

Universidade Estadual de Goiás
Unidade Universitária de Itaberaí
Sistemas de Informação

COMPUTAÇÃO UBÍQUA APLICADA A AMBIENTES INTELIGENTES

João Paulo Silva do Vale

Itaberaí
2010

João Paulo Silva do Vale

COMPUTAÇÃO UBÍQUA APLICADA A AMBIENTES INTELIGENTES

Trabalho de Conclusão de Curso, apresentado a Unidade
Universitária de Itaberaí da Universidade Estadual de Goiás,
para obtenção do título de Bacharel em Sistemas de
Informação.

Orientador (a)
Prof^ª. Walkíria Nascente Valle

Itaberaí
2010

Universidade Estadual de Goiás
Unidade Universitária de Itaberaí
Sistemas de Informação

COMPUTAÇÃO UBÍQUA APLICADA A AMBIENTES INTELIGENTES

Trabalho de Conclusão de Curso, apresentado a Unidade
Universitária de Itaberaí da Universidade Estadual de Goiás,
para obtenção do título de Bacharel em Sistemas de
Informação.

Aprovado por:

**Walkíria Nascente Valle, UEG
(ORIENTADOR)**

**Lauriana Moreira, UEG
(EXAMINADOR)**

**Eliane Cristina, UEG
(EXAMINADOR)**

Itaberaí, 11 de dezembro de 2010.

Dedico esse trabalho a Deus e a minha família que sempre me apoiaram durante esse meu trajeto e me deram coragem para nunca desistir diante das dificuldades que se revelaram.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me concedido a oportunidade do acesso ao conhecimento e a educação e por ter me agraciado com uma família maravilhosa que sempre me apoiaram e deram suporte para que eu trilhasse os caminhos certos buscando meus objetivos através da educação.

Aos meus pais, Pedro Cândido do Vale e Terezinha Divina do Vale, aos quais devo tudo que hoje sou pois através de seus ensinamentos compreendi que a vida é cheia de bons e maus momentos e não devemos nunca desistir diante das dificuldades.

À minha orientadora, Walkíria Nascente Valle, que me ajudou com a elaboração desse trabalho e a todos os meus amigos que me apoiaram mesmo nas vezes em que tive de dizer “não” a eles por estar ocupado com os trabalhos acadêmicos.

Obrigado a todos!

RESUMO

A computação ubíqua surgiu da integração entre várias tecnologias, elaborando assim um novo seguimento computacional que defende a idéia de que os usuários ao necessitarem da utilização de serviços computacionais não se preocupem com as operações a serem realizadas com o computador e voltem suas atenções ao que lhes realmente interessa que são as tarefas a serem cumpridas naquele dado momento.

Com base nas premissas da computação ubíqua surgiram os ambientes inteligentes, assunto principal deste trabalho, que tem como característica evitar que o usuário necessite ir até um dispositivo ou computador para fazer uso dos mesmos, os ambientes inteligentes são um conjunto de tecnologias que trabalham integradamente permitindo o entendimento da situação e realizando operações ou comandos previamente programados para a execução de alguma tarefa sem que seja necessário a intervenção do usuário. Este trabalho ajuda e incentiva os usuários de diferentes níveis a entender melhor as mais variadas aplicações que a computação ubíqua pode prover.

PALAVRAS-CHAVE: Computação ubíqua, ambientes inteligentes, dispositivos inteligentes, redes de sensores, protocolos de descoberta de serviços, consciência de contexto, computação pervasiva, desenvolvedores, interface.

ABSTRACT

Ubiquitous computing has emerged from the integration of various technologies, thus developing a new tracking computer that supports the idea that users need to use computer-services should not worry about the operations to be performed with the computer and turn your attention to what they what really matters are the tasks to be accomplished at any given moment.

Based on the assumptions of ubiquitous computing environments have emerged smarter, the main subject of this work, which has the feature so the user need to go to a device or computer to make use of them, the smart environments are a set of enabling technologies that work in an integrated understanding the situation and performing operations or commands that were programmed to perform some task without requiring user intervention. This work helps and encourages users of different levels to better understand the various applications that ubiquitous computing can provide.

KEYWORDS: Ubiquitous Computing, smart environments, intelligent devices, sensor networks, protocols, service discovery, context awareness, pervasive computing, developers, interface.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

1. INTRODUÇÃO.....	13
2. COMPUTAÇÃO UBÍQUA.....	16
2.1 PARADIGMAS DA COMPUTAÇÃO UBÍQUA.....	18
2.2 LIMITAÇÕES DA COMPUTAÇÃO UBÍQUA ATUAL.....	20
2.3 MARK WEISER: O VISIONÁRIO.....	22
3. AMBIENTES INTELIGENTES.....	23
3.1 AMBIENTES CIENTES DE CONTEXTO.....	26
3.2 INTERAÇÃO HUMANO-COMPUTADOR (IHC).....	28
3.3 INTERFACES.....	31
4. AS REDES WIRELESS (REDES SEM FIO).....	33
4.1 PROTOCOLOS DE DESCOBERTA DE SERVIÇO.....	34
4.1.1 SERVICE LOCATION PROTOCOL (SLP).....	34
4.1.2 JINI.....	35
4.1.3 SALUTATION.....	36
4.1.4 UNIVERSAL PLUG AND PLAY (UPNP).....	36
4.1.5 BLUETOOTH SERVICE DISCOVERY PROTOCOL (SDP).....	37
4.1.6 INTENTIONAL NAMING SYSTEM (INS).....	37
5. DISPOSITIVOS INTELIGENTES.....	38
5.1 UTENSÍLIOS INTELIGENTES.....	38
5.2 CONTROLES INTELIGENTES.....	39
5.3 DISPOSITIVOS DE ACESSO À INFORMAÇÃO.....	40
5.3.1 ASSISTENTE DIGITAIS PESSOAIS (PDAS).....	40
5.3.2 COLETORES DE DADOS.....	41
5.3.3 TELEFONES INTELIGENTES.....	41
5.3.4 TABLET PCS.....	42
5.4 SISTEMAS DE ENTRETENIMENTO.....	43
6. AS REDES DE SENSORES.....	44
6.1 REDES DE SENSORES SEM FIO.....	44
6.2 CARACTERÍSTICAS DAS REDES DE SENSORES SEM FIO (RSSFS).....	45
6.3 MANUTENÇÃO DAS RSSFS.....	46

6.4 COLETA DE DADOS.....	47
7. SOFTWARES PARA AMBIENTES INTELIGENTES.....	48
7.1 SISTEMAS OPERACIONAIS (SO).....	51
7.1.1 PALM OS.....	52
7.1.2 BLACKBERRY OS.....	53
7.1.3 SYMBIAN OS.....	53
7.1.4 WINDOWS MOBILE.....	54
8. SEGURANÇA.....	55
9. CONSIDERAÇÕES FINAIS.....	58
10. REFERÊNCIAS BIBLIOGRÁFICAS.....	60

LISTA DE ILUSTRAÇÕES

Figura 1: Integração entre computação pervasiva e computação móvel originando a computação ubíqua.

Figura 2: Dimensões da Computação Ubíqua.

Figura 3: Arquitetura de automóveis inteligentes

Figura 4: Leitor facial anti-bêbado (Veículo Nissan)

Figura 5: Rede Wireless LAN típica

Figura 6: Classificação dos dispositivos ubíquos (Adaptado de Hansmann, 2001)

Figura 7: Exemplos de PDAs

Figura 8: Exemplo de coletores de dados

Figura 9: Exemplos de telefones inteligentes

Figura 10: Exemplos de Tablet PCs

Figura 11: Palm OS em execução

Figura 12: Interfaces Symbian OS

LISTA DE ABREVIATURAS E SIGLAS

UBICOMP – *Computação Ubíqua*
TI – *Tecnologia Da Informação*
PC – *Personal Computer*
PADs – *Passive Asset Detection System*
MP3 – *MPEG Layer 3*
GPS – *Global Positioning System*
RPM – *Rotações Por Minuto*
GSM - *Global System for Mobile Communication*
CDMA – *Code Division Multiple Access*
LCD – *Liquid Crystal Display*
IHC – *Interação Humano-Computador*
RFID - *Radio-Frequency Identification*
APIs - *Application Programming Interface*
SC – *Smart Card*
OCF - *Framework Open Card*
PDA – *Personal Digital Assistant*
WLAN – *Wireless Local Area Network*
SDPs - *Protocolos De Descoberta De Serviços*
SLP - *Service Location Protocol*
IETF - *Internet Engineering Task Force*
TCP – *Transmission Control Protocol*
IP – *Internet Protocol*
UA - *Agentes Usuários*
AS - *Agentes de Serviços*
DA - *Agentes de Diretório*
URL – *Uniform Resource Locator*
LS - *Serviço de Consultas*
SLM - *Salutation Manager*
UPnP - *Universal Plug And Play*
IEEE - *Institute of Electrical and Electronics Engineers*
IrDA - *Infrared Data Association*
SDP - *Bluetooth Service Discovery Protocol*

INS - *Intentional Naming System*

RSSFs - *Redes De Sensores Sem Fios*

OMG - *Object Managment Group*

MDA - *Model Driven Architecture*

SO - *Sistemas Operacionais*

W/CE - *Consumer Electronics*

CPU - *Central Processing Unit*

RIM - *Research in Motion*

SDK - *Software Development Kit*

1. INTRODUÇÃO

Ao longo dos anos os recursos, funcionalidades e disponibilidades dos serviços computacionais aumentaram significativamente proporcionando maior comodidade e reduzindo os esforços para os usuários finais, entretanto esses serviços tornam-se cada vez mais imperceptíveis.

Mark Weiser, cientista chefe do centro de pesquisa Xerox PARC, havia previsto essa condição em seu artigo “O Computador do Século 21”, publicado em 1991. Foi graças a Weiser que as pesquisas relacionadas à computação ubíqua ganharam destaque, a palavra ubíqua é derivada do latim *ubiquu* e significa “o que esta ao mesmo tempo e em toda parte”. Baseado nessa tese Weiser realizou importantes estudos no campo da computação ubíqua, que passou a ser usada em áreas bem diferentes e com propósitos diversos.

Foram vários os fatores que contribuíram para a expansão da computação ubíqua, entre eles podemos citar a evolução da *web*, o desenvolvimento de aplicações que interajam com os usuários, interfaces mais amigáveis, *hardwares* com capacidade para suportar as mais diversas aplicações, as redes sem fio que permitem uma comunicação mais rápida e fácil entre os usuários, além de inúmeros outros benefícios que faz com que a computação ubíqua seja utilizada em vários ambientes.

A *Web* já se tornou uma plataforma sólida onde aplicações podem ser feitas de forma rápida e eficiente. Com o aumento de velocidade da Internet e o acesso à banda larga, as tecnologias para a *Web* continuam evoluindo para proporcionar novas experiências aos usuários e aumentar a utilidade e a qualidade das aplicações (Loosley, 2006).

Mark Weiser previu em 1998, que no futuro, a computação estaria presente nos mais simples objetos: etiquetas de roupas, canetas, interruptores de luz, copo, botão, utensílios domésticos, etc., de forma invisível para o indivíduo.

Por ser uma tecnologia tão abrangente a computação ubíqua é aplicada em várias áreas, por exemplo, no desenvolvimento de ambientes inteligentes que é o foco deste trabalho. No desenvolvimento de ambientes inteligentes o objetivo é produzir formas de evitar que os usuários se desloquem para ir de encontro aos computadores e dispositivos informatizados e fazer com que os elementos que integram o ambiente possam ser utilizados e combinados entre si, fornecendo através de aplicações pré-programadas opções para que o usuário possa realizar suas atividades mesmo sem perceber a presença de dispositivos computacionais espalhados pelo meio.

Baseando-se em um amplo sentido os ambientes inteligentes são um conjunto de tecnologias que trabalham de maneira integrada, permitindo o entendimento de algumas situações e ativando instruções ou comandos pré-programados sem a necessidade de instruções diretas dos usuários. Um exemplo de ambientes inteligentes é quando dispositivos computacionais podem captar a presença humana em uma residência ascendendo ou apagando as luzes na presença ou ausência dos moradores, respectivamente. A computação ubíqua ou *UbiComp* necessita de avanços na capacidade de comunicação entre humanos e computadores, tornando o processo o mais natural possível. Para tanto, o desenvolvimento de interfaces naturais é peça fundamental neste processo porque assim à interação entre os usuários e os ambientes inteligentes tornam-se mais fáceis e utilizáveis.

Os ambientes inteligentes definem uma linha de pesquisa no qual os usuários são auxiliados em suas atividades diárias através da interação natural com os serviços computacionais. Os ambientes inteligentes possuem como característica trivial a adaptação, eles são capazes de se adaptarem às necessidades e às situações dos usuários. A interação realizada entre os usuários e os ambientes inteligentes pode acontecer de várias maneiras, seja através de interfaces adaptativas, através de sistemas que respondem as ações dos usuários, através de diversas aplicações.

Este trabalho apresenta conceitos, ferramentas e as tecnologias que são utilizadas pela computação ubíqua aplicada no desenvolvimento de ambientes inteligentes. Esta introdução abordou os conceitos iniciais da computação ubíqua juntamente com a delimitação do tema.

A seção 2 relata sobre o surgimento da computação ubíqua, bem como o seu idealizador Mark Weiser que contribuiu efetivamente para a evolução da tecnologia através de estudos e pesquisas dedicadas ao assunto. Além do mais, o leitor encontrará nesta seção alguns paradigmas que contemplam a computação ubíqua e as limitações e problemas que os desenvolvedores de aplicações ubíquas tentem driblar.

A seção 3 aborda os benefícios e vantagens de aplicar a computação ubíqua em ambientes inteligentes e também fala sobre a interação entre humano e computador.

A seção 4 fala sobre as redes *wireless* ou redes sem fio e a importante contribuição dessas redes para os ambientes inteligentes e aborda também alguns protocolos de descoberta de serviços que auxiliam na comunicação entre os usuários.

Para que toda essa tecnologia seja utilizada com qualidade é necessário que os equipamentos de *hardware* sejam capazes de suportar as aplicações, e na seção 5 são abordados alguns dos dispositivos inteligentes que são utilizados atualmente.

A seção 6 trata de assuntos relacionados às redes de sensores que tem um importante papel junto aos ambientes inteligentes fazendo com que os mesmos possuam consciência de contexto e executem as operações sem a intervenção direta do usuário.

A seção 7 evidencia a questão sobre softwares para ambientes inteligentes e os dispositivos nos quais estes softwares são utilizados.

A seção 8 trata sobre segurança, assunto bastante discutido no cenário mundial devido a grande exposição de dados e informações que a computação ubíqua apresenta.

Por fim tem-se, a seção 9 que comenta sobre alguns pontos apresentados nesse trabalho e faz recomendações para trabalhos futuros.

2. COMPUTAÇÃO UBÍQUA

Ubíqua, palavra derivada do Latim *ubiquu* e significa “o que está ao mesmo tempo e em toda parte”. Foi com esse pensamento que Mark Weiser citou pela primeira vez em seu artigo “*The Computer for the 21st Century*” (O Computador do Século 21) o termo Computação Ubíqua. Weiser nasceu em julho de 1952 e faleceu vítima de câncer gástrico em abril de 1999, ele estudou ciência da comunicação e informática na universidade de Michigan e acabou por se tornar cientista chefe do Centro de Pesquisa Xerox PARC onde desenvolveu importantes trabalhos relacionados a computação ubíqua. Por acreditar no crescimento das funcionalidades e no aumento da disponibilidade dos serviços computacionais Weiser percebeu que seria possível a utilização dos dispositivos computacionais em qualquer lugar, computação móvel. E que, será possível também utilizar os meios de computação distribuídos no ambiente de forma perceptível ou não, computação pervasiva. Então Weiser teorizou que a computação ubíqua é a integração entre mobilidade com os sistemas de presença distribuídos e que tem por objetivo fazer com que os usuários finais não se preocupem com a forma de acesso aos dispositivos de computação e nem com o modo de operação dos mesmos e sim fazer com que eles realizem as tarefas que desejarem permitindo o acesso às informações, aplicações e serviços em qualquer hora e lugar que precisarem, sem que percebam os meios computacionais agregados ao ambiente.

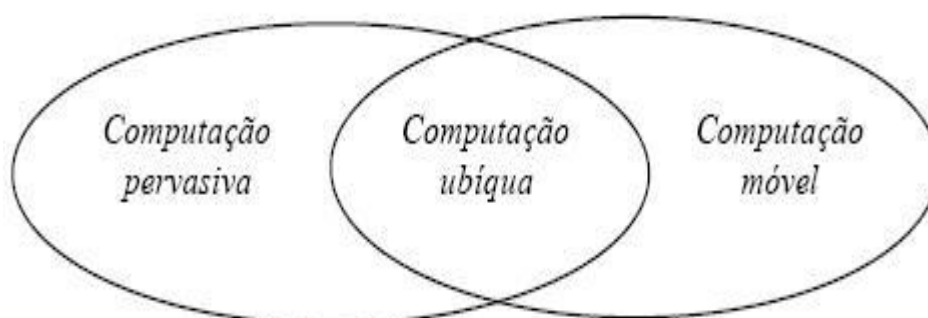


Figura 1: Integração entre computação pervasiva e computação móvel originando a computação ubíqua.

Fonte: <http://upload.wikimedia.org/wikibooks/pt/b/b0/FiguraComputacaoMovelPervasicaUbiqua.JPG>

Segundo Mark Weiser (1991) as tecnologias mais profundas e duradouras são aquelas que desaparecem. Elas dissipam-se nas coisas do dia a dia até tornarem-se indistinguíveis, ou seja, as tecnologias onipresentes são as que possuem capacidades adaptativas exemplares pois

elas fazem uso do meio para se instalarem e os usuários podem ter acesso a elas em qualquer momento e local.

Assim como as demais tecnologias existentes a computação ubíqua também tem o objetivo de facilitar e auxiliar as pessoas em seu cotidiano criando ambientes inteligentes que podem estar presentes em escritórios, residências, museus, automóveis, shoppings e até mesmo na hora de ir ao supermercado realizar suas compras. Por exemplo, quando um indivíduo entra com seu automóvel na garagem de sua residência e ao sair do veículo os faróis permanecem ligados por um período de tempo para que esse indivíduo possa adentrar em sua casa com o caminho iluminado ou até mesmo pegar as chaves da porta para abri-la, isso acontece porque um *software* pré-programado instalado no veículo possibilita condições para tal fato sem que seja necessário comandos do usuário, pois o *software* reconhece o contexto ao qual foi designado e executa as instruções programadas.

Confunde-se muito a computação móvel e a computação pervasiva com computação ubíqua. Pode-se definir a computação móvel como um conjunto de sistemas computacionais que se comunicam entre si e estão distribuídos em diversos dispositivos, essa comunicação é feita através de redes sem fio o que proporciona a mobilidade ao usuário. Já a computação pervasiva defende a idéia de que os dispositivos computacionais deverão estar distribuídos no ambiente de forma perceptível ou não, ou seja, o computador passa a se tornar uma ferramenta transparente para o usuário e não mais aquela máquina que tem de ser ligada, operada e desligada ao fim das tarefas, suas interfaces devem ser amigáveis para que usuários com menos conhecimento possa executar suas atividades da mesma forma que os demais. A figura a baixo exemplifica as dimensões da computação pervasiva, móvel e ubíqua.

	COMPUTAÇÃO PERVASIVA	COMPUTAÇÃO MÓVEL	COMPUTAÇÃO UBÍQUA
MOBILIDADE	Baixa	Alta	Alta
GRAU DE “EMBARCAMENTO”	Alto	Baixo	Alta

Figura 2: Dimensões da Computação Ubíqua.

Fonte: http://ufscar.academia.edu/documents/0039/4997/S%C3%ADntese_Definitiva_1.pdf

Nos tempos contemporâneos a inserção tecnológica na vida das pessoas torna-se cada vez mais comum porém para que isso aconteça os profissionais de *TI* (Tecnologia da informação) estão se aprimorando em ritmo acelerado o que favorece bastante a evolução da computação ubíqua. Interfaces mais amigáveis e fáceis de serem manipuladas são muito importantes porque assim os usuários não terão grandes dificuldades para manipular os dispositivos tendo vista que o foco principal não é a operação dos meios computacionais e sim a realização das tarefas que os usuários necessitam realizar.

A computação ubíqua objetiva evoluir a comunicação entre homem e máquina e essa integração acontece também através de interfaces naturais, mais o que parece ser solução é apenas o início do problema isso porque analistas, projetistas e programadores ainda encontram dificuldades em desenvolver *software* que atendam plenamente as necessidades dos usuários. A maioria dos projetos existentes potencializa seus esforços no reconhecimento da escrita e da fala mais isso ainda não é suficiente. Será preciso desenvolver aplicações que interajam com os utilizadores reconhecendo expressões e gestos e usar todos esses dados dentro de um contexto de operações, podendo até mesmo captar alterações no meio ambiente e armazená-las em memória.

Weiser e Seely Brown (1995) usaram o termo “tecnologia calma” para descrever uma importante visão da computação ubíqua, o fato de que a computação não deve prender a atenção do usuário e sim executar atividades silenciosas para que os usuários possam perceber a realização das tarefas de uma maneira mais ubíqua. Por isso é tão importante o desenvolvimento de aplicações embarcadas no ambiente e pré-programadas para interagir diretamente ou indiretamente com o homem.

2.1 PARADIGMAS DA COMPUTAÇÃO UBÍQUA

A computação ubíqua postula alguns paradigmas, ou seja, princípios que devem ser seguidos com intenção de formular a base da *UbiComp* (computação ubíqua), são eles:

Diversidade: a computação ubíqua adota a tendência de utilizar os mais variados tipos de dispositivos, é claro, vai depender do que se espera que determinado dispositivo faça, existe uma grande diferença entre um *PC desktop* que tem um propósito de executar operações em geral e um dispositivo computacional ubíquo que tende a realizar tarefas específicas que variam de acordo com o grupo de usuários. Cada aplicação pode ter uma relação com dispositivos diferentes, mesmo sabendo que alguns dispositivos podem oferecer

funcionalidades que se sobrepõem é também preciso saber avaliar quando um pode ser mais apropriado para uma função que o outro, por exemplo, um palmtop é um bom dispositivo para fazer anotações rápidas, mas não é um bom dispositivo para navegar na internet. Essa diversidade dos equipamentos faz surgir o desafio de como gerenciar a capacidade dos dispositivos, já que, cada um fornece plataformas com suas próprias características, o que faz com que aplicações homogêneas entre eles tornem-se mais difíceis.

Descentralização: quando os primeiros computadores surgiram com todo o seu tamanho e robustez, eles proviam informações para outros computadores, os chamados “terminais burros” e toda a sua capacidade de processamento, armazenamento e gerenciamento das informações eram executados por esses computadores poderosos, conhecidos como *mainframes* que em uma tradução livre significa “estrutura principal”.

Com o passar do tempo surgiram novas arquiteturas, entre elas a arquitetura cliente-servidor e junto com essa arquitetura surgiram também os computadores pessoais (PCs). A partir daí foi que a descentralização se fez presente e as responsabilidades passaram a ser distribuídas entre os vários dispositivos computacionais cada qual com suas responsabilidades. Na computação ubíqua é assim, a responsabilidade é distribuída entre os diversos dispositivos que assumem e executam determinadas tarefas e funções, esses dispositivos cooperam entre si formando ambientes inteligentes e redes que se relacionam transportando e transmitindo informações de um ponto a outro. A descentralização também trouxe uma nova responsabilidade. Responsabilidade essa que consiste na sincronização e atualização constantes das informações, a base de dados presentes nos dispositivos que possuem capacidades e recursos limitados devem manter-se consistentes.

Simplicidade: para que uma aplicação ubíqua obtenha a simplicidade desejada é preciso fazer com que o computador não seja o foco das atenções, garantir que os usuários realizem suas tarefas sem se preocuparem com o uso da computação e prover a disponibilidade e conveniência dos serviços oferecidos é muito importante. Os dispositivos devem desempenhar bem as funcionalidades para as quais foram designados e o gerenciamento e acesso às informações devem ser desempenhadas sem gastar tempo significativo para aprender como usar a tecnologia.

Conectividade: não dá para falar em computação ubíqua sem citar a conexão entre os diversos equipamentos, eles estão de certa forma ligados uns aos outros, seja através de

infravermelhos, através de cabos ou por meio de redes sem fio e etc. A conectividade ocupa papel de grande importância na computação ubíqua pois através dela é possível fazer com que diversos dispositivos interajam entre si, proporcionando a mobilidade dentro de um determinado ambiente. Através da conexão é possível fazer a troca de documentos via rede e acessá-los de qualquer lugar, é possível enviar pacotes de dados e até mesmo de voz através da *web*, fazer com que telefones celulares se comuniquem com alguns acessórios e eletrodomésticos existentes nas residências e outros. A conectividade entre os equipamentos e dispositivos causa alguns problemas, por exemplo, existem dispositivos que operam em plataformas de softwares semelhantes, mas existem também aqueles que operam em plataformas diferentes e causam alguns conflitos na execução das aplicações. É por isso que os padrões de comunicação existentes são tão importantes para a integração e a conectividade dos dispositivos, pois estabelecem normas e regras para que os protocolos de comunicação possam ser definidos, bem como as conexões físicas necessárias.

2.2 LIMITAÇÕES DA COMPUTAÇÃO UBÍQUA ATUAL

Para atingirmos plenamente a definição de computação ubíqua proposta por Weiser ainda será preciso vencer alguns obstáculos. A computação ubíqua precisa ainda de muitas contribuições para evoluir seus projetos e alcançar seus objetivos, mas acredita-se que os recursos de *hardwares* atualmente disponíveis serão suficientes para suprir as necessidades de desenvolvimento das aplicações ubíquas, entretanto ainda existem algumas limitações no uso desses dispositivos.

As aplicações pelas quais os usuários realizam suas tarefas devem ser dispostas no ambiente de forma que os dispositivos responsáveis por gerenciá-las não se apresentem como fator relevante ao ambiente, eles deverão estar transparentes para os usuários e a manipulação de qualquer desses dispositivos não deve tomar tempo ou atenção do utilizador, pelo contrário, os serviços devem ser fornecidos da forma mais simples o possível seguindo os paradigmas da computação ubíqua e satisfazendo quem os utiliza.

A disponibilidade dos serviços em um ambiente será maior e o número de dispositivos também, as informações estarão dispostas em toda parte sendo possível acessá-las em qualquer hora e lugar e os dados serão sincronizados e atualizados a todo instante, mas para isso ocorrer será preciso deixar para trás alguns limites e desafios ainda em aberto.

Escalabilidade: um dos fatores que devem ser melhorados na computação ubíqua é a escalabilidade, que tem por finalidade garantir que um sistema continue trabalhando de

maneira satisfatória mesmo que haja aumento significativo em suas funcionalidades e no número de usuários que o utilizam. De um modo geral podemos dizer que um sistema escalável é aquele que possui a capacidade de gerenciar o aumento das aplicações, dos dispositivos, dos usuários e os demais serviços que o complementa.

Heterogeneidade: em computação ubíqua a diversidade de dispositivos, protocolos de rede, aplicações, software e sistemas operacionais existentes é muito ampla, e sabemos que para cada ambiente existe uma melhor opção para alcançar os objetivos, então, baseado nisso, podemos dizer que a heterogeneidade é uma maneira de lidar com as diferenças e fazer com que sistemas e dispositivos ubíquos trabalhem em conjunto para tornar os ambientes para os quais foram projetados mais inteligentes.

Integração: fazer com que haja interação entre os componentes e dispositivos de um determinado ambiente consiste em um grande desafio para a computação ubíqua porque não se pode pensar em compartilhar dados e informações sem se preocupar com os fatores confiabilidade e privacidade. Por consequência da mobilidade existente na computação ubíqua os componentes estão em constante variação, dificultando padrões de comunicação.

Segurança e Privacidade: garantir a segurança e a privacidade em ambientes ubíquos é uma tarefa muito importante, afinal, informações pessoais, dados sigilosos trafegam a todo o momento em várias redes sem fios e diversos sensores espalhados pelo ambiente, então nada mais justo que a integridade, a autenticidade desses dados sejam protegidas, seja através de mecanismos de criptografia, regras de uso ou protocolos. O importante mesmo é a garantia de integridade das informações.

Interface com o Usuário: a interface com o usuário ou interface natural ainda é um problema evidente que precisa de maior atenção por parte dos especialistas no assunto. Deve ser pensado em formas de como melhorar a interação com o mundo real, qual a interface adequada para os diversos tipos de dispositivos e ainda deve-se atentar para o desenvolvimento dinâmico dos sistemas devido ao fator mobilidade. Além disso, os usuários devem poder focar nas tarefas sem distrações do sistema.

Mobilidade: presentes em várias aplicações a mobilidade já é algo que contempla a computação ubíqua, mas não deve parar por aí, ter acesso aos aplicativos e aos dados

independentes da localização física do usuário é muito bom mais, imagine poder fazer com que os aplicativos e os dados sigam os usuários, fazer com que eles se movam de um dispositivo para outro sem perder o acesso aos dados, esse é mais um desafio a ser vencido pela computação ubíqua.

Consciência de Contexto: conhecer o meio ao qual está agregado reconhecendo as mudanças e respondendo-as de modo que as necessidades dos usuários sejam satisfeitas é um exemplo de consciência de contexto. Na maior parte os dados são obtidos através de sensores espalhados pelo meio que captam alterações, movimentos, temperatura, luminosidade, etc, o sistema processa os dados e através de comandos pré-programados geram reações que satisfaçam os usuários, por exemplo, ligar ou ascender uma lâmpada, ajustar a temperatura do ar condicionado ou aquecedor e outros.

Invisibilidade: esta talvez seja a questão mais ligada à computação ubíqua. Saha e Mukherjee afirmam que os humanos podem intervir para ajustar os ambientes inteligentes quando estes falham ao tentar atingir a expectativa do usuário automaticamente. O fato de tornar os dispositivos invisíveis tem haver com integrá-los ao ambiente com o propósito de que os usuários não percebam que estão fazendo uso da computação, talvez este seja o fator que envolva a maior quantidade de recursos, desde interfaces naturais até consciência de contexto, tudo isso com o propósito de fazer prevalecer a ubiqüidade das aplicações.

2.3 MARK WEISER: O VISIONÁRIO

Mark Weiser foi o criador da computação ubíqua ou “*ubicomp*”. No Laboratório de Ciência da Computação da Xerox Parc foram realizados vários estudos para se poder compreender como as pessoas trabalhavam e que ferramentas usavam. Weiser percebeu que a melhor utilização de uma ferramenta é quando o usuário não percebe que está utilizando-a, isto é, o usuário não percebe seu uso e dessa forma, ele pode centrar mais atenção no trabalho que está realizando.

Dr. Weiser começou suas pesquisas em 1988 e percebeu que a tecnologia da época não era suficiente para suportar a sua visão. Então ele se focou primeiro, nas diferentes características da computação ubíqua, como dispositivos que eram portáteis e cuja localização era sempre conhecida ou em dispositivos que auxiliariam as pessoas a trabalharem em conjunto. Os primeiros protótipos foram os “*tabs*”, “*pads*” e “*boards*”. “*Tabs*” são similares

a pequenos blocos de anotações eletrônicas similares aos “*Post-It*” e, poderiam haver centenas deles em um escritório, cada um correspondendo a uma pequena superfície com informações. Já os “*pads*” são como blocos amarelos, maiores, ou computadores usados com o objetivo de serem utilizados momentaneamente e depois, poderem ser deixados para outras pessoas quaisquer. Os “*boards*” são enormes quadros negros eletrônicos, que visam permitir o trabalho colaborativo e que estão em uso, hoje em dia.

3. AMBIENTES INTELIGENTES

Quando Mark Weiser (1991) citou pela primeira vez o termo “computação ubíqua” ele previu que o rumo a ser tomado pela computação implicaria fortemente sobre o conceito de computação conhecido atualmente, o computador não seria apenas uma máquina estática projetada para ficar em cima de uma mesa e ser operada por usuários com conhecimentos especializados em informática, não seria só mais um dispositivo para operações com editores de textos, planilhas eletrônicas e manipulação de imagens. Os recursos computacionais passariam a ser onipresentes, as pessoas fariam uso dos meios de computação distribuídos no ambiente de forma transparente, ou seja, os dispositivos computacionais seriam usados de maneira discreta a ponto dos usuários não perceberem e o interessante é que o foco principal dos usuários não sofreria alterações.

Apostando nas funcionalidades e objetivos da computação ubíqua surgiu-se a idéia de implementar aplicações ubíquas aos ambientes tornando-os inteligentes. No desenvolvimento de ambientes inteligentes o objetivo é produzir formas de evitar que os usuários se desloquem para ir de encontro aos computadores e dispositivos informatizados e fazer com que os elementos que integram o ambiente possam ser utilizados e combinados entre si.

Baseando-se em um amplo sentido os ambientes inteligentes são um conjunto de tecnologias que trabalham de maneira integrada, permitindo o entendimento de algumas situações e ativando instruções ou comandos pré-programados sem a necessidade de instruções diretas dos usuários. Um exemplo de ambientes inteligentes é quando dispositivos computacionais podem captar a presença humana em uma residência ascendendo ou apagando as luzes na presença ou ausência dos moradores, respectivamente.

Nesses ambientes, o conjunto de elementos computacionais que participam de um sistema distribuído varia dinamicamente. Ou seja, o usuário se interconecta espontaneamente com diferentes dispositivos à medida que outros usuários se locomovem ao longo do tempo

(Chakraborty, 2006). A infra-estrutura subjacente da computação ubíqua é mais sofisticada e deve tratar questões como a mobilidade do usuário, desconexão, introdução e remoção dinâmica de dispositivos, conexões de redes diversas, bem como a necessidade de mesclar o ambiente físico com a infra-estrutura computacional (Murphy, 2001).

A figura 3 demonstra o exemplo de um dos vários tipos de ambientes inteligentes que podem ser elaborados. A indústria de automóvel ao perceber a oportunidade de novos negócios começou a produzir veículos com funções inteligentes que tiveram boa aceitação no mercado e atualmente é sucesso garantido. No caso em questão representado na figura abaixo está a arquitetura de um automóvel inteligente que tem como funções características de um ambiente inteligente o diagnóstico remoto, assistência a emergências nas ruas em caso de acidentes ou até mesmo se o veículo quebrar, serviços de navegação, entretenimento, telefonia móvel, informações personalizadas e outros.



Figura 3: Arquitetura de automóveis inteligentes

Fonte: http://www.pad.lsi.usp.br/humanlab/trabalhos/workshop_amb_intel.pdf

O suporte à mobilidade se torna uma característica importante no projeto da arquitetura de sistemas ubíquos. Usuários devem ser capazes de tirar vantagem completamente das capacidades e recursos de um dado ambiente, mesmo enquanto outros usuários e dispositivos entram e saem deste ambiente, e os recursos variam (Sousa and Garlan, 2002).

Os ambientes inteligentes definem uma linha de pesquisa na quais os usuários são auxiliados em suas atividades diárias através da interação natural com os serviços

computacionais. Os ambientes inteligentes possuem como característica trivial a adaptação, eles são capazes de se adaptarem as necessidades e as situações dos usuários. A interação realizada entre os usuários e os ambientes inteligentes pode acontecer de várias maneiras, seja através de interfaces adaptativas, através de sistemas que respondem às ações dos usuários, através de aplicações de multimídia e outras.

Os Ambientes Inteligentes podem ser implementados em cenários automatizados como, por exemplo, salas de reunião e salas de aula (NAZARI, 2006), sendo que as redes de dispositivos inteligentes contribuem para a integração entre pessoas, fornecendo informação, comunicação, serviços e entretenimento em qualquer lugar e em qualquer momento (AARTS, 2004). Especialmente a utilização de redes sem fio (“*wireless*”) nesses ambientes também permite a mobilidade das informações sobre atividades e preferências dos usuários (INIGRAPHICSNET, 2003).

A expansão dos ambientes inteligentes vem se tornando cada vez mais expressiva, atualmente várias empresas operam nesse ramo e desenvolvem projetos de inovadores para tornar esses ambientes cada vez mais seguros, confortáveis e econômicos. A qualidade dos equipamentos existentes aliada ao conhecimento técnico das equipes tornam a execução das atividades do cotidiano mais práticas.

Algumas empresas procuram se especializar em determinados seguimentos, por exemplo, linhas náuticas, automação residencial, corporativa, etc. Os ambientes náuticos, por exemplo, são equipados com leitores digitais, banheiras e chuveiros inteligentes e controle de bordo via celular, propiciando assim ao cliente um passeio mais tranquilo com seus amigos e familiares pois tudo que é preciso fica disposição.

Toda a estrutura e cabeamento é feita de forma discreta e completamente segura, integrando todas as funções e mantendo a total eficiência dos equipamentos enquanto garante que a paisagem e o passeio será a única coisa que irá chamar a atenção do consumidor ou usuário. Geralmente os ambientes inteligentes náuticos privam por determinados serviços, esses são os mais oferecidos nesse tipo de ambiente:

- Computador de bordo, o sistema foi especialmente desenvolvido para embarcações com proteção anti-maresia e possibilidade de *display* à prova d'água. Permite ouvir músicas em *MP3* armazenadas no disco, reproduzindo em todos os ambientes da embarcação, receber orientações da previsão do tempo e das marés em tempo real, navegar na internet ou ler ou editar documentos. Além de possuir *GPS* com mapas marítimos, o sistema possibilita integração com sistema de sonar.

- Controle de acesso, por meio de fechaduras com leitura biométrica, o usuário pode controlar o acesso a determinadas cabines da embarcação com a possibilidade de que cada um dos dedos sejam pré-programados para uma determinada função no barco. O sistema também permite ao proprietário cadastro de novos usuários com acessos e horários pré-estabelecidos, abertura da porta por senhas numéricas, que podem ser programáveis de qualquer lugar do mundo através do celular. Além disso, os últimos 64 acessos ao barco ficam armazenados na memória do sistema, com registro de dados do usuário, data e horário em que esteve ou passou pelo local.
- Controle de bordo, sistema de automação que utiliza pulsadores inteligentes ou um painel gráfico *touchscreen* para executar funções como ligar o ar condicionado em uma determinada temperatura, acionar iluminação, abrir ou fechar persianas, bem como controlar outros sistemas da embarcação. Além de poder realizar macro-comandos, que, com apenas um toque, baixa as persianas, apaga a iluminação e aciona o ar condicionado.
- Chuveiro inteligente, com ele é possível programar e memorizar a temperatura da água, fluxo, além de avaliar o nível de água doce do tanque, controlando dessa maneira o consumo, o que permite ao usuário saber se o banho está sendo muito prolongado para a quantidade de água disponível.
- Sistema de pilotagem via celular, agora, a pilotagem do barco pode ser feita pelo celular, com o painel de instrumentos reproduzido na tela do aparelho. O proprietário pode ainda monitorar à distância, através do celular, informações como *RPM*, pressão do óleo, consumo instantâneo, temperatura do turbo e tensão da bateria. Temperatura da água do mar e profundidade também podem ser obtidas.
- Wi-Fi, a internet é provida através de operadora celular *GSM* ou *CDMA*. Um painel de *LCD* instalado no interior do barco informa dados sobre a conexão como: qualidade do sinal, tipo de conexão (2G, 3G), *roaming*, e velocidade do *uplink* e *downlink*. Assim, basta levar o *notebook* a bordo para navegar utilizando a Internet sem fio.

3.1 AMBIENTES CIENTES DE CONTEXTO

Com o aumento crescente das funcionalidades dos dispositivos computacionais novas tecnologias surgem a todo instante, aparelhos celulares e *smartphones*, *notebooks* menores e cada vez mais poderosos permitem que os usuários estejam sempre conectados aos serviços e aplicações disponíveis. Com isso, os ambientes inteligentes aprimoram-se ainda mais.

Os ambientes inteligentes deixam de oferecer apenas funções pré-programadas e passam a ser cientes de contexto. A consciência de contexto consiste na capacidade do sistema computacional perceber características de uma situação ou do ambiente e se adaptar de acordo com o momento, ou seja, uma aplicação ciente de contexto é capaz de obter as informações do meio a qual está condicionado e utilizá-las com a finalidade prover serviços apropriados a um determinado usuário. Por exemplo, um telefone celular pode automaticamente entrar em modo silencioso quando o usuário entra em uma sala de reunião, mas para que isso aconteça o sistema deve conseguir identificar a localização do aparelho e a hora de uma reunião agendada.

Dey (2001) define contexto “como qualquer informação relevante que possa ser utilizada para caracterizar entidades de uma interação usuário computador”, por exemplo, identidade e localização.

Indo além da interação física do usuário, sensores espalhados pelo meio poderão detectar o que acontece ao redor, o que as demais pessoas em uma reunião estão fazendo ou dizendo e até mesmo outros dados como luminosidade, temperatura, expressões faciais, direção do olhar entre outros.

Estes dados poderão ser disponibilizados para as aplicações e através de modelos computacionais as interfaces poderão “entender” o que está acontecendo ao redor do usuário e adaptar-se ou tomar decisões, alterando e facilitando a comunicação com o humano.

A seguir um exemplo de um cenário ciente de contexto:

A montadora japonesa *Nissan Motor Company* apostando em um alto índice de venda de seus veículos e na redução de acidentes de trânsito envolvendo bebidas alcoólicas lançou um carro conceito que faz uso dos ambientes inteligentes sensíveis ao contexto que consiste em abrigar um conjunto de dispositivos encarregados de monitorar a lucidez do motorista. Caso o mesmo não passe nos testes, o carro é imobilizado junto com o disparo de um alerta.

Através de sensores espalhados por várias partes do veículo com a finalidade de verificar se o condutor está dirigindo embriagado ao entrar no carro o sistema faz um breve exame de transpiração na palma da mão do motorista. Caso identificado uma quantidade considerável de álcool no seu organismo o condutor não consegue nem mesmo mudar a marcha.

Além do mais o carro possui sensores de odor no banco do motorista e nos assentos dos passageiros e ainda possui também uma câmera que é apontada diretamente para o rosto do motorista, e com uma leitura de pontos-chaves do rosto (principalmente os olhos), é possível identificar a consciência do motorista. Desta forma, tanto para níveis alcoólicos elevados, como para motoristas com sono, ao detectar uma possível sonolência, o sistema emite o alerta e de quebra dá-lhe um apertão com o cinto de segurança.

A figura abaixo representa como será feita a leitura facial no carro conceito lançado pela Nissan:



Figura 4: Leitor facial anti-bêbado (Veículo Nissan)

Fonte: <http://www.techzine.com.br/arquivo/nissan-lanca-conceito-de-carro-anti-bebados/>

O exemplo demonstra com clareza os principais fatores que elaboram a consciência de contexto dentro de um ambiente, os fatores são a capacidade de obter o contexto e a capacidade de utilizar as informações contextuais a fim de atender as necessidades dos usuários. O contexto é abstraído através de sensores que captam as mais variadas informações presentes no meio e em seguida essas informações são processadas por um *software* específico que interpreta o que deve ser feito, e então o sistema repassa os comandos necessários para os dispositivos computacionais espalhados pelo ambiente para que as diversas ações possam ser realizadas.

3.2 INTERAÇÃO HUMANO-COMPUTADOR (IHC)

A interação entre humano e computador acontece através de interfaces que possibilitam e provêm os serviços aos usuários e tornam de alguma maneira esses serviços mais acessíveis e fáceis de serem utilizados. As interfaces são formadas pelo *software* e

hardware e são utilizadas na operação de diversos tipos de máquinas, sejam elas de grande ou pequeno porte, por exemplo, os controles de navegação de uma aeronave ou até mesmo a simples manipulação de um aparelho celular.

A *IHC* esta atingindo proporções maiores a todo instante, a principio *softwares* e *hardwares* eram materiais usados apenas por pessoas que detinham um conhecimento técnico avançado. Ligeiramente complexos os computadores tiveram de se adaptar a fim de alcançar aceitação unânime. A base dessa aceitação está presente nas interfaces que tornaram-se cada vez mais amigáveis, proporcionando um fácil entendimento aos usuários que as utilizam dispensando assim o conhecimento especializado para grande parte das aplicações oferecidas.

Ainda que tenham aprimorado algumas interfaces muitos usuários sentem dificuldades em utilizar os dispositivos computacionais mais espera-se que isto seja um fator temporário e que homem e computador se entendam perfeitamente.

A interação humano-computador (*IHC*) é uma área multidisciplinar que envolve as áreas da ciência da computação, psicologia, fatores humanos, etc. Os seres humanos percebem as coisas através de seus sentidos, isto é, visual, auditivo e tato. Estes sentidos habilitam o usuário de um sistema interativo perceber a informação, armazená-la (em sua memória) e processar a informação usando o raciocínio dedutivo ou indutivo. A maioria das *IHC* ocorre através do sentido da visão, como por exemplo: relatórios, gráficos, etc. Neste caso, o olho e o cérebro trabalham juntos a fim de receber e interpretar a informação visual baseada no tamanho, forma, cores, orientação e movimento. Muitos elementos discretos de informação são apresentados simultaneamente para o homem absorver. Assim, uma especificação apropriada de comunicação visual é o elemento chave de uma interface amigável.

É comum encontrarmos interfaces que são difíceis de usar, confusas, e até mesmo frustrantes em alguns casos. Apesar de os projetistas gastarem tempo para desenvolver essas interfaces e que seja improvável que eles façam isto propositadamente, os problemas com interfaces acontecem. Quando consideramos um sistema interativo, o termo fatores humanos assume vários significados. Dentro do nível fundamental, deveríamos entender a percepção visual, a psicologia da leitura, a memória humana e os raciocínio dedutivo e indutivo. No outro nível, deveríamos entender o usuário e seu comportamento. Por fim, precisamos entender as tarefas que o *software* executam para o usuário e as tarefas que são exigidas do usuário durante a interação com o sistema.

Quando a informação é extraída da interface, ela deve ser armazenada para ser recuperada e utilizada posteriormente. Além disso, o usuário precisa lembrar de comandos e

seqüências operacionais de uso. Tais informações são armazenadas na memória humana composta de duas partes: a memória de curta duração que possui capacidade de armazenamento e tempo de recordação limitado e a memória de longa duração que possui capacidade de armazenamento e tempo de recordações maiores e onde se tem o conhecimento do ser humano.

Embora seja muito comum encontrar interfaces de comunicação gráficas nos dispositivos computacionais ainda se é muito utilizado as informações textuais, em que, os usuários lêem os comandos e os seguem a fim de encontrar os serviços pretendidos. Tal importância da *IHC* é reconhecida por todos pois, imagine um ambiente inteligente que faz uso de sistemas ubíquos distribuídos por todo o meio, nele é possível que o usuário controle o acionamento da cerca elétrica da sua residência via celular, é possível ter acesso a internet através de *displays* espalhados por toda a casa e os *displays* ainda possuem tecnologia *touch screen*. Se o ambiente descrito não for elaborado para interagir com o usuário com certeza o mesmo enfrentaria problemas para usar os serviços disponíveis, entretanto pesquisas e trabalhos estão desenvolvendo novas formas de melhorar a interação entre o humano e o computador.

Weiser e Seely Brow usaram o termo “tecnologia calma” para descrever um importante aspecto de sua visão de computação ubíqua: o fato de que a computação não deve continuar a monopolizar a atenção do usuário. Em muitos casos, a atividade de computação deve silenciosamente ser executada em segundo plano (sem que o usuário perceba) e tornar seu resultado acessível à percepção daqueles que a utilizam.

Dispositivos ubíquos são ferramentas extremamente especializadas que são utilizadas para um propósito específico e não geral. De um ponto de vista de usabilidade, eles desempenham bem as tarefas para os quais foram projetados. Nesses dispositivos, disponibilidade, conveniência e facilidade de uso são pré-requisitos fundamentais. O acesso e o gerenciamento da informação devem ser realizados sem se gastar tempo significativo em aprender como usar a tecnologia. Computadores ubíquos são intuitivos de usar e não deveriam sequer requerer a leitura de um manual. Processos que hoje requerem procedimentos de instalação e levam vários comandos em um *PC* irão apenas necessitar de um simples apertado de uma tecla ou mesmo aceitar requisições de voz ou possuir reconhecimento de escrita.

Apesar de tudo não se deve confundir simplicidade com primitivo, até porque, os sistemas em geral são mesmo complexos, mas a complexidade das tecnologias deve estar escondida por trás de uma interface amigável com o usuário, fazer com que os serviços e as

tarefas possam ser realizadas de maneira simples e prática e utilizar o conjunto de dispositivos sem nenhum medo ou receio do que possa acontecer se algo for executado de forma incorreta.

3.3 INTERFACES

As interfaces com os usuários com certeza é uma parte significativa da computação ubíqua. Devido às várias restrições existentes na interação usuário-computador tem-se a necessidade de elaborar outras formas de interação, ou seja, através de interfaces. Mas para desenvolver um projeto de interface deve ser levado em consideração fatores como: o *hardware* utilizado, os limites das telas, as diferentes razões de aspecto, os periféricos e as técnicas de entradas de dados, a conectividade, o desempenho e as ferramentas disponíveis para o desenvolvimento.

Devido à mobilidade das aplicações as interfaces elaboradas para ambientes ubíquos devem ser dinâmicas e os usuários devem poder focar em suas tarefas sem que haja distrações causadas pela operação dos sistemas.

Por meio do suporte a formas comuns de expressão humana, as interfaces naturais facilitam a capacidade de comunicação entre usuários e computadores utilizando ações explícitas ou implícitas durante a comunicação.

O objetivo das pesquisas nessa área é aproximar a interação usuário-computador da interação natural que ocorre entre pessoas. Desse modo, a interação usuário-computador seria não-intrusiva, ideal para computação ubíqua. Com isso, projetos voltados para interfaces mais amigáveis investigam técnicas de reconhecimento de escrita e de gestos, interação com canetas, técnicas de voz e percepção computacional, interação com sensores e manipulação de artefatos eletrônicos (Abowd, 2002).

O objetivo dessas interfaces naturais é suportar formas comuns de expressão humana. Esforços anteriores se focaram em interfaces de reconhecimento de voz e escrita com uma caneta eletrônica, mas estas interfaces ainda não lidam robustamente com os erros que ocorrem naturalmente com estes sistemas. Além disso, estas interfaces são muito difíceis de serem implementadas. A computação ubíqua inspira o desenvolvimento de aplicações que não utilizam o *desktop*. Implícito a isto está a consideração que a interação física entre humano e computadores serão bem diferentes do *desktop* atual com teclado, mouse, monitor, e será mais parecida com a maneira que os humanos interagem com o mundo físico. Interfaces que suportem formas de computação humanas mais naturais estão começando a substituir os dispositivos mais tradicionais. Estas interfaces se sobressaem por causa da sua facilidade de

aprendizado e de uso. Além disso, elas podem ser usadas por pessoas com deficiência física, para os quais a manipulação do mouse e do teclado representa na maioria das vezes um problema.

O aumento expressivo de dispositivos pelo ambiente representa também o crescimento e desenvolvimento de interfaces naturais, fazendo surgir assim várias formas de interfaces.

Interfaces Hands-Free (sem as mãos)

São tecnologias já desenvolvidas ou em fase de desenvolvimento como reconhecimento de voz, *liveboards*, *Pads* e *Tabs* quando instaladas em uma sala e utilizadas em conjunto permitirão que uma pessoa se mova pelo ambiente enquanto permanece em contínua interação com o computador.

Interfaces Sensíveis a Objetos

Esta talvez seja a tecnologia mais próxima da realidade, sendo utilizadas em alguns supermercados da Europa e EUA e em algumas linhas de produção através de sistemas *RFID* (*Radio-Frequency IDentification*). Esta tecnologia permite que um objeto dotado de um pequeno sensor ou mesmo com seu formato cadastrado previamente seja passível de reconhecimento pelo computador, levando ao usuário possuidor de um computador portátil acesso a informações relacionadas. Ao aproximar um computador portátil de um objeto, o usuário poderá ser levado ao site do fabricante, obter informações sobre validade, ano de fabricação, preços, etc. Este conceito também é conhecido como *Phicon*, ou *Physical Icon*, ou seja, associar um objeto real a um significado gerando informações.

Interface de Programação da Aplicação – APIs (Application Programming Interface)

As APIs são um conjunto de rotinas padrões estabelecidos por um software para a utilização das suas funcionalidades por programas aplicativos que não querem se envolver com os detalhes da implementação do software, mas apenas usar seus serviços. As APIs são compostas por uma série de funções acessíveis somente por programação, e que permitem utilizar características do software menos evidentes ao utilizador tradicional, por exemplo, um sistema operacional possui uma grande quantidade de funções na API, que permitem ao programador criar janelas, acessar arquivos, criptografar dados, etc, mas a API dos sistemas operacionais costumam ser diferenciadas de tarefas mais essenciais, como manipulação de blocos de memória e acesso a dispositivos.

Atualmente o uso das *APIs* tem-se generalizado nos *plugins*, que são acessórios que complementam as funcionalidades de um programa, existem vários tipos de *APIs*, por exemplo, *APIs JavaPhone* que é utilizada para serviços de telefonia, *JavaTV* que é utilizada em serviços de *TV*, *PC/SC* para *SmartCards* que é uma interface de programação padronizada que permite acessar cartões inteligentes de forma independente da aplicação, existe também o *Framework OpenCard (OCF)* que assim como o *PC/SC*, o *OCF* é uma *API Java* para desenvolver aplicações de *smartcards* no lado do terminal (leitora).

Apesar de todos os avanços na elaboração de interfaces naturais o desenvolvimento das mesmas para a computação ubíqua ainda é um problema sem solução, ainda é preciso encontrar maneiras que façam o computador se encaixar no mundo real e nas necessidades humanas, tornando assim as interfaces mais amigáveis e naturais o possível.

4. AS REDES WIRELESS (REDES SEM FIO)

As tecnologias de rede evoluem a cada dia e se integram cada vez mais a computação ubíqua. As redes *wireless* provêm informação em qualquer hora e lugar e atualmente está presente em uma gama muito extensa de dispositivos computacionais, é o caso dos *notebooks*, *smartphones*, celulares, *PDA*s, impressoras e muitos outros.

As redes *wireless* possuem papel de grande importância no meio ubíquo, elas carregam em sua essência a filosofia da computação ubíqua que é oferecer os serviços computacionais de forma transparente a fim de que os usuários realizem suas tarefas sem a preocupação com o momento e o local onde estão e que os computadores ou dispositivos sejam usados de forma natural tornando-se quase ou totalmente imperceptíveis.

Uma rede *wireless network* ou rede sem fio é, como o próprio nome sugere, uma rede pela qual dois terminais distintos possam fazer comunicação sem que haja ligação telegráfica, as redes *wireless* permitem o deslocamento em um espaço geográfico sem a necessidade de interrupção da conexão.

As redes *wireless* baseiam-se numa ligação que utiliza ondas radioelétricas (rádio e infravermelhas) em vez dos cabos habituais. Existem várias tecnologias que se distinguem por um lado pela frequência de emissão utilizada bem como o débito e o alcance das transmissões.

A figura ilustra o exemplo de uma rede *WLAN* convencional:

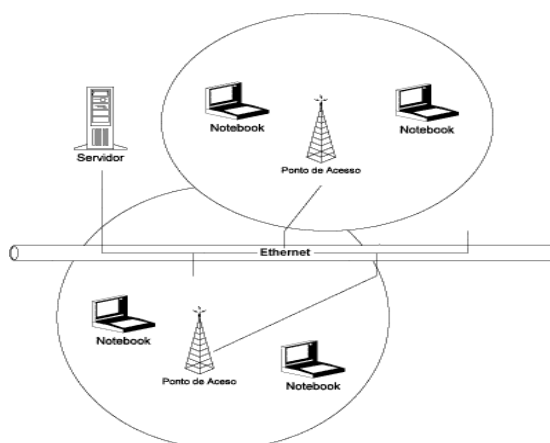


Figura 5: Rede Wireless LAN típica

Fonte: <http://www.rnp.br/newsgen/9805/wireless.html#inicio>

Com várias denominações e seguimentos as redes *wireless* estão mesmo presentes no cotidiano das pessoas e isso é muito importante para a computação ubíqua porque com a conexão dos dispositivos a diversas redes disponíveis torna-se mais fácil expandir os serviços e aplicações ubíquas pelo ambiente e propicia também o conhecimento sobre o que vem a ser a computação ubíqua e os ambientes inteligentes que podem ser criados com base nas premissas da *UbiComp* (computação ubíqua).

4.1 PROTOCOLOS DE DESCOBERTA DE SERVIÇO

Com a crescente evolução dos ambientes de redes sem fio viu-se a necessidade de protocolos de descoberta de serviços (*SDPs*) mais avançados. Um protocolo de descoberta de serviços provê mecanismos para descobrir os serviços disponíveis em uma rede, fornecendo as informações necessárias para procurar serviços específicos, escolher o serviço adequado (aquele com as características desejadas), e utilizar o serviço.

Infra-estruturas para computação ubíqua devem tratar questões como a mobilidade de usuários e dispositivos e a conseqüente desconexão, conexão a redes diversas, introdução e remoção dinâmica de dispositivos e serviços. Neste cenário, componentes devem interagir espontaneamente e para isso necessitam de protocolos de descoberta que localizem dinamicamente instâncias de serviços que combinem com as necessidades de cada componente. A seguir veremos alguns dos principais protocolos de descoberta de serviços existentes.

4.1.1 SERVICE LOCATION PROTOCOL (SLP)

O *SLP* foi desenvolvido por um grupo de trabalho da *IETF* (*Internet Engineering Task Force*), seu objetivo é ser um padrão independente de fabricantes. Foi projetado para redes *TCP/IP* e é escalável para redes de grande porte. Sua arquitetura é constituída de três componentes principais: Agentes do Usuário (*UA*), que executam a descoberta do serviço em nome do cliente (aplicação ou usuário); Agentes de Serviços (*SA*), que anunciam a localização e características dos serviços, em nome dos próprios serviços; e Agentes de Diretório (*DA*), que coletam endereços de serviços e informações recebidas dos *SAs* em suas bases de dados e respondem às requisições de serviços dos *UAs*.

Serviços são anunciados utilizando uma *URL* de Serviço, que contém o endereço *IP*, número da porta e caminho para o serviço, e um *Template* de Serviço, que especifica as características do serviço e seus valores padrão. Entretanto, como o próprio nome indica, *SLP* apenas define uma forma de localizar um serviço e deixa em aberto a interação entre clientes e serviços após a descoberta.

4.1.2 JINI

A tecnologia *Jini* é uma extensão da linguagem de programação Java e foi desenvolvida pela Sun *Microsystems*, para abordar a questão de como dispositivos se conectam uns com os outros para formar uma rede *ad-hoc* simples, e como esses dispositivos provêm serviços para outros dispositivos nessa rede. *Jini* é constituída de uma arquitetura e um modelo de programação.

A tecnologia *Jini* foi criada para permitir que qualquer tipo de rede composta de serviços e dos clientes destes serviços seja facilmente montado, desmontado e mantido. *Jini* possui um modelo que define como os clientes e os serviços vão se comunicar uns com os outros. Os clientes acessam os serviços através de objetos comuns disponibilizados por entidades chamadas “*service providers*” ou “prestadores de serviços” usando qualquer tecnologia de rede.

O protocolo *Jini* é baseado em licenças (*leases*), desta forma todos os anúncios e registros são considerados válidos apenas por um período de tempo específico e relativamente curto. Clientes e serviços que são executados por muito tempo devem renovar suas licenças periodicamente, assim entidades que falham são removidas automaticamente de todos os serviços de busca quando a licença expira. Além disso, a renovação periódica de licenças pelas entidades é capaz de reconstruir o estado global no caso do travamento de um *LS* (Serviço de Consultas).

As especificações de *Jini* são abertas e podem ser usadas livremente. Entretanto, a Sun cobra uma taxa de licenciamento para uso comercial. O código de *Jini* pode ser implementado em 46k de binários de Java. Outra vantagem de *Jini* é ser independente de plataforma, mas a desvantagem é que todos os clientes, serviços e diretórios dependem direta ou indiretamente de ambientes de execução Java.

4.1.3 SALUTATION

Salutation é mais uma solução para descoberta de serviços. Um consórcio aberto de indústrias, chamado Consórcio *Salutation*, é responsável pelo desenvolvimento de sua arquitetura, que define um modelo abstrato com três componentes: Cliente, Servidor e *Salutation Manager (SLM)*. O *Salutation Manager* gerencia toda a comunicação, e faz a ponte através de diferentes meios de comunicação. Serviços registram suas capacidades com um *SLM*, e clientes consultam o *SLM* quando necessitam de um serviço. Depois de descobrir um serviço desejado, clientes podem solicitar a utilização do serviço através do *SLM*.

O protocolo *Salutation* foi projetado para ser compatível com as tecnologias de rede sem fio, mas já existem adaptações para *Bluetooth* e outras.

4.1.4 UNIVERSAL PLUG AND PLAY (UPnP)

Universal Plug and Play (UPnP) foi desenvolvido por um consórcio empresarial fundado e liderado pela *Microsoft*. Pode-se dizer que ele estende a tecnologia *Plug and Play* da *Microsoft* para o caso em que dispositivos podem ser acessados através de uma rede *TCP/IP*. *UPnP* é voltado para redes sem gerenciamento, ou seja, pequenas redes de escritório ou residenciais, onde ele habilita mecanismos *peer-to-peer* para auto configuração de dispositivos, descoberta de serviços, e controle de serviços. *UPnP* é um protocolo de descoberta de serviços orientado a dispositivos. Todas as informações sobre os serviços e comunicação estão no formato *XML*, que é independente de plataforma e de linguagem de programação o que aumenta bastante a interoperabilidade entre dispositivos.

As metas de *UPnP* são para conexão direta e simplificação da implementação de redes em casa e em escritórios. A tecnologia "Ligar e Usar" é para ligação direta entre um computador e um dispositivo.

O termo *UPnP* é derivado de "Ligar e Usar", uma tecnologia para conexão dinâmica de dispositivos a um computador. O *UPnP* segue protocolos IP padrão para permanecer independente de mídia de rede. Os dispositivos em uma rede *UPnP* podem ser conectados usando qualquer mídia de comunicação, inclusive sem fio, linha telefônica, linha de energia,

IrDA, Ethernet e IEEE 1394. Em outras palavras, qualquer meio que pode ser usado para interligar dispositivos em rede pode permitir o *UPnP*. A única preocupação deve ser se a mídia usada suporta a largura de banda necessária para o uso pretendido.

4.1.5 BLUETOOTH SERVICE DISCOVERY PROTOCOL (SDP)

O Bluetooth é uma tecnologia que permite uma comunicação simples, rápida, segura e barata entre computadores, *smartphones*, telefones celulares, mouses, teclados, fones de ouvido, impressoras e outros dispositivos, utilizando ondas de rádio no lugar de cabos. Assim, é possível fazer com que dois ou mais dispositivos comecem a trocar informações com uma simples aproximação entre eles.

Bluetooth é um padrão global de comunicação sem fio e de baixo consumo de energia que permite a transmissão de dados entre dispositivos compatíveis com a tecnologia. Para isso, uma combinação de *hardware* e *software* é utilizada para permitir que essa comunicação ocorra entre os mais diferentes tipos de aparelhos. A transmissão de dados é feita através de radio frequência, permitindo que um dispositivo detecte o outro independente de suas posições, desde que estejam dentro do limite de proximidade.

A pilha de protocolos *Bluetooth* contém o *Service Discovery Protocol (SDP)*, que é usado para localizar serviços fornecidos por ou disponíveis através de um dispositivo *Bluetooth*.

O protocolo trata de descoberta de serviços especificamente para esse ambiente, e suporta as seguintes requisições: busca por serviços pelo tipo de serviços, busca por serviços por atributos de serviços, e busca por serviços sem um prévio conhecimento das características do serviço.

4.1.6 INTENTIONAL NAMING SYSTEM (INS)

No Sistema de Nomes Intencionais (*INS*), aplicações podem ser serviços, que provêm funcionalidades ou dados, ou clientes, que solicitam acesso a esses serviços. O *INS* mantém essa base de dados dos dispositivos correntemente ativos juntamente com os atributos descrevendo seu uso pretendido. Periodicamente, os dispositivos renovam seus registros para evitar que suas entradas expirem (Rudolph, 2001).

INS suporta definição de endereços adiantada, para maior eficiência, e definição de endereços tardia, para favorecer mobilidade. Neste caso, *INS* resolve a busca de serviço para uma localização no momento da entrega (Zhu, 2002).

5. DISPOSITIVOS INTELIGENTES

Atualmente uma diversidade muito grande de equipamentos possui um microprocessador integrado, desde *smartphones* a televisões, passando pelos *tablets* e *MIDIs*. A intenção é fazer com que os dispositivos comuniquem entre si, acessem a internet e desenvolvam suas funções de forma mais prática o possível.

Os dispositivos inteligentes combinam perfeitamente com os paradigmas da computação ubíqua: eles são descentralizados, diversificados, conectados e simples de usar. Alguns dispositivos possuem inúmeras funcionalidades que são aplicadas no dia a dia dos usuários, por exemplo, acesso a internet, caixa de e-mail, jogos, etc. Enquanto outros são elaborados para operar em determinados segmentos, por exemplo, nos setores industriais e outros.

A existência de múltiplos dispositivos, seja para o acesso a informação, ao entretenimento, embutidos em utensílios domésticos, embarcados em ambientes inteligentes constitui-se em um dos desafios da computação ubíqua.

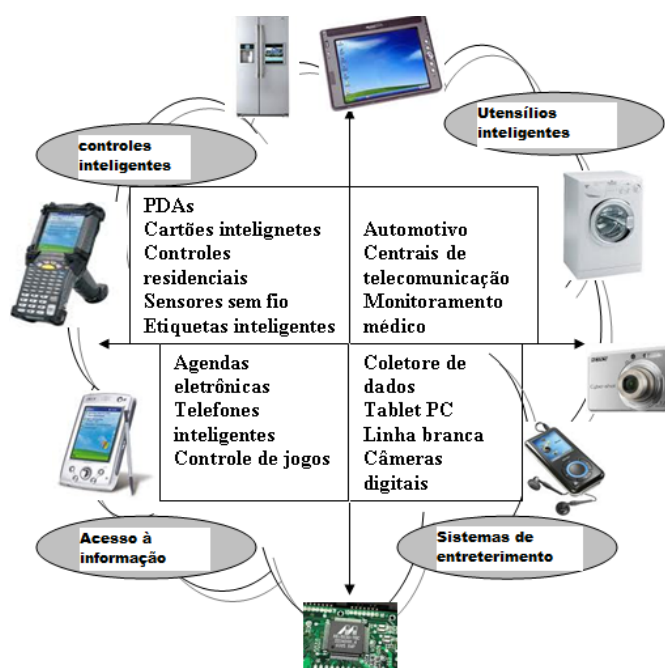


Figura 6: Classificação dos dispositivos ubíquos (Adaptado de Hansmann, 2001)

Fonte: http://www.cpdee.ufmg.br/documentos/Defesas/790/PAULO_BARROS.pdf

Devido à grande variedade de dispositivos existentes eles foram estruturados de acordo com as seguintes categorias.

5.1 UTENSÍLIOS INTELIGENTES

No ambiente ubíquo os utensílios inteligentes interagem entre si para proporcionar maior conforto ao usuário, seja em casa, no carro, no escritório, em bancos, nos hospitais, nas ruas, nos shoppings, etc. Essa categoria abrange uma gama muito grande de utensílios familiares que são melhorados por embutirem mais conectividade e inteligência através de tecnologias ubíquas.

Os utensílios inteligentes aumentam a funcionalidade dos equipamentos já conhecidos que hoje oferecem um conjunto de serviços dedicados. Exemplos de aplicações de utensílios inteligentes podem ser:

- Otimização do consumo de energia (aquecimento seletivo – por cômodo da casa; aquecimento diferenciado por ocupação da casa, aquecimento de água em função do perfil dos moradores);
- Manutenção de utensílios (diagnóstico e atualização remota de micro-código em utensílios da linha branca);
- Comunicação entre etiquetas de roupa e a máquina de lavar;
- Instrumentos de monitoramento médico;
- Dispositivos computacionais integrados a automóveis disponibilizando para os usuários serviços como sistemas de navegação, informações sobre falhas em componentes do automóvel, monitoramento dos dados sobre os veículos pelas montadoras, acesso a informações e entretenimento.

Esses dispositivos podem ser acessados e operados remotamente via *web*, por exemplo.

5.2 CONTROLES INTELIGENTES

Os controles inteligentes caracterizam-se por serem muito pequenos podendo ser integrados a lâmpadas, interruptores, termostatos, sensores, atuadores etc., em aplicações que variam de controle de segurança residencial (sensores em portas e janelas para detectar a entrada de intrusos, atuadores para acender/apagar lâmpadas específicas em horários

específicos etc), a controle de comodidades para o usuário (sensores de temperatura, atuadores para ligar/desligar/programar sistemas de aquecimento/resfriamento de ambiente residencial etc.).

Os controles são conectados a redes domésticas e gerenciados local ou remotamente, através da *Web* ou *applets Java*, em aplicações locais.

Estes dispositivos também podem comunicar-se entre si, por exemplo, em uma aplicação onde um relógio com alarme envia dados para ajustar um aquecedor em um banheiro. São exemplos de controles inteligentes:

- Controles residenciais utilizados em aplicações de domótica;
- Termostatos eletrônicos;
- Etiquetas inteligentes;
- Controles de bombas de ar, água, gás, etc.;
- Cartões inteligentes.

5.3 DISPOSITIVOS DE ACESSO À INFORMAÇÃO

Os dispositivos de acesso á informações tem por objetivo promover a comunicação entre os usuários, acesso a informação, anotações, etc, e ainda visa a expandir a capacidade humana, principalmente para o trabalho. Eles tendem a auxiliar os usuários no seu dia a dia, tornando as tarefas mais fáceis de serem realizadas, agilizando-as a fim de ganhar tempo. Nesses dispositivos geralmente encontra-se agenda de endereços, calendários, funcionalidades de e-mail e outros.

A seguir são apresentadas algumas das principais categorias de dispositivos de acesso ás informação disponíveis no mercado.

5.3.1 Assistente digitais pessoais (PDAs)

Um dos tipos de dispositivos de acesso a dados mais conhecidos são os assistentes pessoais digitais, do inglês *Personal Data Assistant*, ou *PDAs*. Eles caracterizam-se pela tela sensível ao toque. A interação com o dispositivo é baseada em apontador, ou *stylus*, e a entrada de texto normalmente é feita por reconhecimento de escrita manual ou teclado virtual.

Dentre os serviços oferecido estão agenda de endereços e telefones, calendário, editor de texto, calculadora, despertador, e-mail, jogos, navegadores, relógios globais, planilhas, aplicações financeiras, reprodutores de mídia, recursos de desenhos e outros.

Os *PDA*s vêm perdendo espaço para os telefones inteligentes que além de oferecerem as mesmas funcionalidades que os *PDA*s ainda possuem telefone celular integrado.



Figura 7: Exemplos de PDAs

Fonte: http://www.google.com.br/images?hl=pt-BR&gbv=2&tbs=isch%3A1&sa=1&q=PDA&aq=f&aqi=g3&aql=&oq=&gs_rfai=

5.3.2 Coletores de dados

Outra classe de dispositivos de acesso à informação é a dos coletores de dados. Eles, normalmente, são maiores e mais robustos do que os *PDA*s, sendo utilizados em aplicações corporativas, como serviços de leitura de medidores de energia, leitura de hidrômetros de água e controle de almoxarifados. As características dos coletores de dados podem variar muito de um modelo para outro, sendo que alguns possuem câmera digital, *GPS* e celular integrados.



Figura 8: Exemplo de coletores de dados

Fonte: http://www.google.com.br/images?q=coletores+de+dados&hl=pt-BR&gbv=2&tbs=isch:1&ei=KRpGTMDfJ8T6lwej_MDuAw&sa=N&start=90&ndsp=18

5.3.3 Telefones inteligentes

Os telefones inteligentes ou *smartphones*, integram em um só equipamento todas as funcionalidades de um *PDA* e um telefone celular. Atualmente eles são a classe de

dispositivos de acesso a dados mais vendidos, conquistando o mercado que antes pertencia aos *PDA*s.

O *smartphone* é um telefone celular com funcionalidades avançadas que podem ser estendidas por meio de programas executados no seu Sistema Operacional. Os Sistemas Operacionais dos *Smartphones* são "abertos" (não confundir com código-fonte aberto), o que significa que é possível que qualquer pessoa desenvolva programas que podem funcionar nesses telefones. Usualmente um *smartphone* possui características mínimas de hardware e software, sendo as principais: capacidade de conexão com redes de dados para acesso à internet, capacidade de sincronização dos dados do organizador com um computador pessoal e agenda de contatos que utiliza toda a memória disponível no celular.

Atualmente os principais sistemas operacionais usados pelos *smartphones* são: Symbian, Blackberry OS, Windows Mobile, iOS, Samsung Bada, Palm WebOS, Android e Maemo, sendo que os três últimos são baseados em Linux.



Figura 9: Exemplos de telefones inteligentes

Fonte: <http://www.google.com.br/images?q=smartphone&hl=pt-BR&gbv=2&tbs=isch:1&ei=6ERITL-QGcH6lwf4wP2OCw&sa=N&start=18&ndsp=18>

5.3.4 Tablet PCs

Um *tablet PC* se parece com um notebook pequeno, com a tela sensível ao toque. O nome *tablet PC* vem do fato da tela sensível ao toque “girar” para o lado de fora do dispositivo, de modo que ele se transforme em uma espécie de prancheta. Com dimensões similares as de um caderno escolar, a tela possibilita o reconhecimento da escrita do usuário com o auxílio de canetas do tipo *stylus*, semelhante a um *PDA*.

A maior parte dos *tablet PCs* utilizam o Windows XP Tablet PC Edition como sistema operacional, embora tendam a utilizar no futuro o Windows Vista e Windows Seven, dado que algumas versões do novo sistema operacional da Microsoft incorporam já as

funcionalidades dedicadas ao *tablet PC*, entre elas o Windows Journal e o famoso jogo InkBall. São poucos os que utilizam o sistema operacional Linux.

O *tablet PC* vem sendo estudado em vários países. Ele já está circulando no mercado brasileiro. Nos Estados Unidos já é possível encontrá-los por menos de mil dólares.

A popularização deste tipo de computador começa a se dar com o lançamento do iPad, da Apple - marca responsável pela difusão dos *players* de MP3 com o iPod. A HP, famosa marca de computadores, *notebooks* e acessórios, entrará na disputa com o *Slate Tablet PC*. Apesar de chegar atrasada no mercado, a HP pode se tornar competitiva pois inseriu em seu equipamento ferramentas que não constam no aparelho da Apple (como câmera para videoconferência, portas USB, entrada para cartão de memória e *plugin* para visualização de documentos em *Flash*), e o sistema operacional é o *Windows 7*.

Estes equipamentos possuem capacidade de processamento e configuração semelhantes as de um computador desktop, possuindo mais recursos computacionais que as outras classes de dispositivos ubíquos de acesso a informação.



Figura 10: Exemplos de Tablet PCs

Fonte: <http://www.google.com.br/images?q=tablet+pc&um=1&hl=pt-BR&gbv=2&tbs=isch:1&ei=IUxITKGEOcKclgeixYCeCw&sa=N&start=0&ndsp=18>

5.4 SISTEMAS DE ENTRETENIMENTO

Os sistemas de entretenimento envolvem dispositivos cujo uso é voltado para o lazer do usuário, por exemplo, sistemas de TV (via cabo, satélite, microondas), *Set-top boxes* (caixa sobre a TV que promove a interface entre os provedores do serviço de difusão e a TV do consumidor), consoles de jogos (exemplos: *Dreamcast* da Sega, *Playstation 2* da Sony, *Dolphin* da Nintendo e *X-box* da Microsoft), câmeras digitais, brinquedos inteligentes, reprodutores de música MP3, etc.

Claro que muitos dispositivos podem oferecer uma combinação de aplicações e, dessa maneira, se encaixarem em mais que uma dessas quatro categorias.

6. AS REDES DE SENSORES

Com o avanço tecnológico às redes de sensores tem tomado proporções ainda maiores, elas são usadas em diversas aplicações, por exemplo, no monitoramento, no rastreamento, na coordenação e no processamento de diversos contextos.

Na elaboração de ambientes inteligentes os sensores passam a ser itens indispensáveis pois estes instrumentos são os responsáveis pela captação dos estímulos físicos do ambiente ao qual estão inseridos. Atualmente existem diversos tipos de sensores que realizam inúmeras aplicações como, sensores de luz, sensores de calor, sensores de som, sensores de temperatura, sensores de movimento e outros.

Um sensor é um dispositivo que detecta um estímulo físico (calor, luz, som, pressão, campo magnético, movimento) e transmite um impulso correspondente.

Alguns sensores respondem com sinal elétrico a um estímulo, isto é, convertem a energia recebida em um sinal elétrico. Nesse caso, podem ser chamados de transdutores. O transdutor converte um tipo de energia em outro. É geralmente composto por um elemento sensor e uma parte que converte a energia proveniente dele em sinal elétrico.

6.1 REDES DE SENSORES SEM FIO

As redes de sensores sem fios (*RSSFs*) são diferentes das redes de computadores tradicionais em vários aspectos, essa rede geralmente possui um grande número de nodos distribuídos, têm restrições de energia e deve possuir mecanismo pra auto-configuração e adaptação devido a problemas como falhas de comunicação e perda de nodos. Um nodo ou nó representa cada ponto de inter-conexão com uma estrutura ou rede, independente da função do equipamento representado por ele.

São vários os avanços que estimulam o desenvolvimento e o uso de sensores inteligentes em diversas áreas, geralmente tem-se em um único chip vários sensores que são controlados pela lógica do circuito integrado e com interface de comunicação sem fio. O termo sensor inteligente está associado a um chip que possui um ou vários sensores com capacidade de processamento de sinais e comunicação de dados.

Nas redes de sensores sem fio cada nodo é equipado com uma variedade de sensores, que podem ser, sísmicos, infravermelho, vídeo, calor, temperatura, pressão, etc. Os nodos podem ser organizados em grupos onde pelo menos um dos sensores deve ser capaz de detectar um evento que aconteça na região, processar o evento e tomar uma decisão se deve ou não enviar informações do evento a outros nodos da rede.

As redes de sensores sem fio (*RSSFs*) podem ser classificadas quanto ao seu tipo em homogêneas ou heterogêneas, ou seja, as aplicações de monitoração de segurança podem utilizar sensores de imagem e acústicos, embutidos no mesmo nodo sensor ou em nodos diferentes. Neste caso, os tipos de dados coletados pela rede de sensores são imagens, vídeos e sinais de áudio o que faz dessa uma rede sem fio heterogênea, isso devido a variedade de dados que podem ser coletados pelos nodos sensores da rede. Em outro caso, se um nodo sensor é capaz de coletar do ambiente apenas sinais temperatura então definimos esta como uma rede de sensores sem fio homogênea.

As *RSSFs* já são uma realidade e integram ambientes inteligentes em diversos campos, por exemplo, produção industrial, distribuição de água, gás e energia, áreas industriais onde o monitoramento dos dados é de difícil acesso, extração de gás e petróleo, indústria de aviação, entre outras.

6.2 CARACTERÍSTICAS DAS REDES DE SENSORES SEM FIO (RSSFs)

As redes de sensores sem fio apresentam características particulares dependendo das áreas nas quais estão inseridas. A seguir será apresentado algumas dessas características e particularidades das *RSSFs*.

Endereçamento dos sensores ou nodos - Dependendo da aplicação, cada sensor pode ser endereçado unicamente ou não. Por exemplo, sensores embutidos em peças numa linha de montagem ou colocados no corpo humano devem ser endereçados unicamente se somente se deseja saber exatamente o local de onde o dado está sendo coletado. Por outro lado, sensores monitorando o ambiente numa dada região externa possivelmente não precisam ser identificados individualmente já que o ponto importante é saber o valor de uma determinada variável nessa região.

Agregação dos dados - Indica a capacidade de uma *RSSF* de agregar e tratar dos dados coletados pelos sensores. Caso a rede tenha essa funcionalidade, é possível reduzir o número de mensagens que precisam ser transmitidas por ela.

Mobilidade dos sensores - Indica se os sensores podem se mover ou não em relação ao sistema em que estão coletando dados. Por exemplo, sensores colocados numa floresta para coletar dados de umidade e temperatura são tipicamente estáticos, enquanto sensores colocados na superfície de um oceano para medir o nível de poluição da água são móveis. Sensores colocados no corpo de uma pessoa para monitorar o batimento cardíaco durante o seu dia de trabalho são considerados estáticos.

Quantidade de sensores - Redes contendo de 10 a 100 mil sensores são previstas para aplicações ambientais como monitoramento em oceanos e florestas. E na maior parte das aplicações os sensores serão estáticos em relação à sistema sensoramento.

Limitação da energia disponível - Em muitas aplicações, os sensores serão colocados em áreas remotas, o que não permitirá facilmente o acesso a esses elementos para manutenção. Neste cenário, o tempo de vida de um sensor depende da quantidade de energia disponível. Assim, o projeto de qualquer solução para esse tipo de rede deve levar em consideração o consumo, o modelo de energia e o mapa de energia da rede.

Tarefas colaborativas - O objetivo principal de uma *RSSF* é executar alguma tarefa colaborativa onde é importante detectar e estimar eventos de interesse e não apenas prover mecanismos de comunicação.

A *RSSFs* são sistemas auto-organizados formados por nodos sensores que podem espontaneamente criar um rede não premeditada, agrupando-se e adaptando-se dinamicamente quando ocorrem falhas ou defeitos em algum dispositivo. Um sistema de localização é um exemplo de *RSSFs*, que envolve uma coleção ou um grupo de nodos trabalhando a fim de alcançar um objetivo.

6.3 MANUTENÇÃO DAS *RSSFs*

Assim como em qualquer outro material que pode ser degradado às manutenções são um fator importantíssimo para as redes de sensores sem fios. Seu objetivo é prolongar o tempo de vida útil da rede, reduzir os imprevistos, garantir que os serviços para qual a rede foi desenvolvida sejam realizados. Com o passar do tempo alguns nodos de rede podem perder energia e isso restringe a sua capacidade de funcionamento de forma parcial ou total.

As manutenções realizadas podem ser preventivas, corretivas ou adaptativas depende do evento que ocorrer na rede. Essas manutenções devem existir durante todo o tempo de existência das *RSSFs* pois o não funcionamento ou o funcionamento parcial pode comprometer as atribuições dos ambientes na qual essas estão inseridas. Isso não significa que se um sensor deixar de funcionar em um ambiente inteligente todas as suas propriedades serão perdidas, comumente pensa-se nesse fator ao elaborar um *RSSFs* e então cria-se formas de um sensor suprir as ações de outro caso o mesmo venha a faltar, seja por ações do ambiente ou por defeitos que possa ocorrer.

A manutenção de qualquer sistema ou rede de sensores tem de ser tratada de maneira séria e nunca pode sofrer alterações em seus períodos de realizações de serviços, elas precisam seguir um cronograma baseado no tipo de rede que se utiliza, levar em conta os materiais que formam essa rede, considerar o ambiente em que a rede está operando e levar em consideração os fatores naturais.

6.4 COLETA DE DADOS

O principal objetivo de uma rede de sensores é coletar informações de uma região de observação específica. Por isso as redes de sensores são bastante utilizadas na elaboração de um ambiente inteligente, elas captam as informações do meio e através de comandos pré-programados em um sistema específico ações ocorrem em resposta ao contexto que foi encontrado naquele dado momento.

Um bom exemplo desse cenário de coleta de dados é quando as pessoas vão ao shopping e ao aproximar-se das portas que aparentemente estão fechadas elas de repente se abrem permitindo a livre passagem das pessoas ali presentes. Essa ação acontece porque naquele ambiente possui alguns sensores de presença que detectam quando as pessoas se aproximam, então é feito a coleta dos dados e esses dados são distribuídos pela rede até chegarem a uma central de comandos em que um *software* programado para executar a função “abrir a porta” entra em ação. No exemplo é possível perceber três variáveis importantes, que são: a coleta dos dados, o processamento dos dados e a ação correspondente.

A coleta dos dados é o fator muito importante em um ambiente inteligente pois eles são o que determinam a resposta do sistema, por isso estes dados não podem sofrer influência de outros fatores, então tem-se uma preocupação especial com as redes de sensores a fim de evitar a captação de dados que sejam desnecessários.

7. SOFTWARES PARA AMBIENTES INTELIGENTES

Um programa de computador é composto por uma seqüência de instruções, que é interpretada e executada por um processador ou por uma máquina virtual. Em um programa correto e funcional, essa seqüência segue padrões específicos que resultam em um comportamento desejado. Um programa pode ser executado por qualquer dispositivo capaz de interpretar e executar as instruções de que é formado.

Quando um *software* está representado como instruções que podem ser executadas diretamente por um processador dizemos que está escrito em linguagem de máquina. A execução de um *software* também pode ser intermediada por um programa interpretador, responsável por interpretar e executar cada uma de suas instruções.

Ao elaborar um *software* os desenvolvedores utilizam plataformas programáveis ou linguagens de programação que mais se adéquem aos requisitos coletados por toda a equipe de trabalho (analistas de sistemas, projetistas e os demais integrantes). A programação de um software é considerada a etapa trivial do desenvolvimento, pois é a partir desse momento que surge de forma concreta as funcionalidades dos dispositivos apresentados.

Ao pensar em um *software* para ambientes inteligentes é preciso saber que cada funcionalidade executada requer instruções diferentes, um ambiente nem sempre possui as mesmas características que outro, as instruções dadas a determinado dispositivo presente no ambiente pode não equivaler a outro dispositivo semelhante, é por isso que, os especialistas costumam a dizer que o trabalho com *softwares* para ambientes inteligentes tendem a ser intrigantes, pois sempre surgem novos desafios e funcionalidades diversas.

Existem várias formas de classificar um *software*, exemplo, *software* de aplicações, de ambientes e também como sistemas operacionais. As aplicações são aquelas que de alguma maneira provêm serviços para os usuários, já os ambientes são as camadas intermediaras que fornecem a estrutura para a programação e os sistemas operacionais são aqueles que fornecem a estrutura necessária para o acesso ao *hardware* dos dispositivos.

Nigel Davies (2002) aponta em seu trabalho alguns outros desafios para o desenvolvimento de sistemas ubíquos utilizados em ambientes inteligentes:

- Interação de componentes: componentes e plataformas devem ser projetados para interagir com os diferentes componentes existentes em ambientes saturados com dispositivos.

- Sensibilidade ao contexto e adaptação: os ambientes são altamente mutáveis, logo as aplicações e os componentes devem ser projetados para pressentir e se adaptar a essas mudanças.
- Gerenciamento apropriado de mecanismos e políticas: como o número de componentes de alto e diferenciado, o gerenciamento torna-se problemático. Há a necessidade de estabelecer políticas e interfaces padrões para gerenciar essa quantidade expressiva de componentes heterogêneos.
- Associação de componentes e análises de tarefas: determinar as intenções dos usuários e implementar associações entre os componentes para ajudá-lo a alcançar seu objetivo.
- Modelos economicamente viáveis e infra-estrutura de suporte: usuários podem até pagar para viver em um mundo com computação ubíqua, mas poucos estarão dispostos a pagar para ter acesso a determinados componentes ou serviços.
- Interface com o usuário: como há diversas aplicações simultâneas no ambiente, definir e apresentar a interface ativa ao usuário de um problema, conseqüentemente há a necessidade de integração e negociação dessas interfaces.
- Soluções técnicas, legais e sociais para privacidade e segurança: as legislações não abordam questões de privacidade no contexto da computação ubíqua, nem o usuário está consciente da importância de suas informações. Logo, medidas devem ser implementadas para endereçar essas situações.

Durante todo o processo de produção de *software* deve-se considerar a limitação existente nos dispositivos, tais como, capacidade de memória, processamento, tempo de duração da bateria, as redes de comunicação sem fio, a grande diversidade de dispositivos, etc., já no caso de sistemas operacionais é preciso focar nas características dos dispositivos e no objetivo do seu uso. As limitações não param, elas estão por toda parte, seja no uso da linguagem de programação ou até mesmo na escolha da linguagem a ser utilizada. Os desenvolvedores de aplicações para ambientes ubíquos contam com uma gama muito limitada

de linguagem de programação para realizarem seu trabalho, as mais usadas são: *Java*, *C*, *C++* e também algumas linguagens proprietárias.

Na área de modelagem de sistemas ubíquos têm surgido vários trabalhos que procuram responder aos desafios colocados (Christensen, 2002; Ingstrup, 2003). Vários investigadores têm proposto uma abordagem na qual modelo de sistema, e em particular modelos de arquitetura do software, sejam mantidos em *run-time* e utilizados como base para reconfiguração (Cheng, 2002).

Talvez por falta de padrões de desenvolvimento ainda seja tão complicado a questão da produção de *software* para ambientes inteligentes. Acredita-se que o problema não está relacionado à tecnologia e que o *hardware* existente é capaz de suportar as funcionalidades características dos ambientes inteligentes fazendo com que os dispositivos agregados ao meio trabalhem de forma integrada, sejam capazes de aprender com eventos já ocorridos, possuam sensibilidade de contexto, que sejam tolerantes a erros e falhas que vierem a ocorrer não prejudicando assim o funcionamento do sistema.

No desenvolvimento de aplicações para *softwares* que serão usados em ambientes ubíquos é preciso considerar as seguintes restrições:

- Tamanho limitado da tela, capacidade limitada de entrada de dados, poder limitado de processamento, memória, armazenamento persistente e durabilidade da bateria;
- Largura de banda limitada e conectividade intermitente (o que os dispositivos esperam encontrar em termos de conectividade).

Para driblar as limitações citadas acima as aplicações devem seguir as seguintes instruções:

- Conectar-se a rede apenas quando necessário;
- Consumir da rede apenas os dados que realmente precisa;
- Permanecer útil mesmo quando desconectado.

Com a emergência de ambientes de computação ubíqua as práticas de engenharia de *software* têm de evoluir, na medida de que as infra-estruturas de *software* requeridas pela computação ubíqua diferem sensivelmente das que hoje permitem ambientes tecnológicos de informação (Bonnet, 2003).

Segundo Siegel (Siegel, 2001), a *OMG (Object Management Group)*, em 2001, com o objetivo de reduzir a complexidade e acelerar a introdução de novas tecnologias, definiu a

Arquitetura *MDA (Model Driven Architecture)* como uma abordagem para o uso de modelos no desenvolvimento de *software*, com especificações que contribuem para a interoperacionalidade, reutilização e portabilidade de componentes de *software* e modelos de dados com base em modelos: uma maior utilização de normas multiplataforma; a possibilidade de cada norma poder ser implementada na plataforma pretendida; a geração de código de aplicação a partir de modelos *MDA*; o aumento de qualidade final do sistema projetado. Já existem atualmente ambientes de desenvolvimento integrado de suporte à abordagem *MDA (Model Driven Architecture)*.

A visão para as arquiteturas baseadas na norma *MDA* prevê a possibilidade de desenvolver *software* capaz de descobrir automaticamente as propriedades do seu ambiente e de se adaptar dinamicamente às mudanças observadas, modificando o seu comportamento (Poole, 2001).

A proliferação da computação no mundo físico promete mais do que disponibilidade ubíqua da infra-estrutura de computação; sugere novos paradigmas de computação inspirados pelo acesso constante a informação e capacidades computacionais (Abowd, 2000).

7.1 SISTEMAS OPERACIONAIS (SO)

Um sistema operacional é um programa ou um conjunto de programas cuja função é gerenciar os recursos do sistema (definir qual programa recebe atenção do processador, gerenciar memória, criar um sistema de arquivos, etc.), além de fornecer uma interface entre o computador e o usuário. É o primeiro programa que a máquina executa no momento em que é ligada (num processo chamado de bootstrapping) e, a partir de então, não deixa de funcionar até que o computador seja desligado. O sistema operacional reveza sua execução com a de outros programas, como se estivesse vigiando, controlando e orquestrando todo o processo computacional.

Basicamente não existem diferenças significativas entre os sistemas operacionais utilizados nos ambientes ou dispositivos ubíquos dos sistemas operacionais usados nos *PCs*, é sempre mais fácil encontrar semelhanças do que diferenças, no entanto os sistemas operacionais utilizados em ambientes ubíquos podem ser destacados pela especificidade dos dispositivos para os quais foram criados. Geralmente esses *SOs* são desenvolvidos para atuar em um determinado dispositivos que tem como função desenvolver muito bem uma ou mais tarefas.

A competição na área de sistemas operacionais não é comum no ramo dos *PCs* mas em compensação entre os *SOs* ubíquos o número de competidores é bem extenso, por exemplo, *Palm OS* (utilizado nos *PDA*), *EPOC* (utilizados em celulares), *W/CE* (*Consumer Electronics*), *Java Card* e *W/Smart Card* para *Smart Cards*, *QXN*, *VxWorks*, etc.

7.1.1 Palm OS

O sucesso do *PALM* deve-se ao fato dele ser projetado especificamente para *PDA*s, ser fácil de usar, oferecer um número limitado de características, mas que são altamente otimizadas, levando ao uso de pouca memória e de *CPU*, o que garante vida mais longa de bateria. O *PALM* foi desenvolvido pela *PalmSource*, Inc., que detém o domínio de cerca de 70% do mercado dos dispositivos de mão.

Toda a memória de um *Palm* reside em cartões de memória. Cada cartão de memória tem um espaço de endereço máximo teórico de 256MB. O espaço que é deixado para a aplicação é muito pequeno (menor que 36KB), por isso a memória da aplicação deve ser mantida a menor possível, com ações do tipo: evitar geração de estruturas grandes de pilhas, evitar uso de variáveis globais e cópias de base de dados na memória dinâmica, etc.



Figura 11: Palm OS em execução

Fonte: <http://www.intomobile.com/2008/10/26/access-shows-off-access-linux-platform-30-alp-30-as-next-gen-palm-os/>

<http://www.anti-spyware-adware.com/software-programs/home-education/food-beverage/9-95-drinks-for-palm-os.html>

O *Palm OS* é orientado a eventos que são tratados por tratadores de eventos. Eventos típicos incluem: ações do usuário em interação com a aplicação; notificações do sistema (alarme de tempo) e outros específicos da aplicação. Os programadores devem evitar código

que faça processamento enquanto espera por um evento (espera ociosa) para economizar bateria.

O *Palm OS* foi projetado para ser fácil de ser utilizado por meio de uma interface gráfica sensível ao toque, do tipo *touch screen*. Ele é disponibilizado com um conjunto de aplicações básicas, como agenda de telefones, calendário, calculadora e outras.

7.1.2 BlackBerry OS

Esse é um sistema operacional proprietário utilizado nos *smartphones BlackBerry*, fabricados pela empresa canadense *Research in Motion (RIM)* (*Black Berry Developers* (2008)). Segundo Canalys (2008), este sistema operacional obteve 11% do mercado mundial de sistemas operacionais para *smartphones* no ano de 2007. A principal característica do *BlackBerry* é permitir que seus usuários possam receber e-mails sempre que o dispositivo estiver conectado a uma rede de dados.

O *BlackBerry OS* é focado na facilidade de operação e foi originalmente projetado para o mercado corporativo. Recentemente foi liberado o *SDK*, do inglês *Software Development Kit*, que permite que desenvolvedores implementem aplicações para este sistema operacional.

7.1.3 Symbian OS

Symbian OS é um sistema operacional criado para rodar nos celulares com suporte para câmeras fotográficas, *MMS*, *wireless*, bluetooth, entre outras funções. Este sistema operacional é predominantemente baseado em um ambiente gráfico bastante simples. Atualmente é utilizado na maioria dos recentes modelos de celulares dos grandes fabricantes. A grande preocupação do *Symbian OS* é evitar ao máximo o desperdício dos recursos do celular, como bateria e memória. Para isso ele conta com diversos mecanismos que são eficientes ao tratar com esses problemas.

O *Symbian OS* é desenvolvido exclusivamente para dispositivos móveis por fabricantes de celulares como *Nokia*, *Siemens*, *Samsung* e *Motorola*, que se uniram para formar a *joint venture* denominada *Symbian*. O *Symbian* suporta vários protocolos de comunicação, tais como *TCP/IP*, *WAP*, *GSM* e *Bluetooth*, e permite que as aplicações sejam escritas em linguagens como *Java* e *C++*.

O *Symbian* é um sistema totalmente modular, permitindo que cada empresa crie sua própria interface. Portanto, este sistema não tem uma interface definida, podendo ser um

simples sistema de textos em telas monocromáticas ou um completo sistema operacional tão potente quanto o *Palm OS* ou *Windows Mobile*. Segundo Canalys (2008), o Symbian possui a maior participação mundial nas vendas de *smartphones*, atingindo 65% de *Market Share* em 2007, mas está atrás de outras companhias no mercado Norte Americano, que é relativamente pequeno, mas com grande visibilidade.



Figura 12: Interfaces Symbian OS

Fonte: http://www.google.com.br/images?um=1&hl=pt-br&biw=1366&bih=613&tbs=isch%3A1&sa=1&q=interface+symbian+os&aq=f&aqi=&aql=&oq=&gs_rfai=

7.1.4 Windows Mobile

O *Windows Mobile* é um sistema operacional compacto, desenvolvido para rodar em dispositivos móveis como Pocket PCs, Smartphones e aparelhos de multimídia em geral. Projetado para ser capaz de realizar boa parte do que é possível em uma versão *PC* do Windows, o sistema vem com um conjunto de aplicações básicas bem conhecidas no mundo dos PCs, tais como o Word, Excel, PowerPoint, *Windows Media Player*.

Através do *Windows Mobile* é possível ter acesso a várias tarefas, como, anotações, novos e-mails, calendário, dados do proprietário, as tarefas que estão em execução, etc., O *Windows Mobile* já passa de sua terceira versão e parece que seu sucesso não para de aumentar, cada vez mais ele esta sendo utilizado para o uso de dispositivos ubíquos. Talvez esse sucesso se deva as importantes ferramentas que são implementadas em suas versões tornando o trabalho dos usuários cada vez mais fáceis.

8. SEGURANÇA

Segurança está relacionada com a proteção de um conjunto de dados, no sentido de preservar o valor que possuem para um indivíduo ou uma organização. São características básicas da segurança da informação os atributos de confidencialidade, integridade e disponibilidade. O conceito se aplica a todos os aspectos de proteção de informações e dados.

Os incidentes de segurança da informação vêm aumentando consideravelmente ao longo dos últimos anos e assumem as formas mais variadas, como, por exemplo: infecção por vírus, acesso não autorizado, ataques *denial of service* contra redes e sistemas, furto de informação proprietária, invasão de sistemas, fraudes internas e externas, uso não autorizado de redes sem fio, entre outras (Promon, 2005).

Devido á alta conectividade dos dispositivos, sistemas de proteção e *firewalls* deverão ser implantados de forma embarcada a fim de proteger os usuários e o próprio *hardware* de acessos não autorizados, roubo de conteúdo ou mesmo vandalismos.

Os sistemas computacionais possuem um objetivo geral no que diz respeito à segurança. Estes objetivos tem como base os seguintes atributos:

- Integridade da informação contra adulteração dos dados;
- Confidencialidade dos dados contra exposição dos mesmos;
- Disponibilidade do sistema contra recusa de serviços.

Quando se fala em segurança de sistemas ubíquos a situação torna-se ainda mais crítica, porque, os paradigmas da computação ubíqua atribuem aos dispositivos e os serviços computacionais como parte do cotidiano das pessoas, fazer com que os recursos computacionais espalhados pelo ambiente tornem-se transparentes, ou seja, torná-los tão imperceptíveis que os mesmos possam ser usados sem que sejam notados por seus utilizadores. Então surge o grande problema, a segurança. Devido ao aumento das funcionalidades e disponibilidade dos serviços de computação ocorreu um exorbitante aumento do uso de redes de comunicação, transferência e transporte de dados, hardwares com suportes para usarem as mais diversas redes e acessar conteúdos via internet, acesso a redes sem fio, etc., e com isso os dados de diversos usuários estão circulando por inúmeras redes a fim de atender uma determinada solicitação.

Na ocorrência desses fatos pessoas mau intencionadas tendem a se aproveitar de informações alheias para se beneficiarem de alguma maneira.

No processo de segurança principalmente se tratando de ambientes ubíquos é preciso ter muita cautela, deve-se tomar todas as precauções cabíveis a fim de evitar transtornos futuros. É preciso evitar que alguns eventos ocorram, por exemplo, a má configuração de computadores e sistemas, principalmente os específicos em segurança, como *firewalls*, a utilização de versões antigas de softwares pode contribuir para o aparecimento de algumas falhas de segurança, pois estes podem conter erros já conhecidos que facilitem a invasão. As falhas de software são eventos que ocorrem com frequência e pode fazer com que o *software* funcione inapropriadamente deixando brechas que podem ser usadas para acessar conteúdos restritos ou pessoais, além do mais, existe a falta de conhecimento por parte do usuário, causa essa que poderia evitar muita dor de cabeça.

Existem também algumas formas de dificultar o acesso indesejado. Seja através de autenticações de usuários, de *firewalls*, de sistemas de criptografia, etc., é preciso estar sempre atento, no processo de autenticação, um usuário prova que é quem ele diz que é (dispositivo e servidor podem sofrer autenticação também). Dependendo da aplicação, apenas uma senha não é suficiente ou não é viável. No controle do aquecimento doméstico, por exemplo, a idéia de proteger o patrimônio e o bem estar apenas com uma senha que eventualmente pode ser “quebrada” por *crackers*, não é muito alentador. Alternativas como a identidade eletrônica, que usa cartões inteligentes e criptografia, são usadas.

A criptografia também é muito usada e funciona da seguinte forma, um transmissor encripta os dados usando uma chave e o receptor das informações decripta os dados novamente em uma forma utilizável usando uma chave. A segurança de um dispositivo varia muito de acordo com seu tipo, o sistema operacional utilizado, o seu *hardware*, a forma como as aplicações são executadas, todos esses fatores são importantes quando o assunto é segurança de dispositivos ubíquos.

Um ponto importante na segurança dos dispositivos está no poder dos projetistas da aplicação: na implementação de aplicações que precisam de segurança, o projetista deve estar ciente do nível de segurança que o dispositivo cliente suporta (Araujo, 2003).

Durante alguns anos, a criptografia e a garantia de integridade de dados foi considerada a premissa para a segurança de um sistema computacional (Schneier, 2000). Com a evolução dos sistemas computacionais interligados por redes e o surgimento de novas necessidades e, conseqüentemente, novas ameaças à informação, esta preocupação singular ganhou novas companhias (Schneier, 2003).

Segundo Schneier (2000), há alguns anos vêm se desenvolvendo protocolos de criptografia que visam garantir, dentro de níveis de risco calculados, a segurança das

informações durante o trajeto ou armazenamento. Assim, pode-se dizer que os protocolos de criptografia deixaram de ser o elo mais fraco em cadeias que visam garantir a segurança da informação (Stajano, 2001).

Além do implemento de novas técnicas de segurança utilizados nos sistemas baseados em ambientes ubíquos é preciso mais, se a ubiqüidade das aplicações atingirem os níveis previstos por Weiser a questão deixa de ser apenas algo que tenha relação com a automação de ambientes, tornando-os inteligentes e, passa para uma esfera jurídica que terá de elaborar leis que garantam a privacidade das pessoas e que dificultem o acesso a informações dos indivíduos.

9. CONSIDERAÇÕES FINAIS

De acordo com as premissas apresentadas por Mark Weiser em 1991, a evolução constante das funcionalidades dos serviços computacionais impulsionaram o desenvolvimento da computação ubíqua, tornando possível o surgimento de diversos ambientes inteligentes, por exemplo, os ambientes náuticos, citados na seção 3. Falar de ambientes inteligentes envolve diversos outros fatores, tais como dispositivos ubíquos, consciência de contexto, redes de sensores, redes sem fio, etc.

Além de envolver fatores tecnológicos, os ambientes inteligentes precisam associar-se, interagir aos fatores humanos, seja através de dispositivos palpáveis ou não. Interfaces de fácil acesso desenvolvidas para que os usuários utilizem os recursos computacionais disponíveis devem estar presentes por todo o meio de maneira onipresente, ou seja, de uma forma que o usuário perceba ou não o uso do serviço. Sendo assim, dispositivos embarcados no ambiente e conectados entre si através de diferentes tipos de redes executam as solicitações dos usuários através de comandos pré-programados e ainda tem a capacidade de aprender com determinada situação e executar os serviços sem instrução explícita do usuário, a chamada consciência de contexto.

Conhecer o meio ao qual está agregado reconhecendo as mudanças e respondendo-as de modo que as necessidades dos usuários sejam satisfeitas é um exemplo de consciência de contexto. Na maior parte os dados são obtidos através de sensores espalhados pelo meio que captam alterações, movimentos, temperatura, luminosidade, etc., o sistema processa os dados e através de comandos pré-programados geram reações que satisfaçam os usuários, por exemplo, ligar ou ascender uma lâmpada, ajustar a temperatura do ar condicionado ou aquecedor, etc. Apesar de apresentar diversos dispositivos com as mais variadas funcionalidades e serviços a computação ubíqua tenta fazer com que a atenção dos usuários seja voltada para as tarefas a serem realizadas e que os usuários não se preocupem com a operação dos dispositivos, tornando-as transparentes, ou seja, o acesso ao serviço é garantido de forma tão simples que não desvia a atenção dos utilizadores dos seus focos principais, ou seja, as ações ou tarefas.

Mesmo com a rápida evolução a aplicação da computação ubíqua em ambientes inteligentes enfrenta alguns problemas. Essas limitações são ocasionadas devido à interação entre usuário e computador, por consequência da invisibilidade das aplicações, em razão da heterogeneidade dos dispositivos, da mobilidade e diversidade de redes sem fio, de sensores,

etc., além de fatores de segurança e privacidade que causa preocupação em desenvolvedores e também nos usuários das aplicações.

A computação ubíqua pode ser aplicada em vários setores e também pode-se dividi-la em várias vertentes deixando assim uma gama de opções para o desenvolvimento de trabalhos futuros, uma das propostas de estudo futuro que este trabalho destaca é a segurança e a privacidade de aplicações ubíquas, tema muito discutido por estudiosos da área.

10. REFERÊNCIAS BIBLIOGRÁFICAS

ANNIBAL, Leandro. **Computação Ubíqua**. Universidade Federal do Rio de Janeiro, Departamento de Engenharia Eletrônica e da Computação. Disponível em: <http://www.gta.ufrj.br/grad/02_2/leandro/>. Acesso em Fev de 2010.

LOUREIRO, Antônio. **Redes de Sensores Sem Fio**. Disponível em: <<http://homepages.dcc.ufmg.br/~loureiro/cm/docs/sbr03.pdf>>. Acesso em Fev 2010.

PEROZZO, Reiner Franchesco; PEREIRA, Carlos E. **Ambientes Inteligentes: Uma Arquitetura para Cenários de Automação Predial/Residencial Baseada em Experiências**. Universidade Federal do Rio Grande do Sul, Departamento de Engenharia Elétrica. Disponível em: <<http://www6.ufrgs.br/norie/tic2007/artigos/A1127.pdf>>. Acesso em Mar 2010.

CLICK, Ambientes Inteligentes. **Ambientes Inteligentes**. Disponível em: <<http://www.clickambientes.com.br/>>. Acesso em Mar 2010.

ANJOS, André G. Rodrigues. **Aplicações Móveis Cientes de Contexto**. Disponível em: <<http://www.cin.ufpe.br/~tg/2006-1/agra-proposta.pdf>>. Acesso em Fev 2010.

COMPANY, Nissan Motors. **Carro Conceito Anti-Bêbados**. Disponível em: <<http://www.techzine.com.br/arquivo/nissan-lanca-conceito-de-carro-anti-bebados/>>. Acesso em Fev 2010.

PORTELLA, Cristiano. **Computação Ubíqua como construir interfaces invisíveis**. Disponível em: <<http://bsi.cneccapivari.br/?q=node/41>>. Acesso em: Mai 2010.

ARAUJO, Regina Borges. **Computação Ubíqua: Princípios, Tecnologias e Desafios**. Disponível em: <https://im.ufba.br/pub/MAT570FG/LivroseArtigos/045_AraujoRB.pdf>. Acesso em Mai 2010.

LOUREIRO, Antônio A. Ferreira. **Computação Ubíqua Ciente de Contexto: Desafios e Tendências**. Disponível em: <<http://www.eventoexpress.com.br/cdsbr03/pdfs/minicurso3.pdf>>. Acesso em Jun 2010.

CIRILO, Carlos Eduardo. **Computação Ubíqua: Definição, Princípios E Tecnologias**. Disponível em: <http://ufscar.academia.edu/CarlosEduardoCirilo/Teaching/14124/Computacao_Ubiqua_definicao_principios_e_tecnologias>. Acesso em Jun 2010.

WEISER, Mark. **The Computer For The 21st Century**. Disponível em: <<http://nano.xerox.com/hypertext/weiser/SciAmDraft3.html>>. Acesso em Jul 2010.

BARROS, Paulo H. Lossio. **Uma Proposta Para Sincronização De Dados Em Dispositivos Ubíquos**. Disponível em: <http://www.cpdee.ufmg.br/documentos/Defesas/790/PAULO_BARROS.pdf>. Acesso em Jul 2010.

BRAZ, Luiz G. Silva. **Framework e Middleware para Computação Ubíqua**. Pontifícia Universidade Católica Do Rio de Janeiro, Departamento de Informática. Disponível em: <<http://www-di.inf.puc-rio.br/~endler/courses/Mobile/Monografias/04/Luiz-Mono.pdf>>. Acesso em Jul 2010.

COMMONS, Creative. **Redes Sem Fio – Wireless Networks**. Disponível em: <<http://pt.kioskea.net/contents/wireless/wlintro.php3>>. Acesso em Jul 2010.

SILVA, Adailton J. S. **As tecnologias de Rede Wireless**. Rede Nacional de Ensino e pesquisa – RNP. Disponível em: <<http://www.rnp.br/newsgen/9805/wireless.html#inicio>>. Acesso em Ago 2010.

VITERBO, José F. **Descoberta de Serviços em Ambientes de Computação Ubíqua**. Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática. Disponível em: <<http://wwwdi.inf.pucRio.br/~endler/semGSD/monografias/viterboServiceDiscovery.pdf>>. Acesso em Jul 2010.

OFICINA, Net. **Segurança da Informação, Conceitos e Mecanismos**. Disponível em: <http://www.oficinadanet.com.br/artigo/1307/seguranca_da_informacao_conceitos_e_mecanismos>. Acesso em Set 2010.

PROMON, Business e Technology Review. **Segurança da Informação, Um Diferencial Determinante na Competitividade das Corporações**. Disponível em: <http://www.promon.com.br/portugues/noticias/download/Seguranca_4Web.pdf>. Acesso em Out de 2010.

MORAES, João L. C.; SOUZA, Wanderley L.; PRADO, Antonio F. **Ambiente de Computação Ubíqua para o Cuidado de Saúde Pervasivo (ACUCSP)**. Disponível em: <www.sbc.org.br/bibliotecadigital/download.php?paper=1303>. Acesso em Set 2010.

SANTOS, Ivairton Monteiro. **Computação Ubíqua**. Universidade de São Paulo – USP. Disponível em: <http://cpd1.ufmt.br/ivairton/doc/tut/computacao_ubiquaIvairtonSantos.pdf>. Acesso em Jul 2010.

ESCOBEDO, Edgardo P.; KOFUJI, Sergio Takeo. **Introdução a Ambientes Inteligentes**. Disponível em: <http://www.pad.lsi.usp.br/humanlab/trabalhos/workshop_amb_intel.pdf>. Acesso em Mai 2010.

COSTA, Cristiano A.; GEYER, Cláudio F. Resin. **Um Modelo Genérico de Infra-estrutura de Software para a Computação Ubíqua**. Disponível em: <<http://gppd.inf.ufrgs.br/wsppd/2006/downloads/artigos/costa-modelo-artigo.pdf>>. Acesso em Jul 2010.

MILAGRES, Francisco G.; SANTOS, Rodrigo F.; GOULART, Rudinei; MOREIRA, Edson S. **Segurança de Sistemas Ubíquos Baseada em Informações de Contexto**. Disponível em: <<http://www.infoguerra.com.br/misc/ubiquos.pdf>>. Acesso em Out 2010.