO, 2011).

3.2.3 Sistemas Complexos - Residência Inteligente

Os sistemas complexos de automação residencial permitem a integração de todos os subsistemas de uma residência, como por exemplo, a climatização, iluminação, os sinais de áudio e vídeo, entre outros. Para que esse tipo de sistema seja implantado com o menor custo possível, é necessário anteriormente à fase de construção da residência, planejar todo o cabeamento necessário, para que haja uma melhor in-

tegração de forma simples.

Sistemas complexos, assim como os sistemas integrados, permitem controlar e gerenciar todos os subsistemas da residência, porém com um diferencial, que é disponibilizar todo esse conteúdo de maneira remota, através de um computador, um telefone celular, um *smartphone*, ou *tablet* que tenha acesso à *Internet*. A Figura 16 ilustra um *smartphone* da Samsung, o Galaxy S3®.



Figura 16 – Samsung Galaxy S3®.
Extraído de (SAMSUNG, 2013)

3.3 ARQUITETURAS PARA AUTOMAÇÃO RESIDENCIAL

A domótica se divide basicamente em dois tipos de arquiteturas. A ABA (Arquitetura Baseada em Automação), também conhecida como Domótica Estática, e a ABC (Automação Baseada em Comportamento), conhecida por Domótica Inteligente. Ambas são descritas na Seção 3.3.1 e Seção 3.3.2.

3.3.1 ABA - Arquitetura baseada em Automação

A arquitetura ABA implementa um sistema no qual o usuário é quem deve se adaptar à automação implantada. Essa arquitetura favorece a implantação de operações voltadas a deixar o sistema com interfaces simples e objetivas (MEIRA; DELFINO; ORLANDINI, 2011).

A ABA automatiza a residência a partir de dispositivos que são ajustados e configurados pelos próprios usuários, assim os residentes não precisam se adaptar à tecnologia, já que o sistema foi adequado para suprir as necessidades dos usuários. Esses dispositivos podem ser

controles remotos, sensores de movimentos, dispositivos biométricos, etc.

3.3.2 ABC - Automação baseada em Comportamento

O sistema ABC incorpora mecanismos de tomada de decisão baseadas em técnicas de IA (Inteligência Artificial) (BRUNETTE; FLEMMER; FLEMMER, 2009), é modelado e tem suas regras inferidas a partir do comportamento dos habitantes da residência.

O sistema ABC utiliza o algoritmo de aprendizado ID3¹ (YUXUN; NIUNI, 2010), que prevê integralmente o processo de adaptação ao sistema automatizado.

A arquitetura do sistema ABC faz uso de atuadores, banco de dados e dois tipos de sensores, aqueles que podem agir com interferência humana e os que não podem. O controlador monitora constantemente os sensores de cada ambiente, verificando se existe alguma regra de automação armazenada em seu banco de dados ou nas regras de segurança que se aplique ao estado dos sensores (SGARBI, 2007). O esquema da arquitetura ABC pode ser observado na Figura 17.

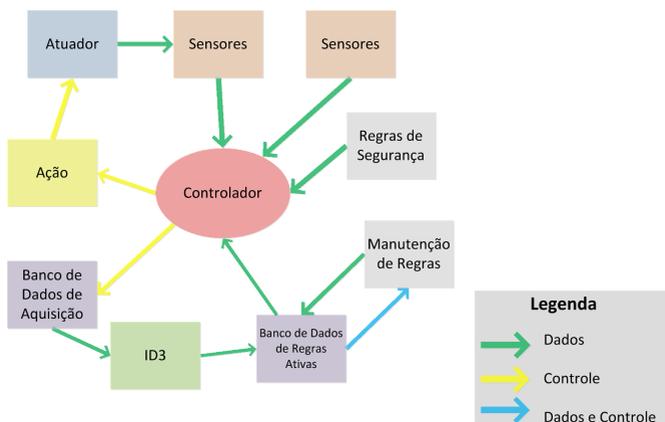


Figura 17 – Arquitetura ABC.
Adaptado de (SGARBI, 2007)

O banco de dados do sistema é alimentado com dados proveni-

¹ID3 - *Iterative Dichotomiser 3* é um algoritmo utilizado para gerar uma árvore de decisões a partir de um conjunto de dados.

entes dos sensores e dos eventos que ocorrem nos atuadores. Quando o banco de dados atinge um número de eventos, ele é inserido no algoritmo ID3, que é responsável por generalizar os dados e criar as regras de inferência, essas regras são guardadas em um banco de dados de regras ativas. O sistema faz uso de mais uma base de dados onde estão as regras de segurança (SGARBI, 2007).

A cada ação do usuário na residência o sistema analisa as regras ativas, e caso alguma regra se aplique, o sistema executa determinadas ações nos atuadores necessários. As regras do sistema podem ser removidas de forma manual ou automaticamente por tempo de inatividade (TAKIUCHI; MELO; TONIDANDEL, 2004).

3.4 TECNOLOGIAS DE AUTOMAÇÃO E CONTROLE

Tecnologias de Automação e Controle, ou TACs, são as tecnologias utilizadas por sistemas de automação para o controle de processos, dispositivos ou máquinas, e são responsáveis por executarem de forma automática ações e funções. Existe uma forte integração entre as TACs e as TICs (Tecnologias da Informação e Comunicação), o que permite que o controle da automação seja feita através de uma rede de computadores, ou por dispositivos informatizados. As TACs podem ser divididas em TACs com meio de comunicação definido e TACs sem fio.

3.4.1 TACs com meio de comunicação definido

O que caracteriza uma TAC com meio de comunicação definido é a adoção de cabos para a comunicação entre os dispositivos e para o envio e recebimento de dados. O cabeamento utilizado pode ser desde par trançado até a PLC (*Powerline Communication*, ou comunicação baseada na rede elétrica), entre outros. O uso de meios definidos de comunicação não significa que uma TAC não possa adotar também uma tecnologia sem fio para a transmissão de dados. Os itens a seguir contém descrições sobre TACs com meio de comunicação definido, como X-10, EIB - KNX e LonWorks.

- **X-10**

O X-10 permite controlar de forma remota aparelhos elétricos através da PLC, que é o principal meio de comunicação para os sistemas que fazem uso dessa TAC. Essa tecnologia foi criada em

meados dos anos 70, a partir de um empreendimento conjunto entre a empresa escocesa de tecnologia Pico Eletrônica e a britânica Birmingham Sound Reproducers (BSR).

Como faz uso da PLC, torna desnecessária a conexão de novos cabos para adicionar dispositivos, podendo ser aplicada a qualquer momento, durante a construção da residência ou posteriormente. Essa TAC é caracterizada por sua arquitetura descentralizada e necessita apenas de atuadores, receptores e da rede elétrica, o que a torna bastante flexível e imune a falhas, pois a falha de um componente não compromete o funcionamento dos demais. Qualquer fabricante pode utilizar a tecnologia X-10 em seus produtos, pois em 1997, o mesmo teve sua patente expirada (YUEJUN; MINGGUANG, 2005).

Um sistema de automação X-10 é constituído por dispositivos que são comandados pelo usuário. Por exemplo, o usuário pode enviar um comando para desligar o abajur através de um controlador (receptor), que está conectado a rede elétrica. Do outro lado, o abajur que também está conectado a rede elétrica por meio de um módulo atuador recebe os comandos do controlador. Esse esquema é ilustrado na Figura 18.

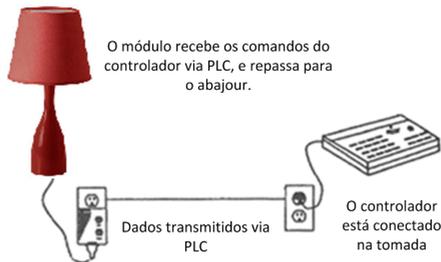


Figura 18 – Exemplo de uma aplicação X-10.
Adaptado de (ARAUJO, 2011)

Para resolver o problema de endereçamento dos dispositivos conectados a rede elétrica, já que todos podem receber os sinais enviados por um controlador, o X-10 implementa um sistema simples de endereçamento que é configurado manualmente pelo usuário.

Os controladores ligados a PLC podem ser endereçados com 16

códigos de controle (com o intervalo de letras de A à P), e enviam sinais para a rede elétrica, que são capturados por atuadores, módulos conectados aos aparelhos dos quais se pretende controlar. Os atuadores podem ser endereçados com 16 códigos de dispositivos (de 1 à 16). Esse sistema de endereçamento permite controlar 256 aparelhos (YUEJUN; MINGGUANG, 2005). A Figura 19 ilustra um módulo controlador fabricado pela POWERHOUSE.



Figura 19 – Exemplo de um módulo controlador X-10.
Extraído de (SIATRON, 2013)

Caso haja dispositivos com o mesmo endereço ambos responderão aos comandos enviados pelos controladores. Um dos pontos negativos do X-10 é a sua baixa velocidade de transmissão, limitada a 50 *bps* (*bits* por segundo). Como consequência dessa baixa velocidade de envio, o sistema é capaz de realizar apenas operações simples como ligar ou desligar aparelhos (ARAUJO, 2011).

Por ser uma tecnologia com um longo tempo no mercado, o X-10 possui baixo custo, facilidade de uso e grande variedade de equipamentos. O X-10 é mais orientado para sistemas caseiros, de pequena escala (MCCRICKARD; WRIGHTON; BUSSERT, 2002).

- **EIB / KNX**

O EIB / KNX (*European Installation Bus - Konnex*) foi criado em 1999, a partir da junção de três TACs: BatiBUS, EIB e EHS (*European Home System*). O principal objetivo dessa TAC é criar um padrão único para o mercado europeu de automação residencial e predial, que seja capaz de competir em número de produtos, qualidade e preços, com outras tecnologias disponíveis oriundas de outros continentes.

Essa TAC se baseia fortemente na antiga tecnologia EIB, porém com novos mecanismos de configuração e a adição de meios físicos de comunicação (RUTA et al., 2011). O EIB é uma tecnologia europeia não proprietária para controle e gerenciamento de ambientes, desenvolvida nos anos oitenta, pela *European Installation Bus Association* (EIBA), atualmente conhecida como *Konnex Association*. A *KNX Association* é responsável pela regulamentação e promoção do EIB / KNX, através da certificação de centros de formação.

EIB / KNX é um sistema descentralizado, onde cada um dos dispositivos tem seu próprio controle multiprocessado por um microcontrolador 68HC05 fabricado pela Motorola, esse processador provê toda a interface necessária para o funcionamento do sistema (SANZ; GARCIA-NICOLAS; ESTOPINAN, 2005). Essa descentralização permite que cada dispositivo detecte as informações necessárias para agir no ambiente, ou ainda, decidir quando enviar comandos e informações a outros dispositivos ou sistemas que estejam conectados.

O EIB / KNX pode ser aplicado em pequenas, médias ou grandes edificações, e suporta redes de estrutura hierárquica, como ilustra a Figura 20, consistindo de nodos, sub-redes, áreas e *backbone*.

Redes EIB / KNX podem suportar aproximadamente 62.000 dispositivos (AMORY; JÚNIOR, 2001). Esta tecnologia permite que os dispositivos interajam por meio de um barramento que pode ser um par trançado, a PLC, rádio frequência, infravermelho, entre outros. A velocidade de transmissão dos dados varia de acordo com o meio de comunicação utilizado.

• LonWorks

Em 1992, a *Echelon Corporation*, companhia norte-americana, desenvolveu um projeto audacioso denominado LonWorks, que é utilizado para implementar redes de controle automatizadas e distribuídas. Devido ao seu custo elevado em relação a outras tecnologias, LonWorks é indicado para edificações que necessitem de um sistema de automatização robusto e confiável, como por exemplo, hotéis, indústrias e edifícios administrativos (SHAHNASER; WANG, 1998).

O responsável por fazer o endereçamento e o transporte da informação ponto-a-ponto das redes LonWorks, é o protocolo LonTalk (Padrão ANSI - *American National Standards Institute* -

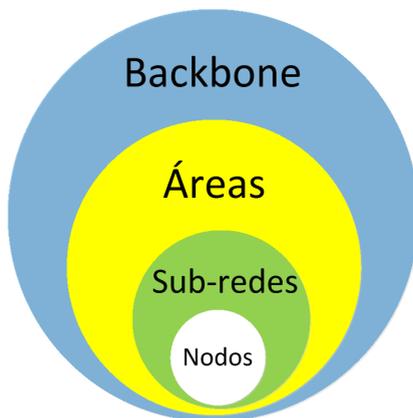


Figura 20 – Hierarquia de uma rede EIB / KNX.

709.1), que implementa todas as sete camadas do modelo de referência ISO/OSI (*International Organization for Standardization / Open Systems Interconnection*) e foi desenvolvido especialmente para aplicações que fazem uso de funções de controle, monitoração, identificação e sensoriamento (IGEI et al., 2007).

O LonWorks permite o uso de cabos elétricos, par trançado, cabo coaxial, infravermelho, radiofrequência, e fibra ótica como meios físicos de transmissão, o que permite uma rede de controle totalmente integrada, por exemplo, um dispositivo conectado através de par trançado, pode enviar uma mensagem para outro dispositivo que responde por meio do PLC.

Todos os dispositivos conectados à uma rede LonWorks tem agregado em si um pequeno microcontrolador especial, denominado Neuron *Chip*, e um *firmware* associado. O Neuron *Chip* incorpora comunicação, controle, gerenciamento e suporte de entrada e saída (IGEI et al., 2007).

É possível dividir os dispositivos de uma rede LonWorks em dois grupos:

- dispositivos de controle, que possuem memória e capacidade de processamento;
- dispositivos atuadores, que enviam informações pela rede para os nós que as necessitem.

Na Figura 21 é ilustrada uma rede LonWorks, dotada de componentes como sensores, atuadores e dispositivos computacionais.

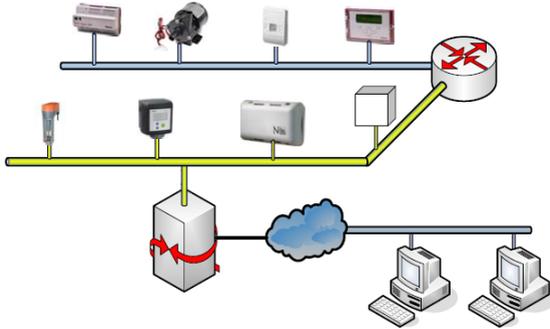


Figura 21 – Ilustração de uma típica rede LonWorks.
Extraído de (IGEI et al., 2007)

O LonWorks é um sistema aberto, isso permite que qualquer fabricante faça uso desta TAC sem depender de sistemas proprietários, o que permite reduzir custos e aumentar a flexibilidade do sistema implantado (HUANG; WAN; ZHOU, 2008).

3.4.2 TACs sem fio

Na área da automação residencial e predial, em sistemas e dispositivos em que as tecnologias sem fio se desenvolvem por meio da utilização da transmissão de sinais sensoriais, os padrões denominados ZigBee e *Z-Wave* apresentam-se de forma destacada no cenário internacional (DIAS; PIZZOLATO, 2004).

- **ZigBee**

Em 2002, um conjunto de grandes empresas de tecnologia, entre elas, Mitsubshi, Samsung, Motorola, Philips, fundam a Aliança ZigBee. A Aliança ZigBee é responsável pelo padrão ZigBee, que oferece soluções em redes sem fio e permite o gerenciamento e controle de dispositivos eletrônicos a um curto alcance e com uma boa relação custo benefício (HUANG et al., 2010).

O ZigBee foi desenvolvido para atender aplicações de baixo custo, com dispositivos que fazem uso de baterias, tais como controle

industrial e comercial, automação residencial e predial (ZAREEI et al., 2011).

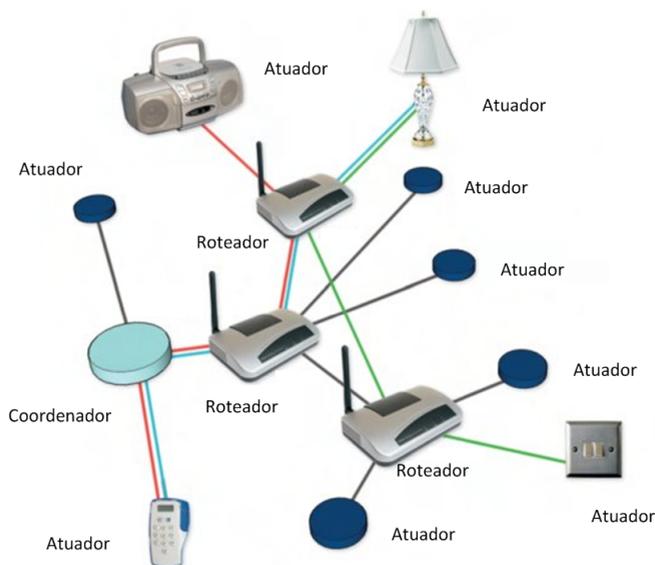


Figura 22 – Exemplo de rede ZigBee.
Adaptado de (GODOI, 2009)

ZigBee utiliza uma fração da capacidade de processamento necessário as redes Wi-Fi, e um décimo de memória se comparado ao Bluetooth. Essas características tornam o ZigBee a melhor solução para sistemas de comunicação com baixa taxa de dados (até 250 Kbps) e curto alcance (até 100 metros) (SUZUKI; MITANI; SHINOHARA, 2010).

O ZigBee possui mais características que o transformam em uma tecnologia simples e eficiente, como (HUANG et al., 2010):

- baixo consumo, adequado para dispositivos que fazem uso de baterias;
- operação em frequência não licenciadas (2,5 GHz, 868 MHz e 915 MHz);
- pode ser inserido em microcontroladores de baixo custo, devido ao seu protocolo simples e *hardware* reduzido;

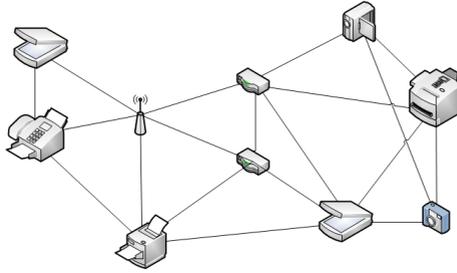


Figura 23 – Esquema simplificado da comunicação em malha.
Adaptado de (DIAS; PIZZOLATO, 2004)

- permite conectar centenas de dispositivos em uma rede e trabalha com várias topologias, como estrela, árvore ou malha.

Em uma rede ZigBee, representada na Figura 22, existem três tipos de dispositivos (MONSIGNORE, 2007): coordenador, responsável pela ligação entre os dispositivos; roteador, responsável por ampliar o alcance entre determinados atuadores; e atuadores, que podem ser lâmpadas, chaves, sensores de temperatura.

- **Z-Wave**

Z-Wave é visto como uma das tendências para sistemas de automação residencial, essa tecnologia estabelece a comunicação entre dispositivos via radiofrequência, foi desenvolvida em 2008 e licenciada pela Zensys (GODOI, 2009). Atualmente um consórcio de empresas é responsável por fabricar produtos que utilizam esse padrão, tais como, Intel, Universal *Electronics*, *Z-Wave*, Zensys e mais de uma centena de empresas. Esse consórcio é denominado *Z-Wave Alliance*.

O *Z-Wave* é uma tecnologia de baixo custo que transmite dados via *Wireless*, e foi desenvolvido para controlar remotamente dispositivos de baixa potência (GODOI, 2009). Este sistema opera na frequência de 900 MHz, que por consequência não interfere no sinal de outras tecnologias *Wireless*, como Wi-Fi e Bluetooth, e atinge velocidades de até 9.6 Kbps. O alcance é de aproximadamente 30 metros, porém, a maioria dos dispositivos *Z-Wave* possuem a função de amplificar o sinal, o que permite uma ampla cobertura dos ambientes, independente da presença de obstáculos (CSERNATH et al., 2008).

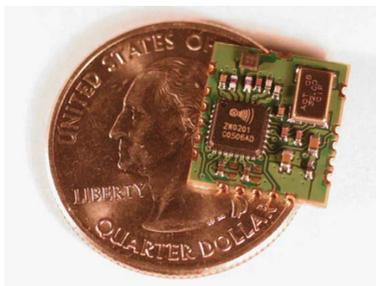


Figura 24 – Tamanho do *chipset* Z-Wave comparado com uma moeda de um quarto de dólar.

Adaptado de (KNIGHT, 2006)

Uma das características do *Z-Wave*, é o uso da topologia de rede em malha, representada na Figura 23, o que permite que um dispositivo X comande outros dispositivos para efetivar sua comunicação com Y . Em caso de interferência nas rotas de comunicação, seja por causa humana ou um simples abrir ou fechar de portas, a topologia em malha permite que automaticamente esse caminho seja revisto e o dado seja entregue ao seu destino por uma rota alternativa (ZAREEI et al., 2011).

A tecnologia Z-Wave possui uma diversidade de dispositivos em sua linha de produtos: controle remoto, controle de acesso, interruptores, interfaces e sensores, módulos de tomada e *chipset* (GODOI, 2009). Um *chipset* Z-Wave é ilustrado na Figura 24.

4 SISTEMA PARA ADAPTAÇÃO DE CONTEXTO EM AUTOMAÇÃO RESIDENCIAL

Este capítulo descreve o sistema de adaptação de contexto residencial. Para um melhor entendimento, as arquiteturas de *hardware* e *software* são descritas separadamente.

4.1 ARQUITETURA DE HARDWARE DO SISTEMA DE ADAPTAÇÃO DE CONTEXTO RESIDENCIAL

A arquitetura de *hardware* do sistema de adaptação de contexto residencial é composta por uma unidade central de controle, unidades locais de controle, e periféricos como sensores e dispositivos eletroeletrônicos.

A unidade central de controle é o servidor do sistema distribuído e é responsável por gerenciar e configurar os dados contextuais dos usuários. As unidades locais de controle são aplicações clientes que realizam o acionamento dos dispositivos baseados na leitura de sensores e em dados contextuais.

Uma visão geral arquitetura de *hardware*, com seus respectivos elementos, é ilustrada na Figura 25.

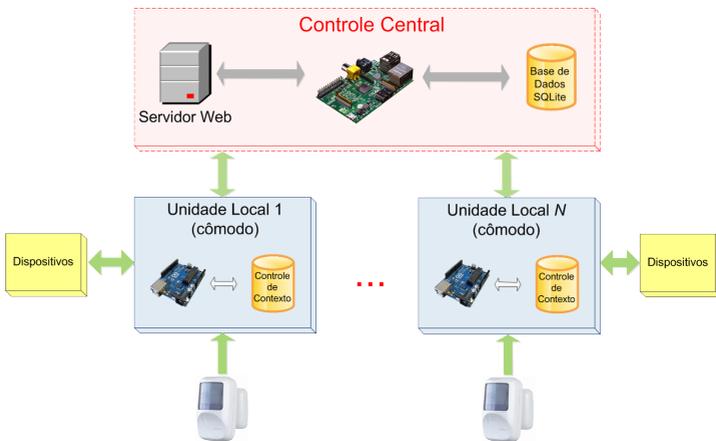


Figura 25 – Arquitetura de *hardware* do sistema.

As subseções 4.1.1 e 4.1.2 contêm descrições detalhadas sobre o *hardware* utilizado para compor as unidades de controle central e local.

4.1.1 Unidade de Controle Central

A unidade de controle central é um dispositivo computacional responsável por manipular uma base de dados que armazena os dados contextuais dos usuários, gerenciar e distribuir informações de contexto computacional para unidades de controle local, e executar um servidor *Web*, que provê uma interface para interação do usuário com o sistema. A unidade de controle central estabelece comunicação com as unidades locais via rede Ethernet.

A unidade de controle central é baseada em um Raspberry PI modelo B, um microcomputador desenvolvido na Grã-Bretanha de dimensões reduzidas, idealizado sob o paradigma *System On a Chip* (XUN et al., 2001).

O Raspberry PI (EDWARDS, 2013) tem capacidade de processamento suficiente para executar tarefas realizadas por computadores *desktop*, como processamento de jogos, edição de textos e planilhas, navegação na *internet*, entre outros. A Figura 26 ilustra um Raspberry PI modelo B.



Figura 26 – Raspberry PI modelo B.
Extraído de (FOUNDATION, 2013)

O Raspberry PI utiliza um processador ARM11 de 700 MHz e 512 MB de memória RAM (*Random Access Memory*), possui placa de rede *onboard*, duas saídas de vídeo (HDMI - *High-Definition Multimedia Interface* e RCA - *Radio Corporation of America*), saída de áudio, entrada para cartão SD (*Secure Digital*) e duas portas USB (*Universal Serial Bus*) 2.0.

O Raspberry PI pode executar o sistema operacional direto em

um cartão SD, o que torna desnecessário o uso um dispositivo de armazenamento de dimensões maiores que o próprio Raspberry PI como um HD (*Hard Disk*) externo, por exemplo. Contudo é possível acoplar ao Raspberry PI um dispositivo externo de armazenamento de dados.

O Raspberry PI pode ser substituído por qualquer computador convencional, como um *desktop* ou um *notebook*, pois todo o sistema foi implementado em Python, uma linguagem de programação interpretada e de alto nível que executa em qualquer dispositivo computacional que possua seu interpretador.

4.1.2 Unidade de Controle Local

A unidade de controle local é responsável por aplicar a adaptação de contexto em um cômodo da residência, ou seja, é responsável por configurar os dispositivos eletroeletrônicos de um determinado cômodo da residência. Além desta função, a unidade de controle local, por meio de sensores de presença, detecta a entrada e permanência de um usuário no cômodo da residência e monitora o contexto de cada dispositivo inserido em seu perímetro de atuação.

O número de unidades de controle local instalados em uma residência varia de acordo com o número de cômodos controlados. O *hardware* da unidade de controle local possui uma capacidade inferior de processamento se comparada à unidade de controle central, e possui as seguintes funcionalidades:

- solicita para a unidade de controle os dados contextuais de um usuário;
- configura os aparelhos dispostos no cômodo a partir dos dados contextuais recebidos;
- captura alterações das configurações dos aparelhos do cômodo;
- envia para a unidade de controle central os dados contextuais atualizados de cada usuário em seu cômodo.

O Arduino UNO foi escolhido para ser responsável pelo gerenciamento das funções da unidade local. O Arduino UNO é uma placa de prototipação *open-source* baseado em um microcontrolador muito versátil que permite o controle de vários dispositivos, e por essa característica é muito utilizado em aplicações de instrumentação embarcada e robótica. O Arduino UNO é ilustrado na Figura 27.



Figura 27 – Placa de prototipação Arduino UNO.
Extraído de (ARDUINO, 2013)

Para utilizar o Arduino foi necessário acoplar uma Ethernet Shield, uma placa que possibilita conexão à uma rede Ethernet através de uma entrada RJ-45. A Figura 28 ilustra um Arduino UNO com uma placa Ethernet Shield acoplada.



Figura 28 – Arduino UNO com um Ethernet Shield.
Extraído de (ARDUINO, 2013)

4.2 ARQUITETURA DE SOFTWARE DO SISTEMA DE ADAPTAÇÃO DE CONTEXTO RESIDENCIAL

A arquitetura de *software* conta com duas estruturas bem definidas, a estrutura de controle central e de controle local. Essas estruturas possuem um canal de comunicação bidirecional, que permite ambos os lados da aplicação distribuída enviar e receber mensagens. A Figura 29 ilustra a arquitetura de *software* do sistema de adaptação de contexto residencial.

As subseções 4.2.1 e 4.2.2 contêm descrições detalhadas sobre o

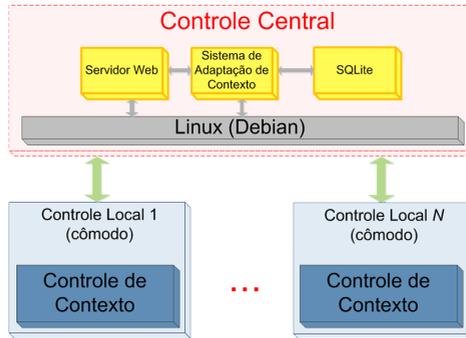


Figura 29 – Arquitetura de *software* do sistema.

controle central e controle local que compõem o sistema.

4.2.1 Controle Central

A estrutura de controle central é executada no sistema operacional Linux e é composta pelo sistema de adaptação de contexto (SAC), servidor *web* e uma base de dados.

- **Servidor Web**

O servidor *web* é responsável pela interação entre o usuário e o sistema, que pode ser acessada local ou remotamente via *browser*. Essa interface permite que o usuário, por meio de autenticação, controle a aplicação, acione comandos, ou se informe sobre o que ocorre dentro da residência diretamente do trabalho ou qualquer outro lugar que disponibilize acesso à internet, ou até mesmo em casa, por meio de um *display* estrategicamente posicionado.

- **Base de dados - SQLite**

A base de dados é responsável por armazenar os dados contextuais de cada usuário. É na base de dados onde estão armazenadas informações como, número de identificação e nome de cada usuário, um breve histórico contendo cada ambiente em que cada usuário se encontrava, e a informação de onde cada usuário se encontra, caso esteja dentro da residência.

Além das informações sobre usuários, a base de dados também armazena informações sobre os cômodos da residência que possuem

uma unidade local de controle. Para armazenar o contexto de um cômodo relacionado a determinado usuário, a base de dados armazena informações como o número de identificação do usuário, o número de identificação de cada dispositivo inserido no cômodo e suas respectivas configurações.

O SQLite foi escolhido para armazenar os dados contextuais pois é um banco de dados prático e consistente, recomendado para aplicações onde a simplicidade de implementação, manutenção e administração são prioridade.

O SQLite se diferencia de seus concorrentes pois é um banco de dados SQL (*Structured Query Language*) embutido, ou seja, não é um cliente para conexão com a base de dados, mas sim a própria base de dados. Essa característica do SQLite permite que ele atenda as necessidades requeridas pelo sistema de adaptação de contexto.

- **SAC**

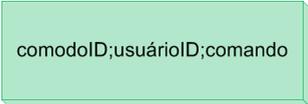
O SAC é responsável por elaborar a lógica da adaptação contextual. Essa lógica opera sobre duas matrizes, que são carregadas através das informações advindas de dois cômodos, o último cômodo onde o usuário se encontrava e o cômodo para onde ele irá. Essas matrizes são compostas pelos eixos “número de dispositivos e configuração de dispositivos”, que permitem a troca das informações necessárias para executar a adaptação de contexto.

Quando o SAC estabelece uma conexão com uma unidade de controle local, ele analisa se é viável executar a ação pretendida pela unidade local naquela conexão (requisitar ou atualizar dados contextuais), por meio de uma mensagem de cumprimento (*Handshake*) recebida via *socket*.¹

Essa mensagem de cumprimento é ilustrada na Figura 30, e contém o número de identificação do cômodo, o número de identificação do usuário, e o que a unidade local pretende fazer, enviar ou requisitar dados contextuais.

Caso o SAC atenda uma requisição de dados contextuais ele realizará a adaptação de contexto, compactando as informações em uma mensagem de contexto que será enviada pelo canal de comunicação para a unidade local de controle. Para receber dados contextuais, o SAC recebe a mensagem de contexto via *socket*, descompacta e armazena os valores atualizados na base de dados.

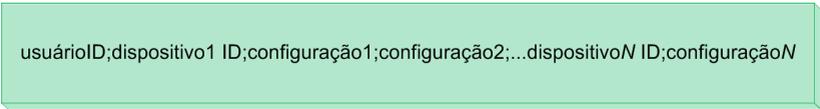
¹Socket - interface de comunicação entre processos.



```
comodoID;usuárioID;comando
```

Figura 30 – Formato da mensagem de cumprimento.

A Figura 31 ilustra uma mensagem de contexto, que contém o número de identificação do usuário, os números de identificação dos dispositivos e suas respectivas configurações.



```
usuárioID;dispositivo1 ID;configuração1;configuração2;...dispositivoN ID;configuraçãoN
```

Figura 31 – Formato da mensagem de contexto.

O SAC foi desenvolvido na linguagem de programação Python, uma linguagem de alto nível, multiparadigma, orientada a objetos. Por ser interpretada, programas desenvolvidos em Python podem executar em diversos sistemas operacionais.

4.2.2 Controle Local

O sistema que executa na unidade de controle local foi implementado na linguagem de programação Wiring, e controla uma estrutura interna que comporta dados para identificação do usuário e a configuração e identificação de cada dispositivo inserido em seu perímetro de atuação.

O controle local, a partir da leitura de alguns sensores, é responsável pelo reconhecimento do usuário no momento em que este entra no cômodo. Com a identificação do usuário o controle local gera uma mensagem de cumprimento, abre uma conexão com o controle central e por meio desta recebe uma mensagem de contexto com as informações do usuário.

Com os dispositivos configurados o controle local entra na fase de monitoramento para coletar dados contextuais dos dispositivos sob seu controle. A cada mudança de estado em um dispositivo o controle local atualiza sua estrutura de controle interna.

Ao perceber que o usuário se retirou do cômodo o controle local desliga os dispositivos de seu perímetro, abre uma conexão com o controle central e por meio da mensagem de cumprimento recebe a autorização para enviar os dados contextuais atualizados do usuário.

Com a permissão para enviar dados o controle local empacota sua estrutura interna de controle no formato de uma mensagem de contexto (conforme a Figura 31), e a encaminha para o controle central, que atualiza a base de dados contextuais.

Encerrada a comunicação com o controle central, o controle local executa em modo de espera, aguardando até que algum usuário se mova até sua área de abrangência.

5 AVALIAÇÃO DO SISTEMA

Este capítulo apresenta os resultados obtidos na avaliação do sistema desenvolvido. Para avaliar o sistema foram elaborados três experimentos, sendo: um experimento onde o usuário controlava o acionamento de um ventilador; um segundo experimento onde o usuário controlava o estado de um televisor, e por fim, o terceiro experimento onde o usuário controlava o sistema de luminosidade de dois cômodos.

5.1 EXPERIMENTO 1 - CONTROLE DE UM VENTILADOR

Este experimento teve como objetivo validar a integridade dos dados contextuais enviados pela unidade de controle local para a unidade de controle central. Para tanto fez-se necessário simular um ambiente contendo um único dispositivo, no caso um ventilador que inicialmente estava desligado.

Neste experimento o usuário alterou o estado do ventilador de desligado para ligado com uma determinada velocidade. Realizada essa etapa, o usuário se retira do cômodo, ao detectar esta situação, o controle local envia uma mensagem para o controle central informando o contexto computacional do cômodo.

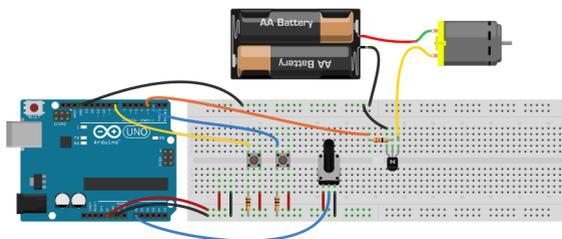


Figura 32 – Esquemático do circuito de controle do experimento 1.

O circuito ilustrado na Figura 32 foi montado em um *protoboard* contendo, dois botões que simulam a entrada e saída do usuário no cômodo, um potenciômetro, pilhas AA de 1,5 V e um pequeno motor, responsável por simular o efeito de um ventilador. O ventilador foi

conectado a uma porta PWM¹ (*Pulse Width Modulation*) do Arduino.

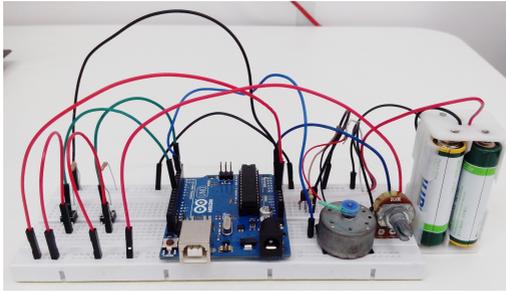


Figura 33 – Imagem do primeiro experimento montado no *protoboard*.

O potenciômetro permite que a velocidade do ventilador seja alterada em tempo de execução, com uma variação de 0 a 255, onde 0 indica que o ventilador se encontra desligado e 255 ligado na potência máxima. A Figura 33 ilustra a montagem do experimento no *protoboard*.

Além dos equipamentos citados, simulou-se também o comportamento de um usuário, que ao entrar no cômodo liga o ventilador na potência máxima, e após alguns instantes se retira do cômodo.

A Figura 34 ilustra os valores enviados e recebidos pela unidade de controle central durante o primeiro experimento.

```
Python 3.3.1 (v3.3.1:d9893d13c628, Apr 6 2013, 20:30:21) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Mensagem de Contexto inicial: '1001:104:0'
Mensagem de Contexto Final: '1001:104:255'
>>>
```

Figura 34 – Imagem da interface de saída do controle central durante o primeiro experimento.

No início do experimento o controle central enviou a mensagem de contexto com o número de identificação do usuário (valor 1001), número de identificação do ventilador (valor 104) e a configuração padrão do dispositivo (valor 0).

¹Porta PWM - Permite a variação da largura de pulsos analógicos em meios digitais.

No final do experimento o controle central recebeu as informações do usuário 1001 no cômodo controlado pelo controle local utilizado no experimento, essa mensagem de contexto continha a informação de que o ventilador (dispositivo 104) estava com valor de configuração 255, o que indicava que o mesmo encontrava-se ligado.

Após concluído o experimento constatou-se que o sistema garante a integridade dos dados contextuais enviados da unidade de controle local para unidade de controle central, e o valor de configuração final foi armazenado com êxito na base de dados contextuais.

5.2 EXPERIMENTO 2 - CONTROLE DE UM TELEVISOR

O experimento em questão teve como objetivo validar a configuração dos dispositivos eletroeletrônicos por uma unidade de controle local. Para tanto se fez necessário simular um cômodo que em seu interior continha um dispositivo que possuía mais de uma configuração, no caso um televisor.

Neste experimento o usuário teve a possibilidade de ligar e desligar, alterar os canais e volume do televisor. Realizada essa etapa, o usuário se retira do cômodo e após um determinado tempo retorna.

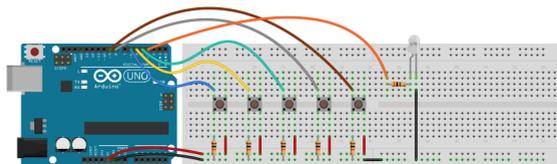


Figura 35 – Esquemático do circuito de controle do experimento 2.

Perante estas condições, ao perceber o retorno do usuário, a unidade de controle local configura o televisor de acordo com as informações armazenadas na base de dados contextuais.

O circuito ilustrado na Figura 35 foi montado em um *protoboard* com dois botões que simulam a entrada e saída do usuário no cômodo, três botões que permitem alterar o estado do televisor, e um emissor infravermelho.

O emissor infravermelho permite que os comandos acionados pelo usuário sejam codificados e executados pelo televisor. A Figura 36 ilustra a montagem do experimento no *protoboard*.

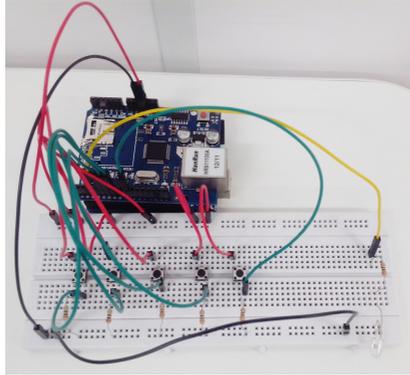


Figura 36 – Imagem do segundo experimento montado no *protoboard*.

Tabela 3 – Lista de canais disponíveis para o experimento.

Emissora	Canal
Canal 1	7
Canal 2	21
Canal 3	52

A Tabela 3 apresenta a lista de canais disponibilizados para o usuário no experimento.

```
Python 3.3.1 (v3.3.1:d9893d13c628, Apr 6 2013, 20:30:21) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ----- RESTART -----
>>>
Mensagem de Contexto Inicial: '1001;102;False;0;0'
Mensagem de Contexto Final: '1001;102;True;21;15'
```

Figura 37 – Imagem da interface de saída do controle central durante o segundo experimento.

A Figura 37 ilustra os valores enviados e recebidos pela unidade de controle central durante o segundo experimento.

No início do experimento o controle central enviou a mensagem de contexto com o número de identificação do usuário (valor 1001), número de identificação do televisor (valor 102) e as configurações padrão do dispositivo (desligado - *False*, canal 0 e com o volume 0).

No final do experimento o controle central recebeu as informações

Tabela 4 – Disposição das lâmpadas nos ambientes.

Lâmpada	Ambiente 1	Ambiente 2	Similaridade
Azul	Sim	Sim	Sim
Verde	Sim	Sim	Sim
Amarela	Sim	Sim	Sim
Laranja	Não	Sim	Não
Vermelha	Não	Sim	Não

do usuário 1001 no cômodo controlado pelo controle local utilizado no experimento, essa mensagem de contexto continha a informação de que o televisor (dispositivo 102) estava ligado (*True*) no canal 21 e com o volume 15.

Após realizado o experimento, constatou-se que o sistema em posse dos dados contextuais do usuário, garante a configuração dos dispositivos em determinado cômodo da residência.

5.3 EXPERIMENTO 3 - CONTROLE DE LUMINOSIDADE

O último experimento teve como objetivo validar a migração da configuração de dispositivos similares para o sistema. Essa característica de similaridade é designada pelo sistema para dispositivos idênticos, com o mesmo número de identificação.

Para a realização deste experimento se fez necessário simular dois ambientes com dispositivos similares. O primeiro ambiente foi configurado com três lâmpadas, que permitiam o controle da intensidade de seu brilho. O segundo ambiente contava com cinco lâmpadas, das quais três eram similares às lâmpadas do primeiro ambiente.

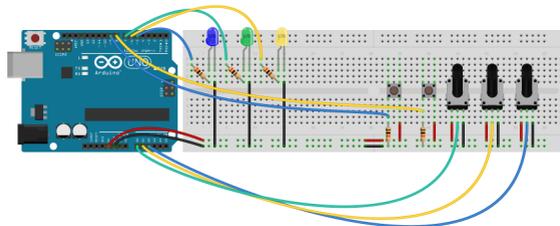


Figura 38 – Esquemático do circuito de controle do primeiro ambiente.

Cada lâmpada foi representada por um LED (*Light-emitting diode*) que possuía uma identificação e cor própria. A Tabela 4 informa a disposição das lâmpadas nos dois ambientes.

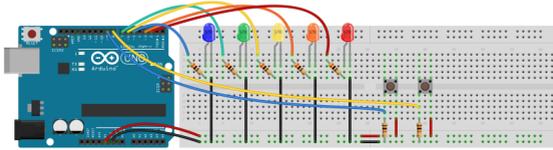


Figura 39 – Esquemático do circuito de controle do segundo ambiente.

Neste experimento o usuário alterou a intensidade de brilho das lâmpadas do primeiro ambiente. Após essa etapa, o usuário entra no segundo ambiente, e ao perceber a presença do usuário, a unidade de controle local do segundo ambiente aciona as lâmpadas de acordo com os dados contextuais da base de dados.

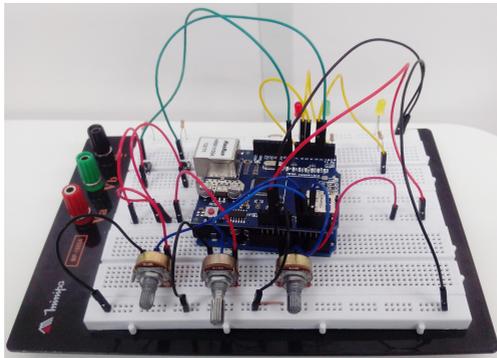


Figura 40 – Imagem do primeiro cômodo do terceiro experimento montado no *protoboard*.

O circuito ilustrado na Figura 38 foi montado em um *protoboard* contendo dois botões que simulam a entrada e saída do usuário no primeiro cômodo, três LEDs e três potenciômetros.

O circuito ilustrado na Figura 39 foi montado em um *protoboard* contendo dois botões que simulam a entrada e saída do usuário no segundo cômodo e cinco LEDs.

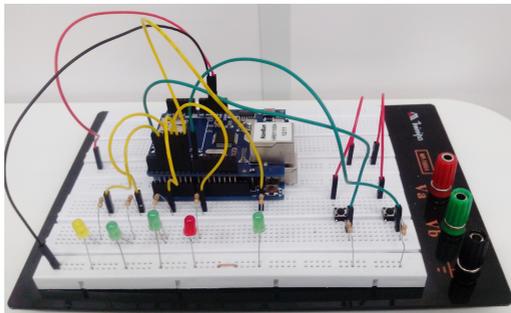


Figura 41 – Imagem do segundo cômodo do terceiro experimento montado no *protoboard*.

```
Python 3.3.1 (v3.3.1:d9893d13c628, Apr 6 2013, 20:30:21) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Mensagem de Contexto Inicial do Primeiro Ambiente: '1001;106;0;107;0;108;0'
Mensagem de Contexto Final do Primeiro Ambiente: '1001;106;125;107;217;108;20'
Mensagem de Contexto Inicial do Segundo Ambiente: '1001;106;125;107;217;108;20;109;0;110;0'
Mensagem de Contexto Final do Segundo Ambiente: '1001;106;125;107;217;108;20;109;0;110;0'
```

Figura 42 – Imagem da interface de saída do controle central.

A Figura 40 ilustra a montagem do primeiro ambiente, enquanto a Figura 41 ilustra a montagem do segundo ambiente no *protoboard*.

A Figura 42 ilustra as mensagens de contexto trocadas entre o controle central e as duas unidades de controle local durante o experimento.

No início da primeira parte do experimento o controle central enviou a mensagem de contexto para o primeiro cômodo com o número de identificação do usuário (valor 1001), número de identificação de cada lâmpada (valores 106, 107 e 108) e as configurações padrão dos dispositivo (valor 0).

No final da primeira parte do experimento o controle central recebeu a mensagem de contexto do primeiro cômodo que informava que o usuário 1001 deixou as lâmpadas com as seguintes configurações: lâmpada 106 - brilho 125, lâmpada 107 - brilho 217, lâmpada 108 - brilho 20.

No início da segunda parte do experimento o controle central

enviou a mensagem de contexto para o segundo cômodo com o número de identificação do usuário (valor 1001), número de identificação de cada lâmpada (valores 106, 107, 108, 109 e 110) e as configurações dos dispositivos armazenadas na base de dados (valores 125, 214, 20, 0 e 0).

No final do experimento o controle central recebeu a mensagem de contexto do segundo cômodo que informava que o usuário 1001 deixou as lâmpadas com as seguintes configurações: lâmpada 106 - brilho 125, lâmpada 107 - brilho 217, lâmpada 108 - brilho 20, lâmpada 109 - brilho 0 e lâmpada 110 - brilho 0.

Após concluído o experimento constatou-se que o sistema migra a computação entre dispositivos similares com êxito.

6 CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS

Na automação residencial tradicional o processo de automação se dá apenas quando o usuário aciona previamente um comando, entretanto as aplicações sensíveis ao contexto permitem que o ambiente comporte-se de maneira inteligente, antecipando-se às situações com soluções que beneficiam seus usuários.

Para tornar a residência proativa ao invés de apenas reativa, foi desenvolvido um sistema de controle distribuído que levasse em consideração não apenas informações advindas de um único cômodo, mas também informações de todos os dispositivos de uma residência.

A partir do uso de sensores é possível mapear as características, necessidades e costumes dos usuários. E um sistema que consiga fazer uso dessas informações pode atuar de maneira proativa, oferecendo maior conforto para os habitantes da residência.

Este trabalho apresentou um sistema de adaptação de contexto residencial, desenvolvido para permitir que uma residência reconheça seus usuários e adapte seus ambientes internos de acordo com as preferências dos mesmos, por meio de um sistema eficiente e confiável que move todo o contexto computacional de um ambiente junto com o usuário, cômodo por cômodo.

Para comprovar a eficiência do sistema foram realizados três experimentos de bancada com níveis de dificuldade gradativos, onde foi possível adaptar a velocidade de um ventilador, alterar o estado de um televisor e configurar o sistema de luminosidade de determinados ambientes.

Perante estes experimentos o sistema se comportou de maneira satisfatória, e comprovou a comunicação entre as aplicações cliente e servidor, mesmo com um pequeno número de troca de mensagens, o sistema realizou de forma eficiente a adaptação do contexto.

Por fim, concluiu-se que o sistema de adaptação de contexto residencial permitiu que ambientes atuassem de maneira proativa, beneficiando o usuário, permitindo que este se concentre na realização de outras tarefas.

6.1 PROPOSTAS PARA TRABALHOS FUTUROS

Nesta seção são listadas algumas propostas para trabalhos futuros.

1. Utilizar um *middleware* orientado à aplicações ubíquas para permitir que o sistema trabalhe com dados provenientes de sensores de modo transparente;
2. Implementar o algoritmo ID3 para que o sistema possa inferir soluções mais inteligentes;
3. Adicionar um sistema de visualização 3D (*Three Dimensional*) para permitir que usuários visualizem o que ocorre na residência em tempo real;
4. Utilizar comunicação sem fio para realizar testes e comparar a eficiência do sistema com e sem cabeamento;
5. Implementar reconhecimento automático de múltiplos usuários por meio de um sistema captação de imagens em tempo real.

REFERÊNCIAS

AMORY, A. M.; JÚNIOR, J. P. Sistema integrado e multiplataforma para controle remoto de residências. In: *VII Iberchip Workshop*. Montevideo - Uruguai: [s.n.], 2001. p. 1 – 10.

ARAUJO, L. Preza de. *Domótica: Protocolo De Comunicação X-10*. 2011. <<http://www.prof2000.pt/users/lpa/x10.ppt>>. Acessado em 31/05/2013.

ARAUJO, R. Borges de. Computação ubíqua: Princípios, tecnologias e desafios. In: UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE. *XXI Simpósio Brasileiro de Redes de Computadores*. Rio Grande do Norte - Brasil, 2003.

ARDUINO. *Products*. 2013. <<http://arduino.cc>>.

BEHAR, P. A.; COSTA, A. C. da R. Computação cooperativa no processo de construção coletiva de conhecimentos. In: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. *II Congresso Ibero-americano de Informática na Educação*. Porto Alegre, 1996.

BORONOWSKY, M. et al. Wearable computing: an approach for living labs. In: *Applied Wearable Computing (IFAWC), 2006 3rd International Forum on*. [S.l.: s.n.], 2006. p. 1–8.

BRUNETTE, E. S.; FLEMMER, R.; FLEMMER, C. L. A review of artificial intelligence. In: *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*. [S.l.: s.n.], 2009. p. 385–392.

CACERES, R.; FRIDAY, A. UbiComp systems at 20: Progress, opportunities, and challenges. *Pervasive Computing, IEEE*, v. 11, n. 1, p. 14–21, 2012. ISSN 1536-1268.

CHAMUSCA, A. *Domótica e Segurança Electrónica : a inteligência que se instala*. Portugal: Ingenium Edições, 2006.

CIRILO, C. E. Computação ubíqua: definição, princípios e tecnologias. In: UFSCAR. *UFSCar*. São Carlos, 2010. p. 1 – 10.

CORREA, S. L. Estudo comparativo de middlewares para computação ubíqua. In: UNIVERSIDADE FEDERAL DE GOIÁS. *Universidade Federal de Goiás*. Goiás, 2006.

COSTA, C. A. da. Um estudo de arquiteturas de software para o desenvolvimento de aplicações pervasivas. In: *Unisinos*. [S.l.: s.n.], 2006.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, J. *Sistemas Distribuídos: Conceitos e Projetos*. 4. ed. Porto Alegre: Bookman Companhia Editora Ltda, 2007. 790 p.

CSERNATH, G. et al. A novel ecg telemetry and monitoring system based on z-wave communication. In: *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. [S.l.: s.n.], 2008. p. 2361–2364. ISSN 1557-170X.

DIAS, C. L. de A.; PIZZOLATO, N. D. Domótica: Aplicabilidade e sistemas de automação residencial. In: *Vértices*. [S.l.: s.n.], 2004. v. 6, n. 3, p. 9 – 32.

DOMINGUES, F. L. Computação ubíqua. In: GDH PRESS. *Guia do Hardware*. [S.l.], 2008.

DONG, Y.; ZHANG, B.; DONG, K. An integrated plc smart home system in pervasive computing environment. *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, v. 1, p. 1 – 4, 2010.

EDWARDS, C. Not-so-humble raspberry pi gets big ideas. *Engineering Technology*, v. 8, n. 3, p. 30–33, 2013. ISSN 1750-9637.

FERREIRA, V. Z. G. *A Domótica Como Instrumento para a Melhoria da Qualidade de Vida dos Portadores de Deficiência*. Dissertação (Monografia) — Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, João Pessoa, 2010.

FOUNDATION, T. R. P. *Raspberry PI*. 2013.
<<http://www.raspberrypi.org>>. Acessado em 15/06/2013.

FREITAS, S. A. G. de. *Desenvolvimento de um Sistema de Controle Embarcado para a Equipe de Futebol de Robôs Araranguá Intruders*. Dissertação (Monografia) — Universidade Federal de Santa Catarina, Santa Catarina, Fevereiro 2013.

GIROUX, S.; PIGOT, H. *From Smart Home to Smart Care*. [S.l.]: IOS Press, 2005.

GODOI, R. S. *Automação de Residências: Uma Análise da Viabilidade da Aplicação da Domótica*. Dissertação (Monografia) — Faculdade de Tecnologia de Guaratinguetá, Guaratinguetá, 2009.

GOMES, A. R. *UbiquitOS - Uma proposta de arquitetura de middleware para a adaptabilidade de serviços em sistemas de computação ubíqua*. Dissertação (Dissertação) — Universidade de Brasília - Instituto de Ciências Exatas, Distrito Federal, 2007.

GRIMM, R. One.world: experiences with a pervasive computing architecture. *Pervasive Computing, IEEE*, v. 3, n. 3, p. 22–30, 2004. ISSN 1536-1268.

HAGEN, A.; CONNORS, D.; PELLOM, B. The analysis and design of architecture systems for speech recognition on modern handheld-computing devices. In: *Hardware/Software Codesign and System Synthesis, 2003. First IEEE/ACM/IFIP International Conference on*. [S.l.: s.n.], 2003. p. 65–70.

HUANG, J. et al. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In: *Network Protocols (ICNP), 2010 18th IEEE International Conference on*. [S.l.: s.n.], 2010. p. 305–314. ISSN 1092-1648.

HUANG, Y.; WAN, C.; ZHOU, Z. Intelligent community system based on lonworks technology. In: *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*. [S.l.: s.n.], 2008. v. 1, p. 237–240.

HUGUES, J.; PAUTET, L.; KORDON, F. Contributions to middleware architectures to prototype distribution infrastructures. In: *Rapid Systems Prototyping, 2003. Proceedings. 14th IEEE International Workshop on*. [S.l.: s.n.], 2003. p. 124–131. ISSN 1074-6005.

IBRAHIM, N. Orthogonal classification of middleware technologies. In: *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBIComm '09. Third International Conference on*. [S.l.: s.n.], 2009. p. 46–51.

IGEI, P. J. et al. Modelagem de redes de controle lonworks em edifícios inteligentes. In: *8º Congresso Iberoamericano de Engenharia Mecânica*,. Cusco - Perú: [s.n.], 2007.

INITIATIVE, A. H. R. *Research Areas*. Georgia, 2000.
<<http://www.awarehome.gatech.edu/>>. Acessado em 10/06/2013.

INSTITUTE, G. T. B. *Georgia Tech's Aware Home Research Initiative*. 1999. <<http://www.cc.gatech.edu/fce/house/house.html>>. Acessado em 10/06/2013.

JUNIOR, P. J. Computação, ubiquidade e transparência. In: UNIANCHIETA. *Ubiquidade*. São Paulo, 2011. v. 1, n. 1.

KAHL, M.; FLORIANO, D. Computação ubíqua, tecnologia sem limites. In: UDESC. *UDESC CEAVI*. Santa Catarina, 2011.

KIDD, C. D. et al. The aware home: A living laboratory for ubiquitous computing research. In: *CoBuild*. [S.l.]: Springer, 1999. (Lecture Notes in Computer Science, v. 1670), p. 191–198. ISBN 3-540-66596-X.

KNIGHT, M. Wireless security - how safe is z-wave? *Computing Control Engineering Journal*, v. 17, n. 6, p. 18–23, 2006. ISSN 0956-3385.

MANFIO, E. R. Como funcionam alguns fonemas no aplicativo balabolka. In: VIA LITTERAE. *Via Litterae - Revista de Linguística e Teoria Literária*. Anápolis, 2012. v. 4, n. 2, p. 191 – 204.

MARIOTONI, C. A.; JR., E. P. A. Descrição de sistemas de automação predial baseados em protocolos plc utilizados em edifícios de pequeno porte e residências. In: *Revista de Automação e Tecnologia de Informação*. [S.l.: s.n.], 2002. v. 1, n. 1.

MCCRICKARD, D. S.; WRIGHTON, D.; BUSSERT, D. Supporting the construction of real world interfaces. In: *Human Centric Computing Languages and Environments, 2002. Proceedings. IEEE 2002 Symposia on*. [S.l.: s.n.], 2002. p. 54–56.

MEIRA, A. B.; DELFINO, S. R.; ORLANDINI, G. Domótica utilizando softwares e hardware livres. In: FACULDADES INTEGRADAS DE OURINHOS. *X Congresso de Iniciação Científica*. Ourinhos, 2011.

MONSIGNORE, F. *Sensoriamento de Ambiente Utilizando o Padrão ZigBee*. Dissertação (Dissertação) — Universidade de São Paulo, São Carlos, 2007.

MOSTEFAOUI, G.; PASQUIER-ROCHA, J.; BREZILLON, P. Context-aware computing: a guide for the pervasive computing community. In: *Pervasive Services, 2004. ICPS 2004. Proceedings. The IEEE/ACS International Conference on*. [S.l.: s.n.], 2004. p. 39–48.

MOZER, M. The adaptive house. In: *Intelligent Building Environments, 2005. The IEE Seminar on (Ref. No. 2005/11059)*. [S.l.: s.n.], 2005. p. 39–79. ISSN 0537-9989.

MURATORI, J. R.; BO, P. H. D. Automação residencial: Histórico, definições e conceitos. In: *O Setor Elétrico*. São Paulo: [s.n.], 2011. p. 70 – 76. Mensal.

PINHEIRO, J. M. S. *Falando de Automação Predial*. 2004. <<http://www.projetoederedes.com.br/artigos/>>. Acessado em 22/03/2013.

PIVOT. *Museum Objects: The First Mobile Phone Call*. 2013. <<http://www.pivotdesigngroup.com>>. Acessado em 22/03/2013.

ROMAN, M. et al. A middleware infrastructure for active spaces. *Pervasive Computing, IEEE*, v. 1, n. 4, p. 74–83, 2002. ISSN 1536-1268.

RUTA, M. et al. A semantic-based evolution of eib konnex protocol standard. In: *Mechatronics (ICM), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 773–778.

SAMSUNG. *Samsung Galaxy S3*. 2013. <<http://www.samsung.com>>. Acessado em 22/03/2013.

SANTANA, E. F. Z. et al. Modelagem específica de domínio em linhas de produto de software na computação ubíqua. *III Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*, v. 1, p. 1 – 14, 2009.

SANZ, A.; GARCIA-NICOLAS, J.; ESTOPINAN, P. A complete node for power line communications in a single chip. In: *Power Line Communications and Its Applications, 2005 International Symposium on*. [S.l.: s.n.], 2005. p. 285–289.

SGARBI, J. A. *Domótica Inteligente: Automação Residencial Baseada Em Comportamento*. Dissertação (Dissertação) — Centro Universitário da Fundação Educacional Inaciana, São Bernardo do Campo, 2007.

SHAFER, S. et al. The new easyliving project at microsoft research. In: DARPA/NIST. *DARPA/NIST Smart Spaces Workshop*. Gaithersburg, Maryland, 1998.

SHAHNASSER, H.; WANG, Q. Controlling industrial devices over tcp/ip by using lonworks. In: *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE*. [S.l.: s.n.], 1998. v. 2, p. 1309–1314 vol.2.

SIATRON. *Test Drive*. 2013. <<http://www.x10brasil.com.br/>>. Acessado em 15/06/2013.

SILVA, I. V. F. da; CARVALHO, S. S. de. Domótica: Uma abordagem sobre redes, protocolos e soluções microprocessadas de baixo custo. In: *Revista Científica Semana Acadêmica*. Fortaleza: [s.n.], 2011. Trimestral.

SMISEK, J.; JANCOSSEK, M.; PAJDLA, T. 3d with kinect. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 1154–1160.

SONY. *Smartwatch*. 2013. <<http://www.sonymobile.com/br/>>. Acessado em 15/06/2013.

STARNER, T. Project glass: An extension of the self. *Pervasive Computing, IEEE*, v. 12, n. 2, p. 14–16, 2013. ISSN 1536-1268.

SUZUKI, N.; MITANI, T.; SHINOHARA, N. Study and development of a microwave power receiving system for zigbee device. In: *Microwave Conference Proceedings (APMC), 2010 Asia-Pacific*. [S.l.: s.n.], 2010. p. 45–48.

TAKIUCHI, M.; MELO, E.; TONIDANDEL, F. Domótica inteligente: Automação baseada em comportamento. In: *CBA 2004-XV Congresso Brasileiro de Automática*. Rio Grande do Sul: [s.n.], 2004.

TEJANI, D.; AL-KUWARI, A.; POTDAR, V. Energy conservation in a smart home. In: *Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on*. [S.l.: s.n.], 2011. p. 241–246.

TELMOTICA. *Soluções Inteligentes*. 2013.

<<http://www.telmotica.pt/>>. Acessado em 22/03/2013.

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

XUN, Y. et al. A platform for system-on-a-chip design prototyping. In: *ASIC, 2001. Proceedings. 4th International Conference on*. [S.l.: s.n.], 2001. p. 781–784.

YEE, G. Using privacy policies to protect privacy in ubicomp. In: *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*. [S.l.: s.n.], 2005. v. 2, p. 633–638 vol.2. ISSN 1550-445X.

YUEJUN, Z.; MINGGUANG, W. Design of wireless remote module in x-10 intelligent home. In: *Industrial Technology, 2005. ICIT 2005. IEEE International Conference on*. [S.l.: s.n.], 2005. p. 1349–1353.

YUXUN, L.; NIUNIU, X. Improved id3 algorithm. In: *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*. [S.l.: s.n.], 2010. v. 8, p. 465–468.

ZAREEI, M. et al. A comparative study of short range wireless sensor network on high density networks. In: *Communications (APCC), 2011 17th Asia-Pacific Conference on*. [S.l.: s.n.], 2011. p. 247–252.