

Anderson Roberto Sato
Eduardo Yassuji Kido
Henrique Shoiti Fugita
Luiz Paulo Rocha Yanai
Rodrigo Seisho Hanashiro

Domínio Wireless para Aplicações Móveis

Projeto de Formatura apresentado à
Disciplina PCS 2050 – Laboratório de
Projeto de Formatura II, da Escola
Politécnica da Universidade de São Paulo

São Paulo
2005

Anderson Roberto Sato
Eduardo Yassuji Kido
Henrique Shoiti Fugita
Luiz Paulo Rocha Yanai
Rodrigo Seisho Hanashiro

Domínio Wireless para Aplicações Móveis

Projeto de Formatura apresentado à
Disciplina PCS 2050 – Laboratório de
Projeto de Formatura II, da Escola
Politécnica da Universidade de São Paulo

Área de Concentração:
Engenharia de Computação

Orientador:
Prof. Dr.
Jorge Luis Risco Becerra

São Paulo
2005

PCS

TF-2005

S83d

DEDALUS - Acervo - EPEL



31500016653

Sysno: 1520457
M 2005A 0

FICHA CATALOGRÁFICA

Sato, Anderson Roberto

Domínio wireless para aplicações móveis / A.R. Sato, E.Y.

Ki-

do, H.S. Fugita, L.P.R. Yanai, R.S. Hanashiro. – São Paulo, 2005.

93 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Comunicação móvel I.Fugita, Henrique Shoiti II.Hanashiro, Rodrigo Seisho III.Kido, Eduardo Yassuji IV.Yanai, Luiz Paulo Rocha V.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais VI.t.

*Você ainda não consegue ver,
Mas mesmo isso tudo parecendo ruim,
No futuro ainda agradecerá
Por tudo que está te acontecendo.*

Ivone Hiroko Hanashiro

Agradecimentos

Aos nossos pais, que são os principais responsáveis por todas as nossas conquistas.

Aos nossos parentes e amigos, os quais sempre nos apoiaram em momentos difíceis mesmo que tenhamos, muitas vezes, deixado de nos dedicar a eles em virtude de outros compromissos.

Aos colegas de poli, pelas lições de companheirismo, solidariedade e amizade dos últimos cinco anos.

Aos professores, que nos deram toda base necessária para atingirmos o sucesso profissional.

A todos aqueles que nos incentivaram, motivaram e ajudaram a continuarmos na busca dos nossos objetivos de vida.

Resumo

O presente trabalho tem como objetivo o desenvolvimento e a implementação de um sistema de aplicações móveis voltadas para um domínio de rede sem fio. Este domínio consiste em um conjunto de *access points* que permitem o acesso dos dispositivos móveis ao sistema. O sistema possui como base aplicações de comunicação, como mensagens instantâneas de texto e de voz, e também um serviço de localização de dispositivos, capaz de identificar em qual *access point* o dispositivo está conectado. Em cima desta base, são desenvolvidas outras aplicações para o sistema.

Este projeto engloba o estudo das seguintes tecnologias: o protocolo padrão 802.11 para redes sem fio de microcomputadores, a configuração de redes sem fio formadas por *access points*, o serviço baseado em localização (LBS) através de mensagens SNMP, o modelo de aplicações VoIP, o protocolo ponto-a-ponto SIP para implementação de transferência de voz e dados, e desenvolvimento para *PocketPC*.

Abstract

The objectives of the following study are the development and the implementation of a system created for a wireless network domain. This domain consists of a cluster of access points which allows the mobile devices to log in the system. This system is based on communication applications, such as instant text and voice messages, and a device location service, able to identify in which access point this device is connected. Over this platform, other applications can be developed for the system.

This project includes the study of the following technologies: the standard protocol 802.11 to wireless network, the configuration of access points in a wireless network, the location based services (LBS) through SNMP messages, VoIP applications model, the end-to-end protocol known as SIP to implement the transferring of voice and data, and the development for PocketPC.

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

1	INTRODUÇÃO	13
1.1	CONTEXTO GERAL.....	13
1.1.1	<i>A importância da mobilidade nas empresas</i>	13
1.2	OBJETIVO.....	14
1.3	METODOLOGIA.....	15
1.4	ORGANIZAÇÃO.....	15
2	ASPECTOS TECNOLÓGICOS E CONCEITUAIS	17
2.1	LOCATION BASED SERVICES.....	17
2.1.1	<i>Introdução</i>	17
2.1.2	<i>Aplicações</i>	18
2.2	ADDRESS RESOLUTION PROTOCOL (ARP).....	19
2.2.1	<i>Introdução</i>	19
2.2.2	<i>Endereço Físico</i>	19
2.2.3	<i>Resolução de endereço</i>	21
2.3	WIRELESS LAN.....	22
2.3.1	<i>Introdução</i>	22
2.3.2	<i>Aplicações</i>	23
2.3.3	<i>Benefícios</i>	23
2.3.4	<i>Topologia</i>	24
2.3.5	<i>Segurança</i>	25
2.4	SESSION INITIATION PROTOCOL (SIP).....	28
2.4.1	<i>Introdução</i>	28
2.4.2	<i>Rede SIP</i>	29
2.5	CONVERGÊNCIA.....	31
2.5.1	<i>Introdução</i>	31
2.5.2	<i>Convergência de redes e de aparelhos</i>	32
2.5.3	<i>Contexto</i>	33
2.5.4	<i>Benefícios</i>	33
2.6	VOIP.....	35
2.6.1	<i>Introdução</i>	35
2.6.2	<i>Arquitetura</i>	36
2.6.3	<i>Benefícios</i>	37
2.6.4	<i>Tendências</i>	38
2.7	WEB SERVICES.....	39
2.7.1	<i>Introdução</i>	39
2.7.2	<i>SOAP e Web Services</i>	40
2.7.3	<i>Documento WSDL</i>	41
2.8	CRÈME.....	43
2.8.1	<i>Introdução</i>	43
2.8.2	<i>Vantagens</i>	43
2.9	JAVA API FOR INTEGRATED NETWORKS (JAIN).....	44
2.9.1	<i>Introdução</i>	44
2.9.2	<i>Objetivos</i>	44
2.10	JAVA MEDIA FRAMEWORK API (JMF).....	45

2.10.1	<i>Introdução</i>	45
2.10.2	<i>Vantagens</i>	45
3	ESPECIFICAÇÃO DO PROJETO	46
3.1	ESCOPO DA SOLUÇÃO	46
3.1.1	<i>Contexto</i>	46
3.2	REQUISITOS FUNCIONAIS	47
3.3	REQUISITOS NÃO-FUNCIONAIS	48
3.4	CARACTERÍSTICAS DOS USUÁRIOS	49
3.5	RESTRICÇÕES	50
3.6	MODELO DE CASOS DE USO	51
3.6.1	<i>Descrição dos Atores</i>	51
3.6.2	<i>Descrição dos Casos de Uso</i>	51
3.7	MODELO DE CLASSES	55
3.8	MODELO DA INTERFACE HOMEM-MÁQUINA	58
3.9	ARQUITETURA DO SISTEMA	59
4	PROJETO DE IMPLEMENTAÇÃO	60
4.1	METODOLOGIA DE PROJETO	60
4.2	IMPLEMENTAÇÃO	61
4.2.1	<i>Infra-estrutura de Hardware</i>	61
4.2.2	<i>Arquitetura de Software</i>	63
4.2.3	<i>Aplicativo Cliente</i>	66
4.2.4	<i>SIP Proxy Server</i>	70
4.2.5	<i>Web Services</i>	71
4.2.6	<i>Serviço de Autenticação</i>	72
4.2.7	<i>Serviço de Localização</i>	73
4.2.8	<i>Serviço de Arquivos</i>	77
4.2.9	<i>Módulo de Cadastro Web</i>	78
4.2.10	<i>Serviço de Persistência</i>	82
4.2.11	<i>Banco de Dados</i>	84
4.3	TESTES E RESULTADOS	84
4.3.1	<i>Envio de mensagens instantâneas</i>	85
4.3.2	<i>Realização de chamadas de voz</i>	85
4.3.3	<i>Serviço de localização</i>	86
4.3.4	<i>Disponibilidade de arquivos</i>	86
5	CONSIDERAÇÕES FINAIS	87
5.1	CONCLUSÃO DO PROJETO	87
5.2	TRABALHOS FUTUROS	88

ANEXO A – DESCRIÇÃO DOS CASOS DE USO

LISTA DE TABELAS

TABELA 1: Componentes de uma WLAN.....	24
TABELA 2: Principais Características do SIP.....	28

LISTA DE FIGURAS

FIGURA 1: Representação do endereço MAC.....	20
FIGURA 2: Representação do endereçamento MAC.....	20
FIGURA 3: Troca de mensagens ARP.....	21
FIGURA 4: Acesso a rede sem fio.....	22
FIGURA 5: Benefícios das WLANs.....	24
FIGURA 6: Arquitetura geral de uma WLAN.....	25
FIGURA 7: Estabelecimento de uma sessão SIP através de um servidor de proxy.....	29
FIGURA 8: Estabelecimento de uma sessão SIP através de um servidor de redirecionamento.....	30
FIGURA 9: Exemplo de convergência de aparelhos.....	32
FIGURA 10: Voz empacotada em formato de dado.....	36
FIGURA 11: Representação da arquitetura em camadas do VoIP.....	36
FIGURA 12: Projeção do tráfego de VoIP em relação à voz tradicional.....	40
FIGURA 13: Representação de uma mensagem SOAP.....	41
FIGURA 14: Modelo de classes da aplicação cliente.....	55
FIGURA 15: Modelo de classes da aplicação servidor.....	56
FIGURA 16: Arquitetura física e funcional do DWAM.....	59
FIGURA 17: Diagrama da infra-estrutura de hardware do DWAM.....	61
FIGURA 18: Diagrama da arquitetura de software do DWAM.....	64
FIGURA 19: Interface principal do DWAM Client.....	67
FIGURA 20: Mapa gerado dinamicamente com posição do usuário.....	69
FIGURA 21: Interface do JAIN SIP Proxy.....	70
FIGURA 22: Lista dos Endereços MAC conectados.....	75
FIGURA 23: Regulagem da potência de transmissão.....	75
FIGURA 24: Diagrama do tratamento das requisições no servidor de localização.....	77
FIGURA 25: Diagrama de Funcionamento do Struts Framework.....	79
FIGURA 26: Página web de Gerenciamento de Usuários no sistema DWAM.....	82

LISTA DE ABREVIATURAS E SIGLAS

AAA: Authentication, Autorization, Accounting

AP: Access Point

API: Application Programming Interface

ARP: Address Resolution Protocol

Cell Id: Cell Identification

DWAM: sistema Domínio Wireless para Aplicações Móveis

GPS: Global Positioning System

HTTP: HyperText Transfer Protocol

IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Forces (estabele padrões para Internet)

IHC: Interface Homem-Computador

IHM: Interface Homem-Máquina

IM: Instant Messenger

IP: Internet Protocol

JAIN: Java API for Integrated Networks

JDBC: Java Data Base Connection

JMF: Java Media Framework API

JSP: Java Server Pages

J2ME: Java 2 Micro Edition

LBS: Location-Based Services

MAC: Media Access Control

PDA: Personal Digital Assitant

PSTN: Public Switched Telephone Network

QoS: Quality of Service

SGBD: Sistema de Gerenciamento de Banco de Dados

SIP: Session Initiation Protocol

SNMP: Simple Network Management Protocol

SOAP: Simple Object Access Protocol

TCP/IP: Transmission Control Protocol / Internet Protocol

TOA: Time of arrival

UA: User Agent

UAC: User Agent Client

UAS: User Agent Server

VoIP: Voice over IP

XML: Extensible Markup Language

WEP: Wireless Equivalent Privacy

Wi-Fi: Wireless Fidelity ou simplesmente Wireless

WLAN: Wireless Local Area Network

WSDL: Web Services Description Language

1 INTRODUÇÃO

Este trabalho apresenta os principais motivos que nos levaram a desenvolver e a estudar aplicações para redes sem fio, bem como os resultados de nossos estudos e esforços.

1.1 Contexto Geral

O sistema foi projetado e desenvolvido imaginando-se as futuras necessidades das corporações de um modo geral. O escopo do projeto teve alguns motivadores importantes que estão apresentados a seguir.

1.1.1 A importância da mobilidade nas empresas

Segundo Robert Kiyosaki, “há trezentos anos, a terra era riqueza, de modo que quem a possuísse, possuía riqueza. Então, apareceram as fábricas e a produção. Os industriais eram os donos da riqueza. Hoje é a informação. E a pessoa que tiver a informação na hora certa, terá a riqueza. E esse é o grande desafio nos dias de hoje, ter acesso a essas informações de qualquer lugar do planeta.”

Quando viajam, muitos profissionais precisam acessar informações na Internet ou em máquinas remotas. Tendo em vista que é impossível ter uma conexão por fios em meios de transporte (como carros e aviões), surgiu um grande interesse em redes sem fio.

Com as redes sem fio tornou-se possível o surgimento dos escritórios portáteis. Assim, as pessoas passaram a usar seus equipamentos eletrônicos para enviar e receber ligações telefônicas, fax e correio eletrônico, navegar pela *Web*, acessar arquivos remotos e se conectar a máquinas distantes. Além disso, nas conferências de

informática de hoje, muitas vezes é configurada uma rede sem fio para que os participantes possam, através de um *notebook*, ter acesso aos materiais.

Outra importância das redes sem fio é seu baixo custo de instalação. Para empresas sediadas em edifícios antigos, onde não possuem cabamentos e fiações, pode ser mais econômico a implementação de uma rede sem fio. [2]

1.2 Objetivo

Com o notável crescimento e desenvolvimento das redes sem fio, novas aplicações e tecnologias surgem a cada dia. As conveniências e facilidades promovidas pela mobilidade têm sido muitas vezes decisivas para o sucesso dos negócios das empresas. Mas não são somente as companhias que se interessam por essas facilidades. Muitos estudantes, professores e até mesmo usuários domésticos também apreciam muito a conveniência e organização que aparelhos sem fios podem prover. Uma prova disso é o grande crescimento nas vendas de *notebooks* e PDA's.

Além disso, tem-se notado que muitas vezes o tempo das aulas é sub-aproveitado. Seja com as dúvidas dos alunos sobre a disponibilização dos materiais utilizados ou mesmo com os professores que, eventualmente, têm problemas com seus materiais digitais. Então, com o objetivo de otimizar o tempo de aula, modernizar as classes e aperfeiçoar o sistema de controle dos professores resolvemos desenvolver uma aplicação para esse segmento utilizando redes sem fio.

Assim, nosso trabalho visa projetar e criar um sistema para disponibilização de arquivos digitais de acordo com a localização do usuário e o horário pretendido.

Um diferencial do projeto será a implementação de aplicações que podem ser muito úteis como a localização física de usuários na rede e a comunicação por mensagens e por voz.

1.3 Metodologia

Este trabalho foi escrito seguindo a ordem cronológica das etapas de desenvolvimento do projeto. Assim, a ordem dos capítulos está estruturada de forma que, o primeiro exhibe as primeiras definições e o último, os resultados. Mas, antes da apresentação do projeto em si, existem sessões prévias na qual são registradas as motivações para desenvolver esse projeto, as tecnologias utilizadas e outras informações importantes para que o leitor possa compreender melhor como foi desenvolvido o sistema DWAM.

1.4 Organização

O presente trabalho está organizado em 5 capítulos e 1 anexo. Esta organização visa seguir o padrão definido pelo Serviço de Bibliotecas da Escola Politécnica da USP no documento “Diretrizes para apresentação de dissertações e teses” editado pelo serviço de bibliotecas da Escola Politécnica de São Paulo em 2001.

O primeiro capítulo, a Introdução, apresenta o projeto explicando os objetivos, o escopo e o fator motivador para sua realização. Nesse capítulo também é abordado o contexto no qual se insere o sistema DWAM e como ele pode contribuir para o dia-a-dia das pessoas.

O segundo capítulo, Aspectos Tecnológicos e Conceituais, apresenta as informações técnicas importantes para o entendimento do projeto. Nele são abordados e explicados

as tecnologias, os conceitos, os protocolos e os métodos utilizados para o desenvolvimento do sistema DWAM.

Já o terceiro capítulo, Especificação do Projeto, com as informações importantes já apresentadas, passa a detalhar e especificar o sistema. São apresentados todos seus requisitos, suas características e suas limitações.

O capítulo de número quatro, Projeto de Implementação, mostra o ambiente de implantação e a metodologia utilizada.

Por fim, o capítulo da conclusão exhibe os resultados e discute alternativas de projetos de melhoria.

2 ASPECTOS TECNOLÓGICOS E CONCEITUAIS

O objetivo deste capítulo é apresentar as tecnologias utilizadas na realização do projeto, assim como alguns aspectos conceituais importantes.

2.1 Location Based Services

Nos dias de hoje, com a forte concorrência entre as empresas de telecomunicações, as operadoras de redes móveis travam uma busca constante por novidades tecnológicas que possam agregar valor para o dia-a-dia das pessoas. Um dos serviços que mais tem sido explorado nessa ocasião é o de localização. Assim apresentaremos a seguir alguns dos conceitos importantes.

2.1.1 Introdução

Os métodos de localização podem ser divididos em três grupos básicos: *Cell Id (Cell Identification)*, *TOA (Time of Arrival)* e *GPS (Global Positioning System)*.

O *Cell Id* localiza o usuário de acordo com o AP onde ele está conectado. Assim, sabendo o alcance das ondas podemos saber sua área de localização. O TOA baseia-se no tempo de propagação do sinal de rádio transmitido. Assim, quando se deseja localizar um usuário, envia-se o sinal e mede-se o tempo. Dessa forma é possível calcular a posição do usuário. A técnica do GPS é bastante semelhante ao TOA, porém os terminais recebem sinais provenientes de pelo menos três satélites. Depois, baseando-se nos tempos de propagação dos sinais, calcula-se a posição do terminal.

No sistema DWAM o método utilizado é *Cell Id*, pois ele já atende as premissas do projeto e possui uma maior simplicidade. [3]

2.1.2 Aplicações

Apesar de o LBS ser um sistema apenas para localização, ele nos gera informações que podem nos servir de diversas formas:

- **Conveniência:** o usuário pode receber informações de restaurantes, farmácias, hospitais e outros pontos de conveniências próximos à sua localização. Além disso, ele pode se informar sobre condições de trânsito e se orientar caso esteja perdido.
- **Segurança:** com as freqüentes ondas de violência, a segurança tem sido uma preocupação muito grande das famílias. Com o sistema de localização, casos como seqüestros e furtos poderão ser mais rapidamente resolvidos.
- **Vigilância:** os transportes de valores poderão ser vigiados e monitorados de maneira mais eficaz.
- **Emergência:** vítimas de acidentes poderiam receber socorro mais rapidamente.
- **Aplicações corporativas:** possibilitaria às empresas ter um melhor gerenciamento de sua força móvel de trabalho. [3]

2.2 Address Resolution Protocol (ARP)

O sistema DWAM utiliza, em seu módulo de localização, o protocolo ARP. Assim as principais características desse protocolo são apresentadas a seguir para possibilitar uma melhor compreensão do sistema.

2.2.1 Introdução

Para que haja comunicação entre dois nós de uma rede, não basta conhecer apenas seu endereço lógico (endereço IP), é preciso também conhecer seu endereço físico (MAC address). O ARP é um protocolo usado pelo protocolo Internet (IP) para mapear os endereços lógicos dos dispositivos de uma rede aos seus respectivos endereços físicos.

O termo “resolução de endereço”, tradução de *address resolution*, se refere ao processo de encontrar um endereço de um computador em uma rede. O endereço é “resolvido” através de uma requisição de um processo cliente a um processo servidor. A informação recebida permite ao servidor identificar a rede à qual o dispositivo procurado pertence. Assim, a resolução do endereço é completada quando o cliente recebe a resposta do servidor com o endereço requisitado. [4]

2.2.2 Endereço Físico

Também conhecido como endereço MAC, o endereço físico é dado por quarenta e oito bits. Esses bits são divididos em duas partes de vinte e quatro sendo que a primeira identifica o fabricante e a segunda é uma seqüência dada por esse fabricante.

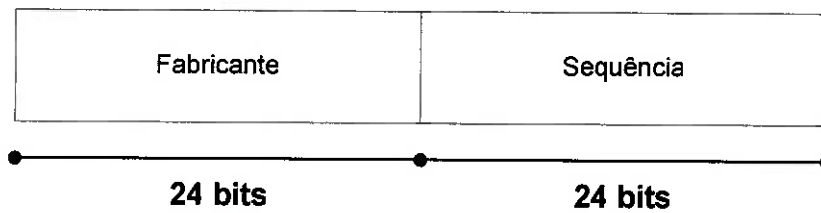


FIGURA 1: Representação do endereço MAC [4]

Por questões de facilidade, esses endereços são expressos por uma seqüência de doze números hexadecimais como mostra a figura 2 abaixo.

```

Command Prompt
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Wireless Network Connection:

Media State . . . . . : Media disconnected
Description . . . . . : Intel(R) PRO/Wireless 2200BG Network
Connection
Physical Address. . . . . : 00-0E-35-BF-94-35

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . : 
Description . . . . . : Realtek RTL8139 C+ Fast Ethernet NIC

Physical Address. . . . . : 00-0F-B0-45-2F-37
Dhcp Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IP Address. . . . . : 200.207.209.244
Subnet Mask . . . . . : 255.255.255.192
Default Gateway . . . . . : 200.207.209.193
DHCP Server . . . . . : 200.207.209.193
DNS Servers . . . . . : 200.204.0.10
                          200.204.0.138

```

FIGURA 2: Representação do endereçamento MAC

Uma característica importante dos endereços MAC é que eles são embutidos no *hardware* das placas de rede e são únicos. Assim, é impossível que dois equipamentos de rede possuam o mesmo endereçamento físico.

2.2.3 Resolução de endereço

A resolução de endereço provido pelo ARP é executada através do envio de uma mensagem de broadcast na rede. Através da troca de mensagens é possível mapear o endereço MAC do dispositivo a partir do seu endereço IP. Assim, conhecido o endereço lógico do destino, a máquina origem envia uma mensagem em broadcast do tipo: “quem possui o endereço IP XXX. XXX. XXX. XXX, por favor, me informe seu endereço MAC”. Então a máquina procurada retorna a mensagem com os dados requisitados. A figura 3 abaixo, ilustra essa troca de mensagens. [5]

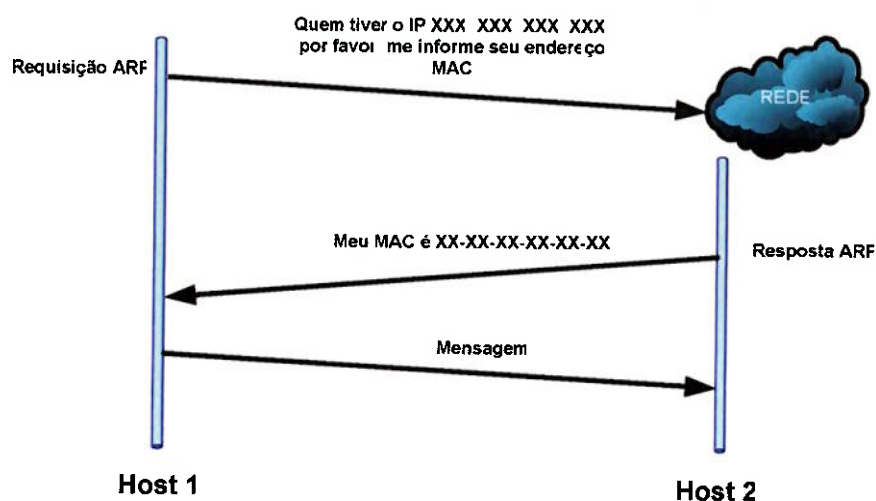


FIGURA 3: Troca de mensagens ARP [5]

No sistema DWAM, o *ARP Request* é utilizado para mapear o endereço MAC de um usuário que esteja conectado no sistema. Essa informação será utilizada no servidor de localização como uma chave de procura para determinar a localização do usuário.

2.3 Wireless LAN

O sistema DWAM é composto por uma rede WLAN, logo é de extrema importância o conhecimento dos principais conceitos desse tipo de rede. Sendo assim, são introduzidos a seguir alguns aspectos da arquitetura e protocolos de comunicação das WLANs, que são baseados no padrão IEEE 802.11.

2.3.1 Introdução

A WLAN é uma rede local sem fio implementada como extensão ou alternativa para redes convencionais. Nessa rede, os equipamentos utilizam sinais de rádio frequência para se comunicarem uns com os outros. Desta forma, uma WLAN oferece aos seus usuários comunicação de dados e mobilidade. A figura 4 abaixo ilustra alguns equipamentos conectados à Internet através de um acesso sem fio.

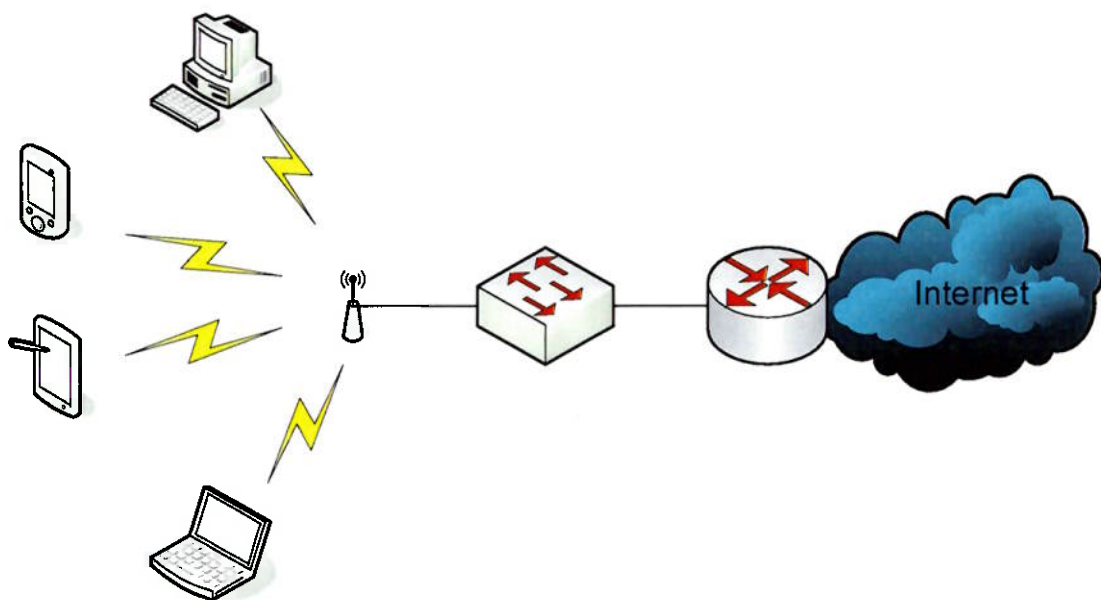


FIGURA 4: Acesso a rede sem fio

Padronizadas pelo IEEE 802.11, essas redes possibilitam transmissões de altas taxas de dados (até 11 Mbit/s) a distância de algumas dezenas de metros (distância nominal de 100 metros). [2]

Serão apresentadas mais detalhadamente a arquitetura e protocolos da rede WLAN IEEE 802.11. Este padrão define as especificações das camadas físicas e de acesso ao meio (MAC).

2.3.2 Aplicações

As WLANs têm sido usadas em campi de instituições de ensino, prédios comerciais, aeroportos, condomínios residenciais, transações comerciais e bancárias. Além disso, as WLANs também são empregadas onde a estrutura de cabeamento é inexistente, como por exemplo prédios antigos.

2.3.3 Benefícios

Os benefícios proporcionados pelas redes sem fio não são restritos à questão da mobilidade. Outras características que tornam as WLANs atraentes são sua rapidez de instalação e seu baixo custo.

Para montar uma rede tradicional, é necessária infra-estrutura de cabeamentos, bem como equipamentos de difícil manuseio. Já as WLANs necessitam apenas de APs. Sua configuração é simples e rápida.

A figura 5 resume as vantagens oferecidas pelas redes sem fio.

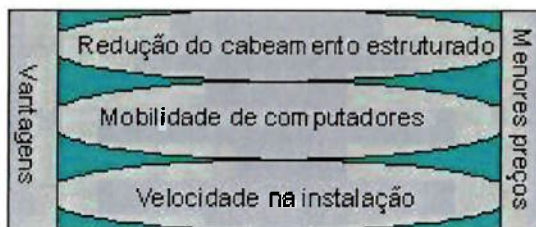


FIGURA 5: Benefícios das WLANs [6]

Assim, muitas empresas e até mesmo pessoas físicas montam sua própria rede sem fio sem dificuldades. [2]

2.3.4 Topologia

A topologia geral de uma WLAN 802.11 é composta pelos seguintes elementos, listados na tabela 1:

BSS - Basic Service Set	Corresponde a uma célula de comunicação da rede sem fio.
STA - Wireless LAN Stations	São os diversos clientes da rede.
AP - Access Point	É o nó que coordena a comunicação entre as STAs dentro da BSS. Funciona como uma ponte de comunicação entre a rede sem fio e a rede convencional.
DS - Distribution System	Corresponde ao backbone da WLAN, realizando a comunicação entre os APs.
ESS - Extended Service Set	Conjunto de células BSS cujos APs estão conectados a uma mesma rede convencional. Nestas condições uma STA pode se movimentar de uma célula BSS para outra permanecendo conectada à rede. Este processo é denominado de Roaming.

TABELA 1: Componentes de uma WLAN [7]

A figura 6 ilustra um exemplo de uma arquitetura com seus elementos.

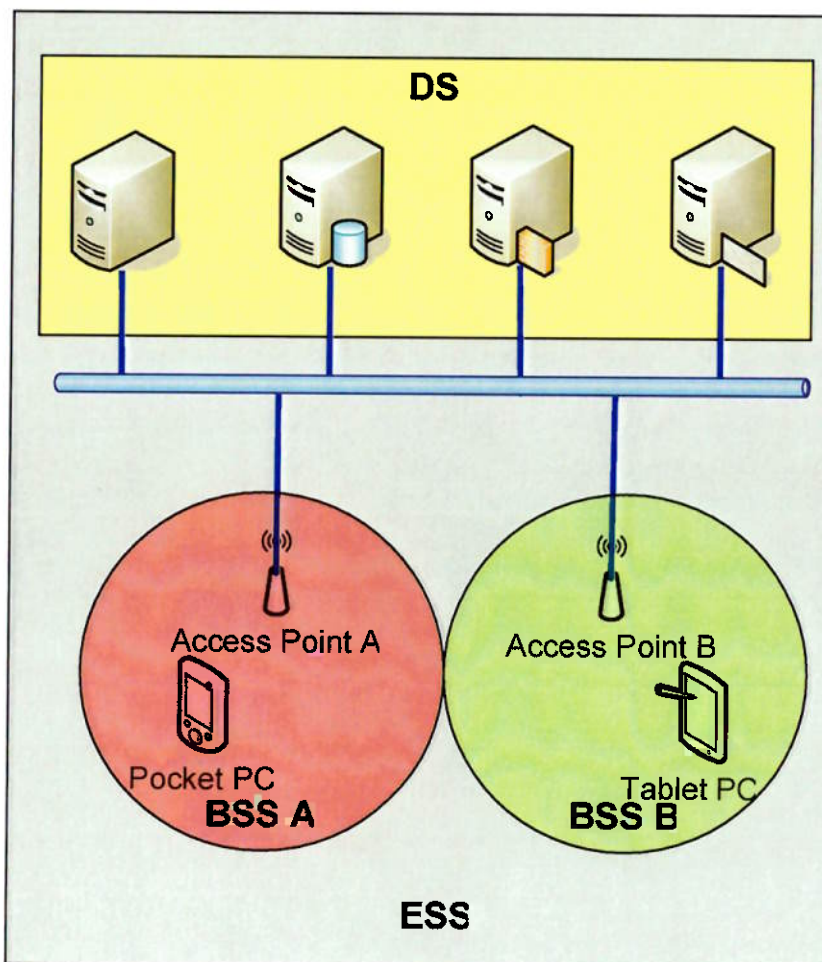


FIGURA 6: Arquitetura geral de uma WLAN

Nessa arquitetura, os STA's são representados pelo Pocket PC e pelo Tablet PC.

2.3.5 Segurança

Durante o início das redes de computadores, a segurança não era uma preocupação. Porém, com a evolução tecnológica, aplicações de caráter sigiloso, como transações

bancárias, começaram a trafegar pela rede. Assim, para garantir que pessoas mal-intencionadas não leiam nem modifiquem secretamente dados enviados a outros destinatários, começaram a surgir mecanismos de segurança.

A maior parte dos problemas de segurança é causada por pessoas mal-intencionadas que tentam obter algum benefício, chamar a atenção ou prejudicar alguém. Porém esses ataques normalmente são feitos por pessoas inteligentes e bem instruídas, o que acarreta em complicados desafios de segurança. [2]

Serão apresentados dois métodos de serviço de segurança que o padrão IEEE 802.11 oferece:

- Autenticação;
- Criptografia.

2.3.5.1 Autenticação

Autenticação é o serviço que verifica se uma estação está autorizada a se comunicar com outra estação em uma dada área de cobertura. Na infra-estrutura do sistema, a autenticação é feita entre um AP (*Access Point*) e cada estação. Esse serviço será baseado em endereçamento físico (*MAC Address*) e a autorização é dada confirmando-se o nome e a senha do usuário. [2]

Assim, todos os equipamentos utilizados deverão ter seu *MAC address* cadastrados no sistema. Se confirmado, o sistema solicitará ao usuário um nome e uma senha. Se todos os dados estiverem de acordo com a base de dados, o usuário terá permissão de acessar o sistema.

2.3.5.2 Criptografia

A criptografia consiste em codificar os dados à medida que saem de um equipamento e decodificá-los assim que os pacotes entram em outro sistema. Porém, só podem decodificar a informação os equipamentos que possuem a “chave”. Assim, as informações podem trafegar pela rede seguramente, pois mesmo que elas sejam interceptadas, o intruso terá dificuldades para decifrá-la. [2]

O método utilizado no sistema DWAM é conhecido como WEP (*Wireless Equivalent Privacy*) e se destina especialmente às redes sem fio.

2.4 Session Initiation Protocol (SIP)

Para as aplicações IP oferecidas pelo DWAM, o protocolo utilizado foi o SIP. Portanto, para entender de maneira aprofundada a implementação do sistema é essencial entender e compreender esse protocolo. Assim, segue um descritivo com as características mais importantes.

2.4.1 Introdução

O protocolo SIP foi padronizado pelo IETF com o objetivo de oferecer uma forma mais simples e modular de comunicação sobre IP. Utilizando o método de requisição/resposta ele estabelece comunicação entre vários componentes de uma rede e, ultimamente, devido ao VoIP, estabelece também uma sessão telefônica entre dois ou mais usuários finais. Pelo fato do modelo de voz sobre IP do IETF ser altamente modular, ele é flexível e pode ser adaptado com facilidade a novas aplicações. Um resumo das características do SIP é apresentado na tabela 2: [8]

Item	SIP
Projetado por	IETF
Compatibilidade com PSTN	Ampla
Compatibilidade com a Internet	Sim
Arquitetura	Modular
Abrangência	O SIP lida apenas com a configuração
Sinalização de chamadas	SIP sobre TCP ou UDP
Formato de mensagens	ASCII
Transporte de mídia	RTP/RTCP
Chamadas de vários participantes	Sim
Conferências de multimídia	Não
Endereçamento	URL
Transmissão de mensagens instantâneas	Sim
Criptografia	Sim
Tamanho do documento de padrões	128 páginas
Implementação	Moderada
Suporte para Firewall	Fácil
Status	Ganhando presença no mercado

TABELA 2: Principais Características do SIP

2.4.2 Rede SIP

Uma rede SIP é composta pelos seguintes componentes:

Servidor de Proxy: funciona como um equipamento intermediário que recebe as requisições SIP do cliente e a encaminha para a rota adequada. O servidor de proxy pode fornecer serviços como autenticação, autorização, controle de acesso à rede, roteamento e segurança [8]. A figura 7 ilustra esse modo:

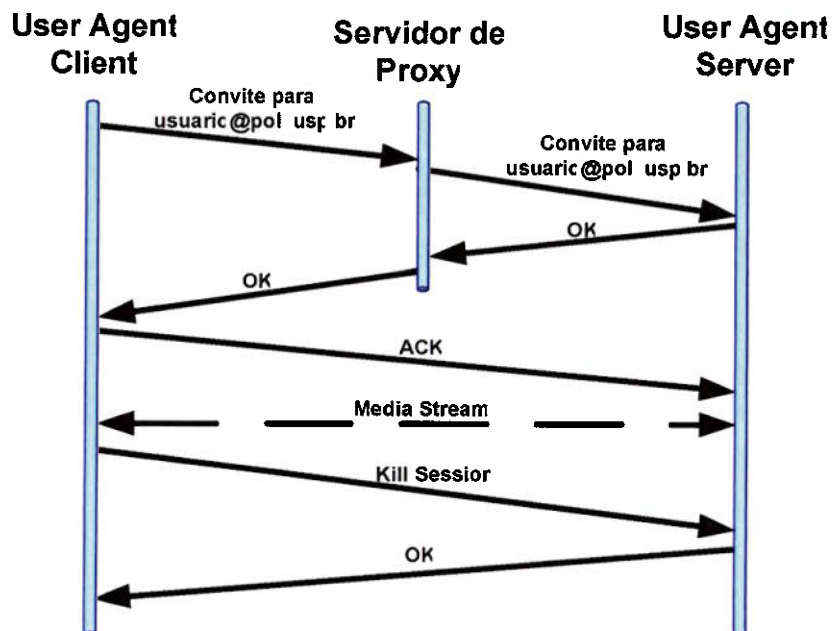
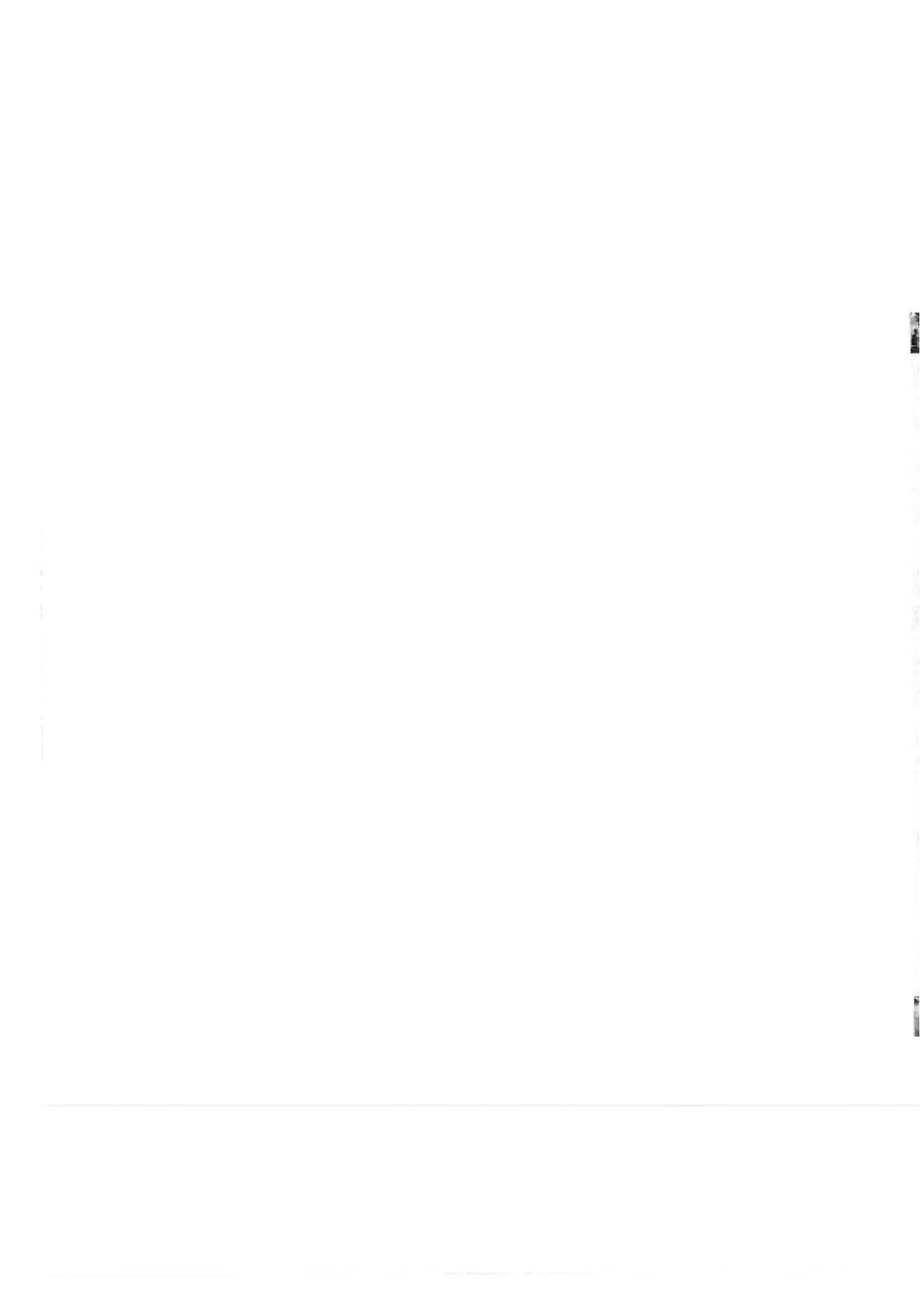


FIGURA 7: Estabelecimento de uma sessão SIP através de um servidor de proxy [9]

Servidor de redirecionamento: fornece ao cliente informações sobre os próximos saltos dos pacotes. Isso faz com que o cliente possa se conectar a esses equipamentos diretamente [8]. A figura 8 ilustra esse modo:



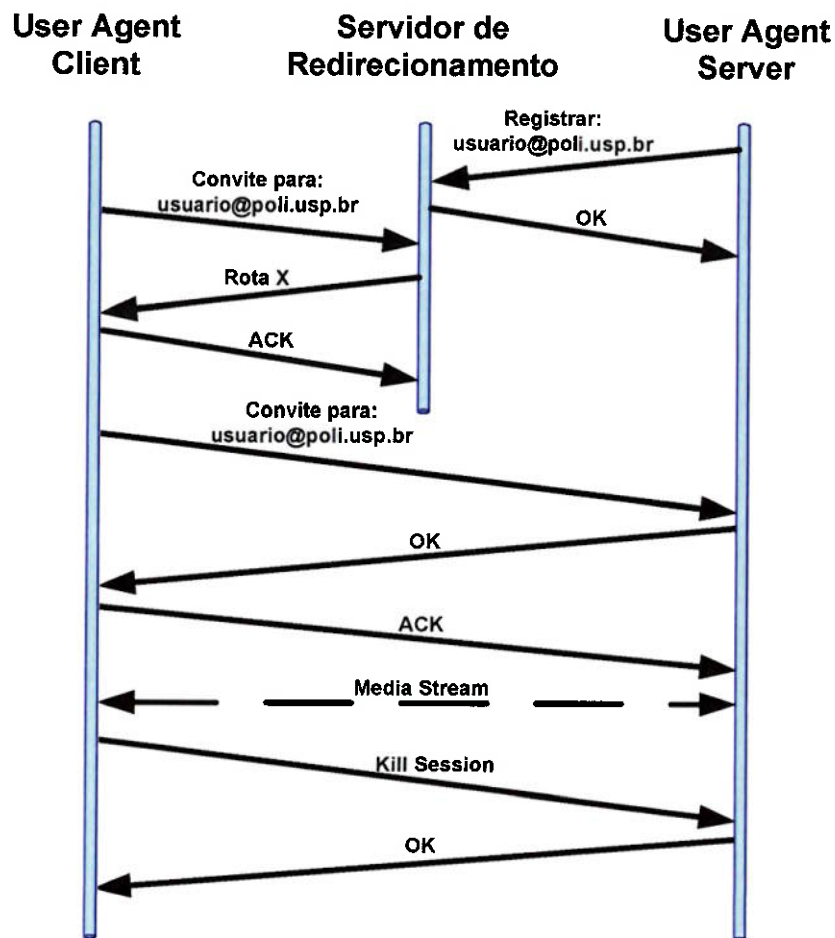


FIGURA 8: Estabelecimento de uma sessão SIP através de um servidor de redirecionamento [9]

Servidor de registro: os servidores de registro processam as requisições dos usuários e atualizam as informações dos *user agents* no servidor de localização. Normalmente os servidores de proxy e de redirecionamento possuem um servidor de registro [8].

User Agent (UA): um user agent client (UAC) faz a requisição SIP. O user agent server (UAS) é um servidor de aplicação que comunica o usuário quando este recebe uma requisição SIP e retorna uma resposta [8].

2.5 Convergência

O termo “convergência” tem sido bastante discutido nos últimos anos. O sistema DWAM, também utiliza a convergência em seu projeto. Portanto, segue uma explicação sobre esse importante conceito e suas principais características.

2.5.1 Introdução

Desde o surgimento da primeira rede de comunicação, com o telégrafo em 1830, era criada uma nova infra-estrutura de rede para cada nova mídia que surgia. Assim, com os sucessivos surgimentos do telefone, Internet e TV a cabo, surgiam também diferentes tipos de rede, cada uma oferecendo diferentes serviços. Dessa forma, hoje em dia é comum as pessoas possuírem em suas residências conexões distintas para se ter acesso a essa variedade de serviços. Quando falamos sobre convergência, nos referimos à unificação desses serviços em uma única infra-estrutura de transporte.

Apesar de ser um bastante utilizado para os meios de transporte, o conceito de convergência também pode ser utilizado para aparelhos. Quando os telefones celulares começaram a se popularizar, eles eram grandes, caros e seus serviços, bastante limitados. Porém, com o passar o tempo, a tecnologia permitiu que não só o tamanho dos aparelhos diminuísse, mas também que o custo e a qualidade do serviço melhorassem. Então, para poderem se manter no mercado, as empresas fornecedoras de aparelhos celulares precisavam agregar novos valores aos seus produtos. E não demorou para que surgissem os aparelhos com acesso à Internet, câmeras e MP3 *player*. [10]

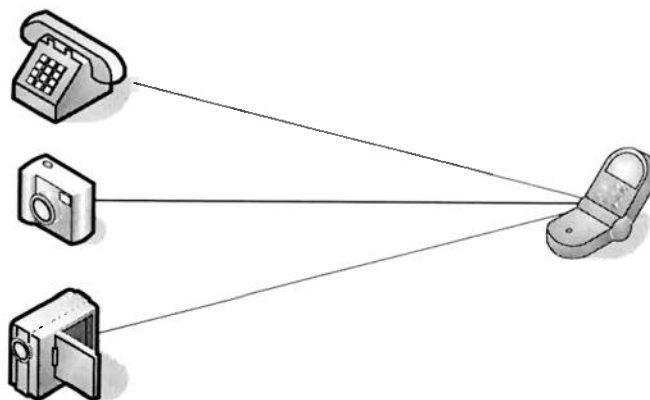


FIGURA 9: Exemplo de convergência de aparelhos

2.5.2 Convergência de redes e de aparelhos

A convergência de redes envolve, além da unificação dos diferentes tipos de redes, projetos de QoS. Assim, não se trata apenas de “empacotar” voz e vídeo em formato de dados e enviá-los pela rede. Deve se garantir a qualidade de serviço e controlar o tráfego de acordo com a prioridade dos pacotes. Devido à limitação no tempo de execução do projeto, o DWAM englobou uma convergência bastante simplificada, visto que não existe controle de QoS.

O foco do DWAM é, certamente, a convergência de aparelhos. Assim, devido ao aumento nas vendas de *Pockets PCs* nos últimos anos e a pouca quantidade de ferramentas e aplicativos existentes para esse mercado, foram projetados serviços que possam contribuir para a maior comodidade dos usuários.

2.5.3 Contexto

Nas redes de telefonia tradicionais os protocolos que trafegam são proprietários. Isso quer dizer que os diferentes fornecedores não conseguem se comunicar entre si. Essa pouca interoperabilidade entre os diferentes fabricantes tem limitado a oferta de serviços. A tentativa de se resolver esse problema tem sido a padronização de protocolos. Esse fato gera a possibilidade de se integrar às redes composta por elementos de diferentes fabricantes fazendo com que os serviços possam atingir um público maior.

Apesar dessa maior facilidade de se criar aplicativos e oferecer serviços, a plataforma dos PDAs ainda é bastante limitada. Não somente pela pouca oferta de ferramentas, mas também pela falta de documentação a respeito desse ambiente.

2.5.4 Benefícios

A convergência nos proporciona muitos benefícios como:

- Interoperabilidade: com a padronização dos protocolos de comunicação, equipamentos de diversos fabricantes podem se comunicar entre si;
- Facilidade de se criar aplicações: APIs de código aberto e tecnologias que suportam diferentes plataformas, como o Java, permitem a criação de aplicações em um curto espaço de tempo e a baixo custo;
- Baixo custo de gerenciamento: unificando-se as redes, torna-se muito mais fácil realizar o gerenciamento, pois ao invés de duas ou mais infra-estruturas de transporte, passa-se a ter apenas uma. Além disso, o investimento com a capacitação dos funcionários também é reduzido visto que se diminui a quantidade de tecnologias a serem dominadas;

- Baixo custo de manutenção: a unificação das infra-estruturas e a interoperabilidade fazem com que os custos de manutenção diminuam consideravelmente. As companhias telefônicas não ficam tão dependentes das empresas fornecedoras, diminuindo assim a necessidade de se contratar equipes de especialistas para cada fornecedor, como acontece nos dias de hoje.

2.6 VoIP

O VoIP é uma tecnologia de comunicação que vem ganhando cada vez mais adeptos tanto entre usuários domésticos quanto corporativos. Dessa forma, embasado no ganho de popularidade de aplicações VoIP, o sistema DWAM aborda esse tema entendendo ser uma clara tendência de mercado. [11]

Para se entender melhor essa tecnologia, segue abaixo uma explicação das suas características.

2.6.1 Introdução

Com a evolução tecnológica e a popularidade da Internet, programas que permitem a chamada *free voice* entre computadores (como *Google Talk* e *Skype*) passaram a ser bastante utilizados tanto no âmbito residencial quanto comercial. Isso porque o custo das ligações por intermédio desses programas é muito menor. Assim, a tecnologia VoIP se tornou uma tendência de mercado bastante clara e uma ameaça às companhias telefônicas. É por esse motivo que o sistema DWAM, com uma visão futura de negócio, aborda o VoIP.

Mas, afinal, o que é o VoIP? Nada mais é do que uma tecnologia que permite a realização de sessões de comunicação por voz entre dois equipamentos IP. Podem ser eles computadores, PDAs, tablets ou até mesmo os modernos telefones IP.



FIGURA 10: Voz empacotada em formato de dado

É nesse contexto de migração das tecnologias de telefonia tradicional para a telefonia IP que foi abordado o VoIP. [11]

2.6.2 Arquitetura

Dividida em camadas, a arquitetura do VoIP se apresenta bem mais flexível se comparada à da rede de telefonia tradicional (também conhecida como PSTN). Nessa arquitetura, os limites das funcionalidades são bem definidos, mas a interação entre elas é bastante forte. Segue a representação das camadas:

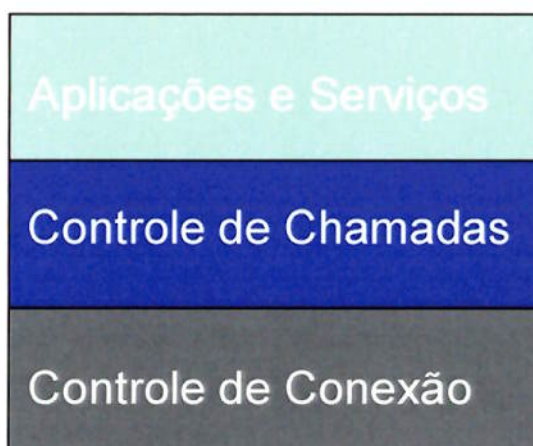


FIGURA 11: Representação da arquitetura em camadas do VoIP

- **Controle de Conexão:** essa camada, de mais baixo nível, é responsável pela digitalização, codificação/decodificação e empacotamento dos sinais de voz. O equipamento que faz essa “tradução de sinais” ou a “conversão de protocolos” entre a PSTN e a rede TCP/IP é conhecido como *Media Gateway*.
- **Controle de Chamadas:** responsável por estabelecer e manter as chamadas. Essa camada congrega grande parte da inteligência da arquitetura. O *SoftSwitch*, elemento que representa essa camada, tem a função de controlar as chamadas e supervisionar o *Media Gateway* por meio de instruções de AAA e tradução de números telefônicos tradicionais para endereços IP.
- **Aplicações e Serviços:** a camada mais alta da arquitetura fornece aplicações como chamada em espera e áudio-conferência. Como ela é conectada ao *SoftSwitch* por meio de APIs e protocolos abertos de sinalização, existe a possibilidade de se criar aplicativos em um curto espaço de tempo. [11]

2.6.3 Benefícios

Os benefícios oferecidos pelo novo modelo de telefonia são [11]:

- **Baixos custos:** os *SoftSwitches* já conseguem oferecer a mesma capacidade de processamento de chamadas do que a oferecida pelas centrais telefônicas tradicionais porém, a um custo mais reduzido.
- **Facilidade de gerenciamento:** atualmente são mantidas e administradas duas redes separadas – rede de dados e rede de voz. Com a voz sendo transmitida pela rede TCP/IP, há uma convergência e assim, os custos de manutenção e administração são reduzidos.

- Oferecimento de novos serviços: devido à convergência de redes e ao fato de a plataforma de aplicações ser aberta, existe uma facilidade muito grande no que se refere a aperfeiçoamento dos serviços existentes e criação de outros. [6]

2.6.4 Tendências

Considerando o aumento no tráfego de voz empacotada desde 2001, o aumento na oferta de serviços e a opção de algumas empresas pelo VoIP, podemos dizer que estamos vivenciando uma mudança no panorama da telefonia. Assim, apesar de a telefonia analógica tradicional tardar a se extinguir, seu volume de tráfego tende a ser reduzido e substituído pelo VoIP, conforme mostra o gráfico. [11]

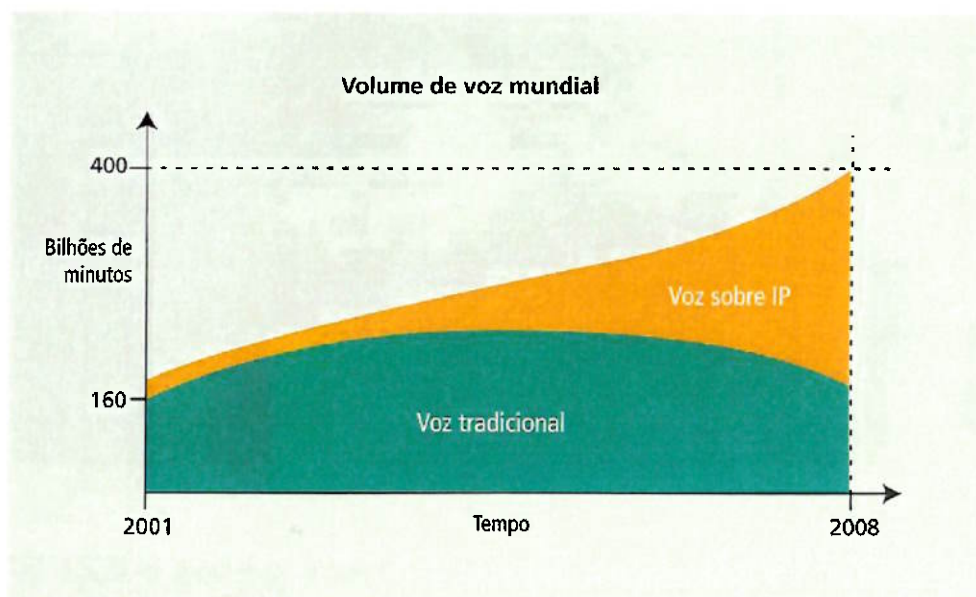


FIGURA 12: Projeção do tráfego de VoIP em relação à voz tradicional [11]

2.7 Web Services

A arquitetura do sistema DWAM baseia-se na comunicação entre diferentes aplicações. Esses serviços foram implementados utilizando-se os Web Services. Portanto, seguem os principais fundamentos e conceitos desse tema para auxiliar na compreensão da arquitetura proposta.

2.7.1 Introdução

Com a popularização da Internet e o surgimento freqüente de novas tecnologias e *frameworks* de desenvolvimento, muitos serviços e aplicativos começaram a surgir. Porém devido à utilização de diferentes ferramentas, tecnologias e plataformas em seu desenvolvimento, muitos desses produtos que eram lançados não conseguiam interagir com outros. [12] Assim, surgiu a necessidade de se desenvolver alguma arquitetura que pudesse resolver esse problema. Nesse contexto, surgem os *Web Services* que, analisando em alto nível, nada mais são do que serviços e aplicativos distribuídos através de uma rede. Esses serviços podem ser acessados e utilizados, enviando seus resultados instantaneamente. De um ponto de vista mais conceitual, os *Web Services* são um conjunto de padrões e protocolos que nos permitem processar as requisições de sistemas remotos utilizando linguagens e protocolos de transporte comuns. [10]

Para ilustrar uma aplicação real de *Web Services* podemos citar o *Internet Banking*, serviço comumente oferecido pelas instituições financeiras. As operações que os usuários realizam, como pagamentos, transferências e investimentos, devem ser registrados, conferidos e validados no sistema central: checa se realmente existe crédito na conta onde se efetuará o débito, transfere a quantia estipulada de uma conta para outra, etc. Os *Web Services* são esses serviços que são acessados para efetuar essas transações e verificações com o sistema central. [12]

2.7.2 SOAP e Web Services

O SOAP é um protocolo de comunicação baseado em XML e definido pelo consórcio W3C (*World Wide Web Consortium*). Projetado para comunicação via Internet, esse protocolo invoca aplicações remotas através de trocas de mensagens. Devido ao seu ambiente independente de plataforma e de linguagem, o SOAP pode garantir a interoperabilidade e intercomunicação entre diferentes sistemas. Por esses motivos, esse protocolo é um padrão normalmente aceito para se utilizar juntamente com *Web Services*. [12]

Em suma, as principais características do SOAP são [12]:

- Definido pelo consórcio W3C;
- Protocolo baseado em XML para troca de informações em um ambiente distribuído;
- Padrão de utilização com *Web Services*;
- Normalmente utiliza o HTTP como protocolo de transporte.

Uma mensagem SOAP é formada por 3 (três) partes básicas [12]:

- *Envelope*: campo obrigatório, ele é o elemento raiz do documento XML. O *Envelope* contém informações como declarações de *namespaces* e o *encoding style*, que define como os dados são representados no documento XML.
- *Header*: é o cabeçalho da mensagem. Ele carrega informações adicionais, como por exemplo, se a mensagem deve ser processada por algum nó intermediário da rede. Apesar de esse campo ser opcional, quando utilizado, o *Header* deve ser o primeiro elemento do *Envelope*;
- *Body*: parte principal da mensagem, esse elemento contém a informação a ser transportada ao destino final (*payload*). Existe também um outro campo, opcional, que armazena dados de status e erros gerados nos nós durante o processamento da mensagem.

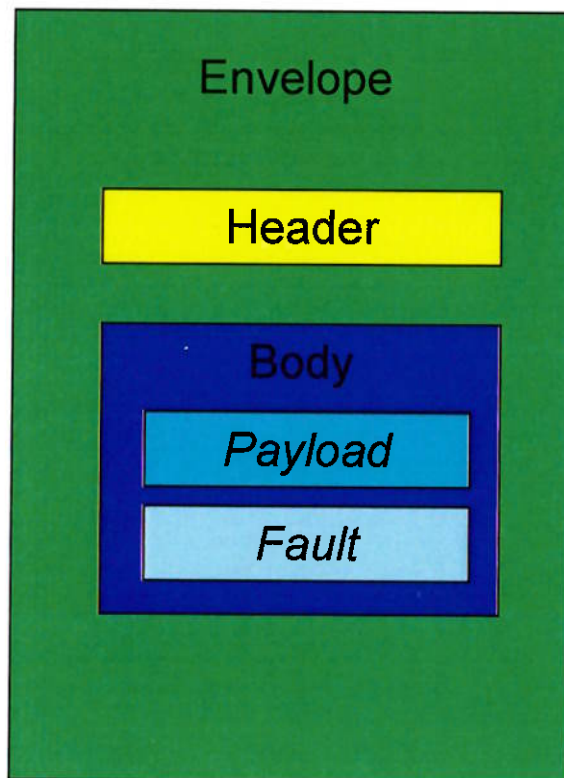


FIGURA 13: Representação de uma mensagem SOAP [12]

2.7.3 Documento WSDL

Devido ao fato de os *Web Services* serem baseados em XML, um protocolo padrão, sua aceitação é bastante ampla. Mais especificamente, esse serviços são descritos através de uma linguagem chamada WSDL (*Web Service Description Language*). Essa linguagem define todas as interfaces, operações, esquemas de codificação e as demais informações de um *Web Service*.

Um documento WSDL define um XML *Schema* para descrever um *Web Service*. Assim que o cliente tiver acesso à descrição do serviço a ser utilizado, a implementação do *Web Service* pode ser feita utilizando qualquer linguagem de programação.

Usualmente, são utilizadas linguagens com forte interação com a *Web* como, por exemplo, ASP, Java *Servlets* e PHP.

Basicamente, quando o cliente deseja consumir algum *Web Service*, ele localiza o documento WSDL desse serviço e obtém sua descrição. Em seguida, a mensagem é construída e, de acordo com a descrição do WSDL, os dados necessários são enviados. O servidor onde o serviço se encontra recebe a requisição. O *Web Service*, assim que receber a mensagem, faz a validação junto ao WSDL. “A partir de então, o serviço remoto sabe como tratar a mensagem, como processá-la (possivelmente enviando-a para outro cliente) e como enviar a resposta ao cliente.” [12]

2.8 CrEme

Mesmo com as altas tecnologias computacionais que encontramos hoje em dia, as aplicações para PDAs ainda possuem um campo muito grande para ser explorado. Nesse contexto entra a contribuição do CrEme, explicado a seguir.

2.8.1 Introdução

Com a mobilidade cada vez mais presente na vida das pessoas, surge a necessidade de desenvolver produtos para esse setor. Assim, a Microsoft oferece o Windows CE, seu sistema operacional para equipamentos móveis. Mas isso, por si só não era suficiente, pois a quantidade de aplicações compatíveis com o segmento era muito limitada. Para o desenvolvimento de aplicações Java, torna-se necessário a implementação de máquinas virtuais para os dispositivos móveis. Nesse contexto, surge o CrEme, uma Máquina Virtual Java para Windows CE. [13]

2.8.2 Vantagens

A utilização do CrEme no projeto se deu devido às suas seguintes vantagens [13]:

- Fácil instalação: apesar de sua rápida expansão, o acesso aos equipamentos móveis ainda é privilégio de poucos. Assim, é importante facilitar o manuseio desses equipamentos e seus serviços.
- Suporte ao Internet Explorer: necessário visto que os usuários podem se conectar à Internet.
- Forte integração com o ambiente do Windows CE.

2.9 Java API for Integrated Networks (JAIN)

Apesar de não ser nenhuma tecnologia propriamente dita, o JAIN apresenta idéias, sugestões e ferramentas que foram de grande utilidade para o desenvolvimento do DWAM. Portanto, segue uma breve apresentação dessa comunidade. Possivelmente essa união de grupos poderá indicar caminhos para a divulgação de importantes informações.

2.9.1 Introdução

O JAIN é uma comunidade de mais de 80 membros que especifica e cria APIs para IP e telefonia que inclui o desenvolvimento de aplicações de alto nível e também de baixo nível como APIs de sinalização (SIP). [14]

2.9.2 Objetivos

A comunidade JAIN visa transformar o mercado de telecomunicações que tem sistemas fechados de grandes proprietários para uma arquitetura aberta de rápida integração.

O programa JAIN tem três grandes objetivos [14]:

- **Portabilidade de Serviços:** pelo fornecimento de APIs padrões.
- **Convergência de Redes:** Fornecendo APIs abstratas de modo que aplicações possam rodar não importando a tecnologia da rede.
- **Service Provider Access -** Permitindo que aplicações rodem fora do domínio da rede de segurança do operador.

2.10 Java Media Framework API (JMF)

O JMF, sendo uma API de grande importância na realização desse trabalho, também será explicado em um capítulo a parte. Dessa forma, segue uma breve explicação sobre as características mais importantes do JMF.

2.10.1 Introdução

O JMF é uma API que possibilita que mídias de áudio e vídeo sejam adicionadas a aplicações programadas sobre a plataforma Java. Nesse “pacote” existem métodos que podem auxiliar na captura e exibição de *streaming* de diversos tipos de mídias. [15] Além disso, essa API suporta sincronização, controle e processamento de tais mídias. [14]

2.10.2 Vantagens

Desenvolvido pela Sun, Silicon Graphics e Intel, o JMF possui algumas vantagens que o torna bastante atraente. [16] Muitas dessas vantagens são provenientes do fato do JMF ser uma API não proprietária. Assim, podem ser citadas [15]:

- **Interoperabilidade:** uma característica bastante marcante dos códigos abertos é sua interoperabilidade. Devido à tendência de convergência, essa capacidade está sendo cada vez mais valorizada.
- **Custo:** a utilização de APIs reduzem os custos de projeto tanto em relação ao tempo de desenvolvimento quanto aos custos com licenças.
- **Manuseio de mídias:** por conseguir trabalhar com mídias de áudio de diversos formatos (MPEG-1, MPEG-2, AVI, WAV, AU e MIDI) o torna mais flexível ao programador pois existem menos restrições ao seu manuseio.

3 Especificação do Projeto

3.1 Escopo da Solução

O sistema Domínio Wireless para Aplicações Móveis (DWAM) visa facilitar a comunicação e a interação entre pessoas pertencentes a um mesmo contexto (acadêmico, corporativo ou comercial) através de aplicações que fazem uso de redes sem fio e de recursos de localização nessas redes.

3.1.1 Contexto

Para a implantação do sistema DWAM, utilizou-se o ambiente acadêmico. Assim, todas as aplicações e exemplos dados na documentação estão voltados especificamente para esse contexto. Porém vale a pena lembrar que o sistema não foi desenvolvido exclusivamente para este fim.

3.2 Requisitos Funcionais

O sistema DWAM consiste em um domínio de rede sem fio, no qual os usuários podem utilizar aplicações, como comunicação por mensagens de texto e voz, compartilhamento de arquivos e localização física de usuários.

Os usuários podem se conectar ao domínio através de um dispositivo móvel (*PocketPC*, *Notebook* ou *TabletPC*) com acesso a redes sem fio. Por motivos de segurança, para entrar no sistema o usuário deverá fazer sua autenticação através de um usuário e uma senha.

O domínio de rede será formado por um conjunto de *access points* (APs). Cada AP define uma localidade no domínio, ou seja, um subdomínio de rede. Desta forma, a localização de usuários será implementada de modo a localizar em qual AP o dispositivo do usuário está conectado, indicando assim em qual localidade o usuário se encontra.

No sistema será possível configurar os serviços e arquivos que serão disponibilizados de acordo com a localidade e o horário. Assim, se o professor quiser disponibilizar seu material digital apenas para sua sala durante o horário da sua aula, ele poderá fazê-lo. Além disso, cada usuário terá uma lista de contatos e poderá interagir apenas com os usuários dessa lista. Essa interação poderá dar-se por meio de envio de mensagens instantâneas ou por chamadas de voz. Logicamente, um professor terá a opção de não disponibilizar esse serviço durante as aulas.

O recurso de localização também estará disponível somente para encontrar os usuários de sua lista de contatos.

3.3 Requisitos Não-Funcionais

Os requisitos não-funcionais do sistema são os seguintes:

- **Transparência:** os diferentes tipos de usuários do sistema não precisam ter conhecimento da estrutura computacional envolvida. Eles só precisam passar as informações necessárias que o DWAM realiza as tarefas designadas.
- **Modularidade:** o sistema oferece diversos tipos de serviços. Assim, para facilitar a implantação e a manutenibilidade e proporcionar maior escalabilidade, esses serviços possuem uma estrutura modular.
- **Escalabilidade:** fácil devido à modularidade e ao uso da tecnologia Java.
- **Manutenibilidade:** garantida pela modularidade, pois não há necessidade de indisponibilizar o sistema inteiro para realizar algum reparo.
- **Segurança de acesso (*security*):** com a finalidade de fazer um controle de acesso e gerenciar os usuários, login e senha serão requisitados. Além disso, a criptografia dos dados irá assegurar aos usuários confidencialidade durante seu acesso ao sistema.

3.4 Características dos Usuários

No sistema DWAM, existem três tipos de usuário: administrador do sistema, administrador de localidades e usuário comum.

O administrador do sistema é o responsável pelo gerenciamento dos equipamentos centrais como o servidor de aplicações, banco de dados e o *Hub/Switch*.

O administrador de localidades, no nosso contexto, seriam os professores e palestrantes. Eles possuem acesso à configuração dos APs e, a partir dele, disponibilizam os serviços e arquivos desejados.

O usuário comum são todos os alunos, funcionários e convidados que possuem um login e senha para se conectar ao sistema. Esse tipo de usuário não possui permissão para alterar nenhuma configuração do sistema, estando sujeito a utilizar apenas os serviços disponibilizados pelos administradores.

Apesar da necessidade de todos eles terem conhecimentos básicos de informática, existem características específicas para cada um desses tipos de usuários.

Tanto o administrador do sistema quanto o administrador de localidade devem conhecer a estrutura do DWAM e saber operá-lo. Não há necessidade de um conhecimento profundo de informática devido ao uso de padrões e orientações de usabilidade. Já o usuário comum deve possuir apenas conhecimentos básicos de informática, saber operar seu equipamento móvel e interagir com ele.

3.5 Restrições

Para utilizar os serviços oferecidos pelo DWAM, os administradores (do sistema e de localidade) devem possuir um computador conectado ao sistema. Para poder configurá-lo conforme o desejado, os servidores do sistema deverão ter instalado o servidor Web Jakarta Tomcat e o banco de dados PostGreSQL.

O usuário comum deverá possuir seu próprio equipamento móvel, ou seja, ele deverá possuir um *notebook*, *tablet* ou um PDA com acesso a redes sem fio. Além do hardware, o usuário comum também deverá possuir instalado em seu PDA a máquina virtual Java e o aplicativo cliente do DWAM.

3.6 Modelo de Casos de Uso

3.6.1 Descrição dos Atores

Os atores dos casos de uso são os usuários do sistema, ou seja, o administrador do sistema, o administrador de localidade e o usuário comum.

3.6.2 Descrição dos Casos de Uso

Segue uma breve descrição dos casos de uso do sistema:

Caso de uso 1: Autenticação de Usuário no Sistema

Descrição: Este caso de uso descreve a autenticação de um usuário no sistema.

Caso de uso 2: Manipular Usuário

Caso de uso 2.1: Cadastro de Usuário

Descrição: Este caso de uso descreve o cadastro de um usuário no sistema.

Caso de uso 2.2: Remoção de Usuário

Descrição: Este caso de uso descreve a remoção de um usuário do sistema.

Caso de uso 2.3: Alteração de Usuário

Descrição: Este caso de uso descreve a alteração das informações de um usuário do sistema.

Caso de uso 2.4: Busca de Usuário

Descrição: Este caso de uso descreve a busca de um usuário no sistema.

Caso de uso 3: Manipular Grupo de Usuário**Caso de uso 3.1: Cadastro de Grupo de Usuários**

Descrição: Este caso de uso descreve o cadastro de um grupo de usuário no sistema.

Caso de uso 3.2: Remoção de Grupo de Usuário

Descrição: Este caso de uso descreve a remoção de um grupo de usuário do sistema.

Caso de uso 3.3: Alteração de Grupo de Usuário

Descrição: Este caso de uso descreve a alteração das informações de um grupo de usuário do sistema.

Caso de uso 3.4: Busca de Grupo de Usuários

Descrição: Este caso de uso descreve a busca de um grupo de usuários no sistema.

Caso de uso 4: Manipular Localidade**Caso de uso 4.1: Cadastro de Localidade**

Descrição: Este caso de uso descreve o cadastro de uma localidade no sistema.

Caso de uso 4.2: Remoção da localidade

Descrição: Este caso de uso descreve a remoção de uma localidade do sistema.

Caso de uso 4.3: Alteração de Localidade

Descrição: Este caso de uso descreve a alteração das informações de uma localidade do sistema.

Caso de uso 4.4: Busca de localidade.

Descrição: Este caso de uso descreve a busca de uma localidade no sistema.

Caso de uso 5: Enviar Mensagem Instantânea

Descrição: Este caso de uso descreve o processo de envio de mensagens instantâneas entre dois usuários, estabelecendo uma conversa.

Caso de uso 6: Fazer Chamada de Voz

Descrição: Este caso de uso descreve o processo de fazer uma chamada de voz para outro usuário do sistema.

Caso de uso 7: Receber Chamada de Voz

Descrição: Este caso de uso descreve a recepção de uma chamada de voz e o estabelecimento de uma conversa de voz.

Caso de uso 8: Disponibilizar Documento

Descrição: Este caso de uso descreve o processo de disponibilização de documentos para outros usuários do sistema.

Caso de uso 9: Download de Documento

Descrição: Este caso de uso descreve o processo de *download* de documentos disponíveis em uma localidade.

Caso de uso 10: Manipular Lista de Contatos**Caso de uso 10.1: Adicionar Usuário à Lista de Contatos**

Descrição: Este caso de uso descreve a adição de outro usuário do sistema à lista de contatos de um usuário.

Caso de uso 10.2: Autorizar Adição à Lista de Contatos

Descrição: Este caso de uso descreve o processo de um usuário autorizar outro a adicioná-lo à sua lista de contato.

Caso de uso 10.3: Remover Usuário da Lista de Contatos

Descrição: Este caso de uso descreve o processo de remoção de um usuário da lista de contato.

Caso de uso 11: Localizar Usuário da Lista de Usuário

Descrição: Este caso de uso descreve a busca (física) de um usuário no sistema, identificando sua localidade.

A descrição detalhada dos casos de uso está apresentada no anexo A.

3.7 Modelo de Classes

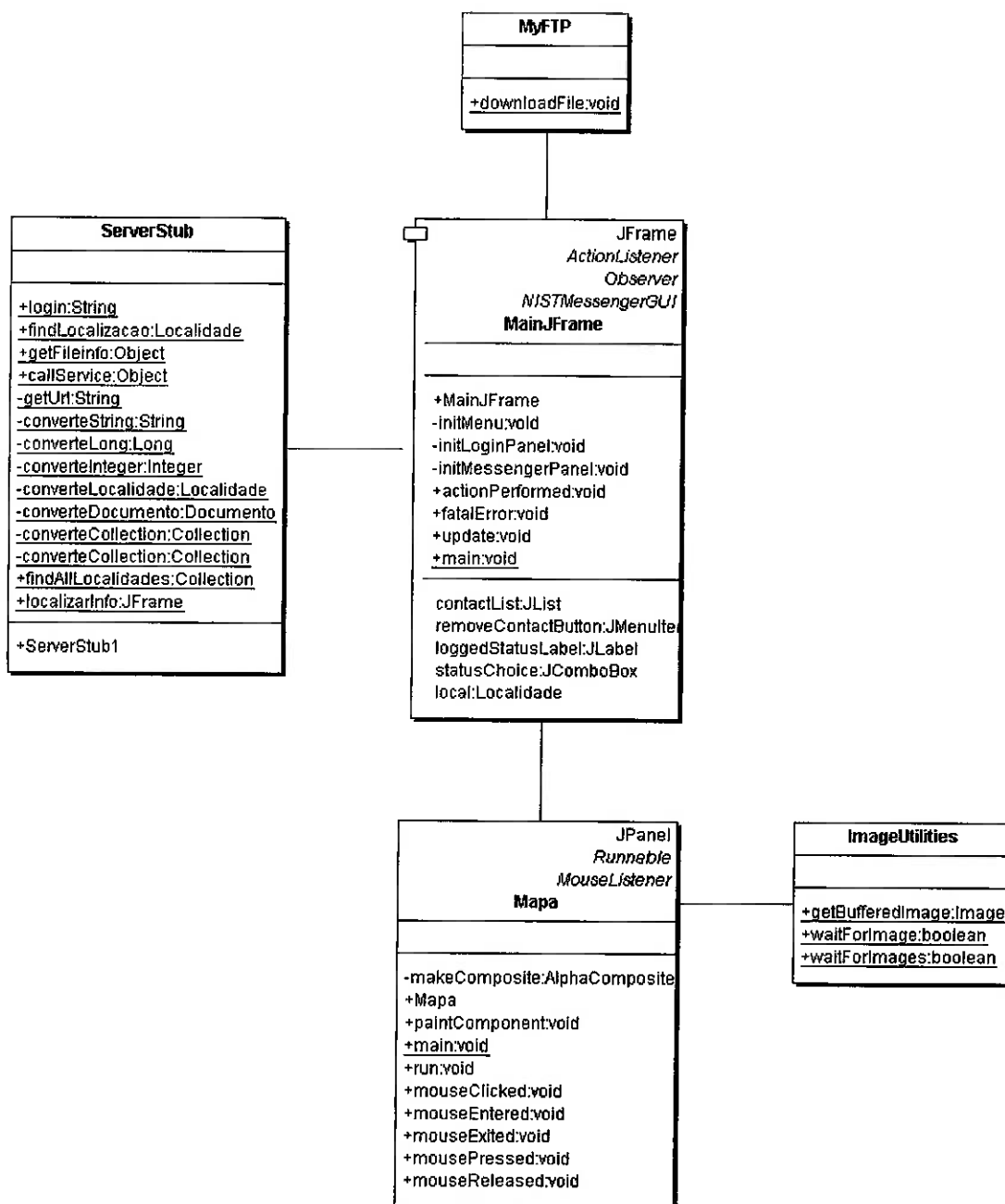


FIGURA 14: Modelo de classes da aplicação cliente

Segue uma breve descrição das classes pertencentes ao cliente do sistema DWAM:

MainJFrame: Classe principal que exibe a tela principal do aplicativo com as abas para cada serviço disponível no sistema.

Mapa: Classe responsável pela renderização do mapa que informa a localização dos usuários no domínio.

ImageUtilities: Classe que é responsável por carregar as imagens das localidades onde o sistema foi implantado.

MyFTP: Classe responsável por fazer o download de arquivos de uma localidade para o dispositivo do usuário.

ServerStub: Classe que encapsula as chamadas aos webservices que oferecem as informações de localidades, arquivos e autenticação para o usuário.

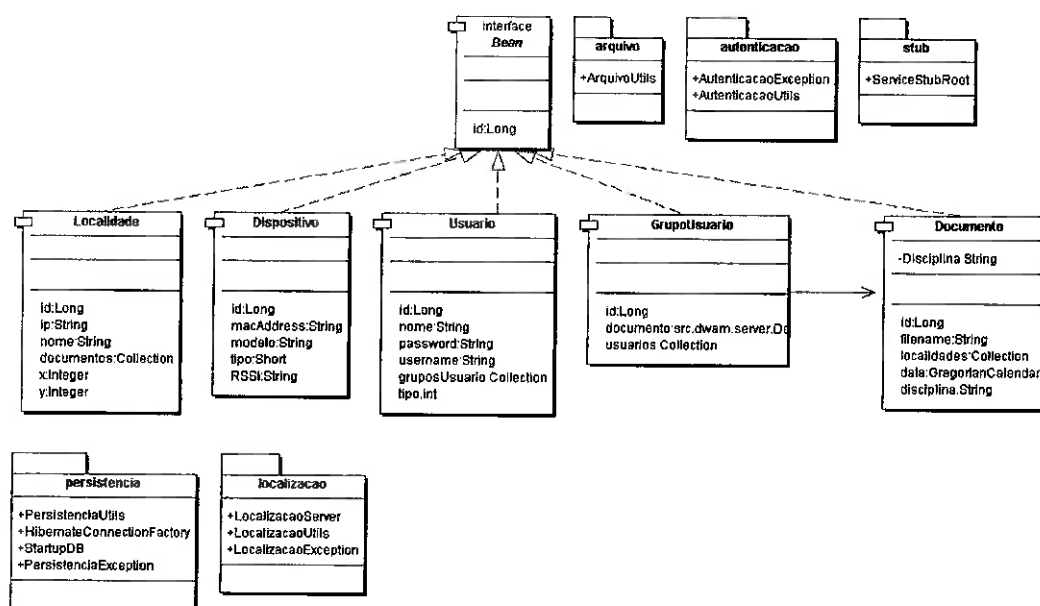


FIGURA 15: Modelo de classes da aplicação servidor

Segue abaixo a descrição das classes que fazem parte do lado servidor do sistema

DWAM:

Bean: Interface para as classes que serão persistidas no sistema.

Localidade: Bean que guarda informações de localidades do sistema.

Dispositivo: Classe persistente que representa um dispositivo móvel, pertencente a um usuário. É necessário cadastrar os dispositivos para implementar segurança e localização por endereço MAC.

Usuario: Classe persistente que representará os usuários do sistema. Cada usuário possuirá uma senha, que será utilizada na autenticação.

GrupoUsuario: Classe que reúne informações de grupos de usuários do sistema.

Documento: Bean que guarda informações dos documentos que são disponibilizados pelo sistema.

AutenticacaoUtils: Classe que possuirá os métodos e algoritmos de autenticação de usuários.

LocalizacaoUtils: Classe que implementará o algoritmo de localização de usuários.

PersistenciaUtils: Classe que possuirá os métodos de persistência e acesso ao banco de dados.

ArquivoUtils: Classe que implementará as relações de negócio referente a disponibilização de arquivos.

3.8 Modelo da Interface Homem-Máquina

Devido aos diferentes tipos de usuários, mais de um modelo de interface foi projetado. Porém, para facilitar a administração e gerenciamento do sistema, todas as interfaces possuem características em comum: são bastante intuitivas e até mesmo familiares para os usuários de computador.

Os administradores (do sistema e de localidade) acessam o sistema via *Web*. Toda a programação deverá ser realizada através de um *browser* interativo. Assim, o administrador deverá apenas inserir as informações requeridas que o sistema se encarrega de executar o programado.

Para o usuário comum, as aplicações de comunicação (mensagem e voz) e localização serão acessadas através de uma lista de usuários. Essa lista terá formato similar aos serviços oferecidos pelo *MSN*, *Yahoo Messenger*, *Skype* e *Google Talk*. Devido ao grande número de adeptos a esses programas nos ambientes corporativos, acredita-se que essa familiaridade auxiliará esses usuários no que se refere à utilização do DWAM.

3.9 Arquitetura do Sistema

Para facilitar o entendimento do sistema DWAM, a figura 16 abaixo ilustra a arquitetura física e funcional do sistema.

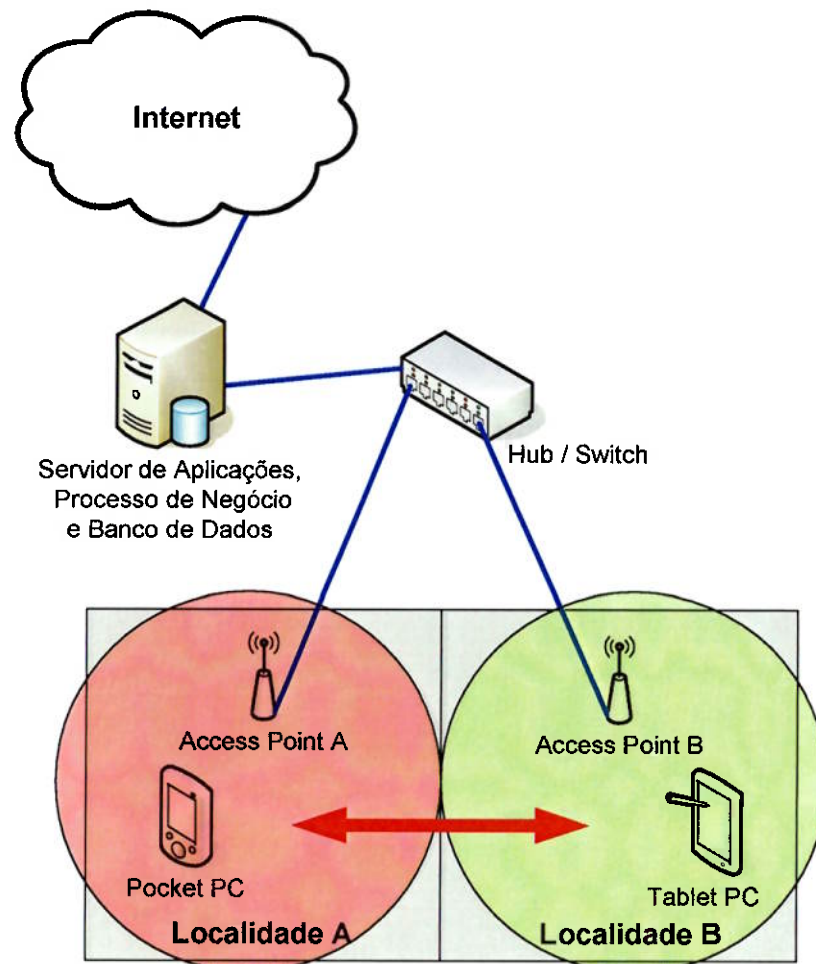


FIGURA 16: Arquitetura física e funcional do DWAM

4 Projeto de Implementação

O objetivo desse capítulo é apresentar o sistema DWAM como produto final. Assim, serão apresentados todos os seus módulos separadamente, além dos testes e resultados obtidos.

4.1 Metodologia de Projeto

A metodologia de projeto adotada para o desenvolvimento do sistema DWAM foi o processo em cascata, que é composto pelas etapas consecutivas de produção de software: levantamento de requisitos, especificação, análise, implementação e testes.

A fase de levantamento de requisitos correspondeu ao período de definição do tema do trabalho de formatura e ao fechamento de seu escopo. Foram determinadas as funcionalidades que o sistema deveria ter e o domínio de aplicação do mesmo.

Na fase de especificação, foram elaborados os modelos preliminares de casos de uso, de classes e de interação, além da formalização dos requisitos não-funcionais do sistema. Em seguida, foi realizada a etapa de análise, em que, a partir dos casos de uso, foi elaborada a solução, com a arquitetura de software do sistema e seus módulos constituintes. As funcionalidades especificadas pelo modelo de casos de uso foram agrupadas em módulos e as tecnologias a serem utilizadas em cada um deles foram selecionadas e estudadas.

Na fase de implementação, cada integrante do grupo ficou responsável pelo desenvolvimento de um ou mais módulos, que foram posteriormente integrados entre si. Nesta fase também foi definida a infra-estrutura de hardware necessária e realizada a implantação do sistema na mesma.

Por fim, o sistema foi testado para verificar seu funcionamento e atendimento aos requisitos.

4.2 Implementação

4.2.1 Infra-estrutura de Hardware

Para possibilitar o funcionamento do sistema DWAM, foi projetada uma infra-estrutura de hardware composta por uma rede sem fio formada por diversos *access points* e um conjunto de servidores interligados em rede. O esquema abaixo representa a infra-estrutura de rede exigida pelo DWAM e os dispositivos móveis suportados. Junto a cada equipamento representado é informado, entre parênteses, respectivamente, os softwares que devem estar instalados.

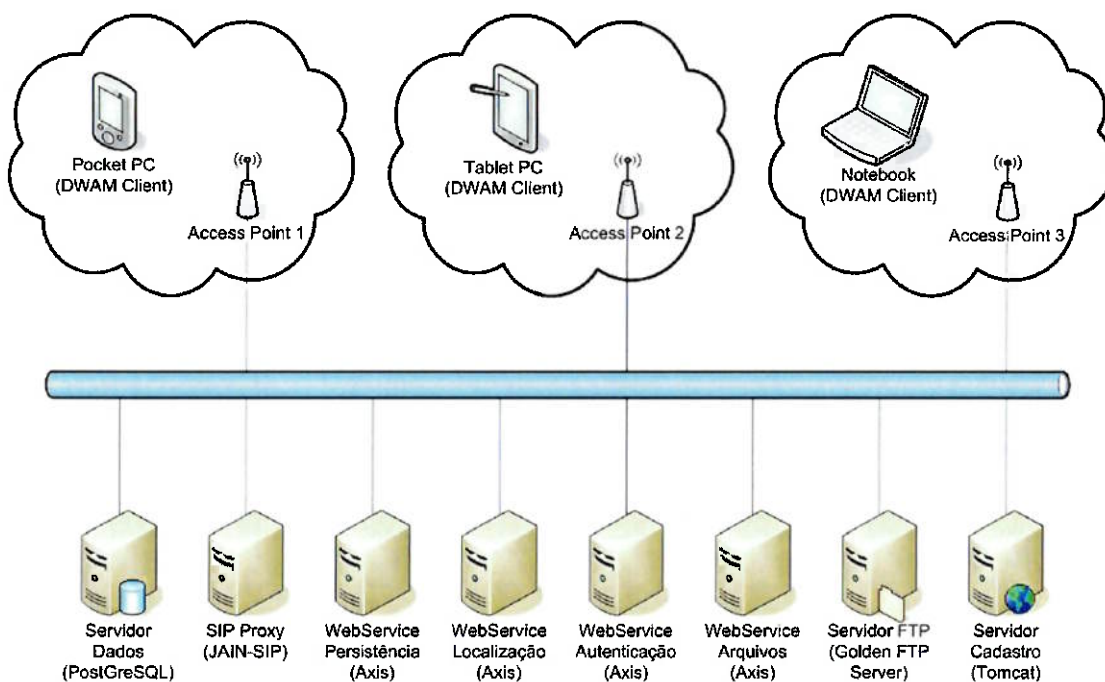


FIGURA 17: Diagrama da infra-estrutura de hardware do DWAM

Os usuários poderão ter acesso ao sistema por meio da rede sem fio. O dispositivo móvel cliente, que poderá ser um *Pocket PC*, *tablet PC* ou um *notebook* equipado com a tecnologia Wi-Fi, deverá tanto estar cadastrado no sistema para que seu endereço MAC seja autenticado pelo *access point*, como também deverá possuir instalado o aplicativo cliente do DWAM (*DWAM Client*). Através deste aplicativo, é realizada a autenticação de usuário e o acesso aos demais serviços que sistema disponibiliza. Na implementação e nos testes realizados, foram utilizados *Pockets PCs* da Hewlett-Packard modelos HP iPaq h5550, cujo sistema operacional instalado é o Microsoft Windows CE versão 4.2. Para executar o *DWAM Client*, que é implementado em Java, foi instalado no *Pocket PC* a máquina virtual NSIcom CrEme, que oferece suporte à tecnologia J2ME.

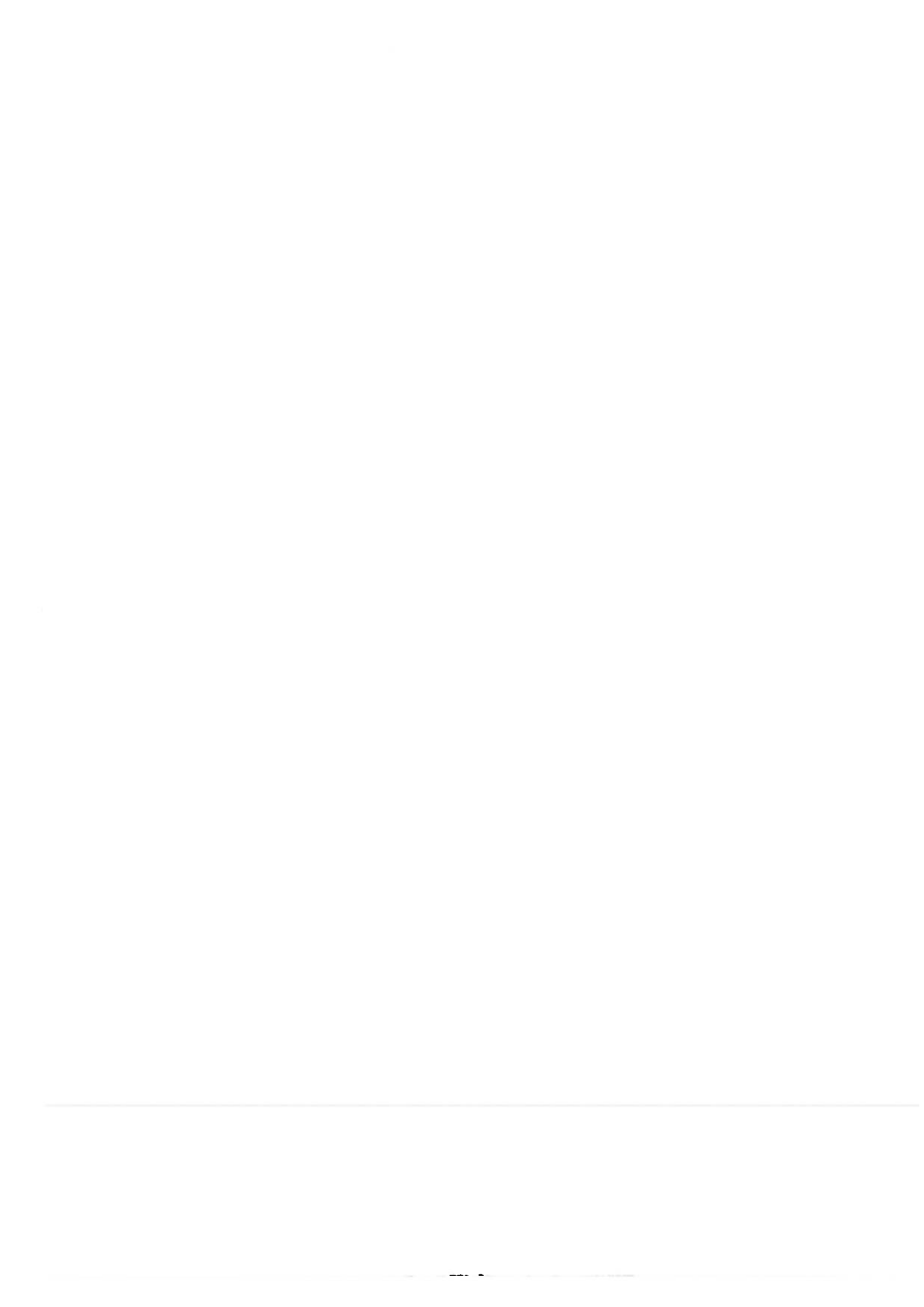
O sistema DWAM foi projetado de forma que os servidores pudessem ser implantados de forma distribuída, com cada serviço sendo executado por uma máquina separada. Estes não precisam estar necessariamente separados, podendo também ser instalados em um único servidor. Porém, a arquitetura de software do DWAM, dividida em módulos e baseada no *middleware WebServices*, é flexível a ponto de permitir que os servidores sejam distribuídos fisicamente da maneira mais conveniente. A única configuração que deve ser feita é o cadastro no sistema dos endereços IP das máquinas onde cada serviço está instalado.

Os servidores e os *access points* deverão estar interconectados por meio de uma rede Ethernet cabeada, de preferência utilizando *switches* para melhorar o desempenho da rede. Os *Access Points* modelo Linksys WRT54g serão utilizados na implementação e serão configurados especialmente para a solução. A topologia da rede proposta é topologia em estrela (*hub and spoke*) e será cabeada utilizando cabo par-trançado categoria 5.

4.2.2 Arquitetura de Software

O sistema DWAM foi elaborado com base numa arquitetura de software de três camadas, sendo uma camada cliente, uma de aplicação e outra de dados. O *middleware* escolhido para a integração entre os módulos do sistema foi a tecnologia *Web Services*, por apresentar características de interoperabilidade e por possibilitar a distribuição dos serviços. Deste modo, diminui-se a dependência e o acoplamento entre os módulos, permitindo que um módulo possa ser reimplementado ou substituído de forma transparente para os outros módulos.

Por meio da figura 18 é possível visualizar a distribuição dos módulos do DWAM entre as camadas, bem como os protocolos utilizados na comunicação entre eles. Entre parênteses, em cada módulo, estão os nomes das tecnologias e APIs usadas em sua implementação.



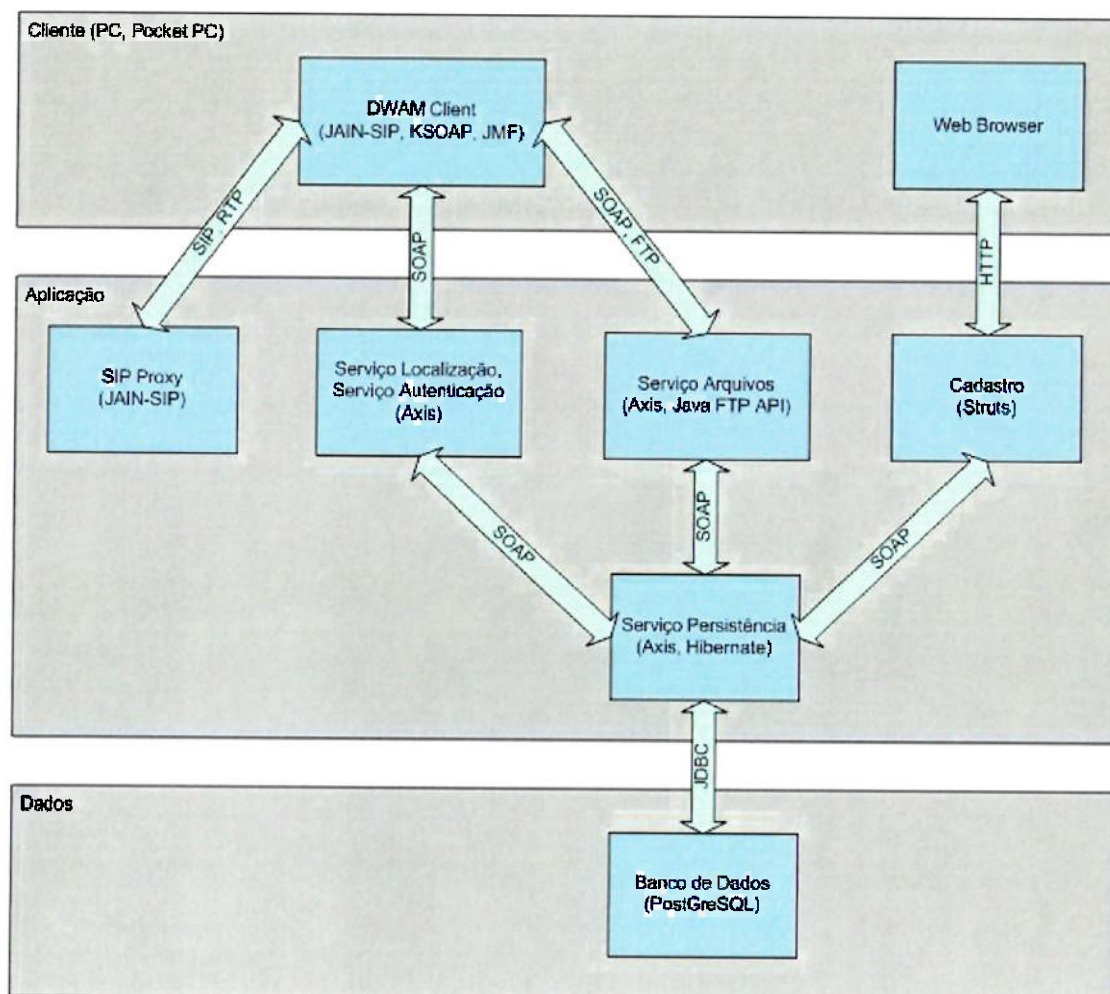


FIGURA 18: Diagrama da arquitetura de software do DWAM

Na camada cliente, representada pelos dispositivos móveis que se conectarão ao sistema, estará instalado o módulo *DWAM Client*, aplicativo através do qual o usuário acessará os diversos serviços. Por meio do *DWAM Client*, o usuário poderá interagir com outros através de mensagens instantâneas ou de voz, visualizar a localização de seus contatos e baixar arquivos associados à sua localidade. A troca de mensagens ocorre via SIP (com adição do RTP para voz) e é intermediada pelo *SIP Proxy*; enquanto o acesso aos módulos de localização, autenticação e arquivos é realizado via *Web Services*, pelo protocolo SOAP.

Além do *DWAM Client*, pode-se considerar como parte da camada cliente o *web browser* utilizado para acessar o módulo de cadastro.

Em outra camada, a de aplicação, temos o *SIP Proxy*, o módulo de Cadastro e os serviços de Autenticação, Localização, Arquivos e Persistência.

O *SIP Proxy* é o módulo responsável por gerenciar a presença dos usuários conectados e encaminhar as mensagens instantâneas e de voz de um usuário para outro.

Já o módulo de Cadastro é composto por páginas *web*, que serão acessadas por usuários administradores do sistema. Através deste módulo, estes administradores poderão realizar a manutenção do cadastro de usuários, de dispositivos e de localidades. Para a persistência dos dados no banco, o módulo de cadastro acessa o serviço de persistência via *Web Services*.

O serviço de Localização é o responsável por determinar em que localidade cada usuário está conectado, acessando informações dos *access points*, da rede e do serviço de Persistência. Enquanto o serviço de Autenticação é o que possibilita a realização do *login* dos usuários no sistema, através de uma validação com senha. A aplicação de compartilhamento de arquivos é oferecida pelo serviço de Arquivos, que exhibe para o cliente os arquivos disponíveis na localidade à qual este se encontra conectado, realizando assim, a transferência via FTP.

Estes últimos três módulos, por sua vez, realizam o acesso ao banco de dados através do serviço de persistência. Este serviço encapsula o acesso aos dados, trabalhando como uma interface entre a camada de aplicação e a camada de dados. Deste modo, elimina-se o acoplamento entre a implementação dos serviços e o banco de dados utilizado.

Por fim, a camada de dados é representada pelo banco PostGreSQL, escolhido por ser uma ferramenta livre que atende aos requisitos do sistema.

4.2.3 Aplicativo Cliente

O aplicativo cliente foi desenvolvido tomando como referência o projeto *Applet Phone* desenvolvido pelo JAIN, que utiliza a API JAIN-SIP para sinalização com o protocolo SIP o JMF para a captura e reprodução de voz. Este projeto consistia em um Java *applet* que funcionava como mensageiro instantâneo com suporte a comunicação por voz. O *Applet Phone* é implementado com base em um *framework* de envio e tratamento de mensagens SIP, contendo dados de texto ou de áudio. Com acesso aos códigos-fonte do projeto *Applet Phone* e do JAIN-SIP, que são abertos, foi possível desenvolver o *DWAM Client* a partir deste *framework*, inclusive realizando adaptações para permitir o funcionamento do aplicativo no *Pocket PC*.

O *DWAM Client* oferece ao usuário um conjunto de funcionalidades relacionadas à comunicação e interação entre pessoas em redes sem fio. São elas:

- Comunicação por mensagens instantâneas
- Comunicação por voz
- Atualização de status de usuário
- Localização de usuários
- Download de arquivos associados à localidade

A interface do *DWAM Client* foi implementada utilizando a API Swing do Java, que é suportada pela máquina virtual CrEme no *Pocket PC*. A janela principal do *DWAM Client* foi criada a partir da classe *JFrame* do Java Swing e oferece acesso aos serviços por meio de abas.

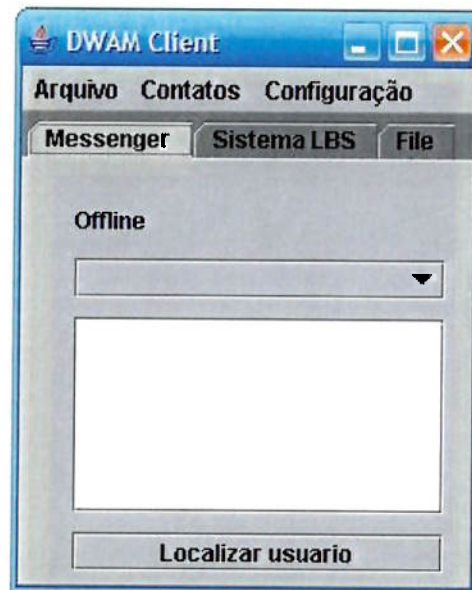


FIGURA 19: Interface principal do DWAM Client

Para realizar a implementação das funcionalidades de mensageiro instantâneo do DWAM Client, foram utilizadas algumas classes do *framework* de mensagens SIP do *Applet Phone* do JAIN. Por meio destas classes, que são apresentadas a seguir, o DWAM Client envia e recebe mensagens do SIP Proxy, que realiza o roteamento das mensagens de todos os usuários registrados.

- *MessengerController*: Classe que possui os métodos de manipulação de contatos, permitindo adicionar e remover contatos e atualizar suas informações de status. Foi adaptada para permitir a exibição do nome da localidade dos contatos junto ao status.
- *MessengerManager*: Classe que expõe as funções de registro do usuário no SIP Proxy, envio e tratamento de mensagens de texto e gerenciamento de mensagens de voz para a interface com o usuário.
- *ChatSessionManager*: Gerencia as conversas com outros usuários.

- *MediaManager*: Classe que gerencia a transmissão e recepção de áudio durante as chamadas de voz. Utiliza o JMF para capturar e reproduzir som e o RTP para realizar o controle da transmissão.

Além do código do *framework* do *Applet Phone*, foi utilizado também o código do próprio JAIN SIP. Foram necessárias algumas modificações no código para que a API pudesse ser executada no *Pocket PC*. Alguns recursos, como o suporte a SSL, não eram compatíveis com a máquina virtual CrEme. Por isso, estes recursos foram simplificados ou suprimidos para uso no sistema DWAM, porém sem ocasionar perda de funcionalidade.

A funcionalidade de comunicação por voz, prevista para funcionar em todos os dispositivos móveis suportados pelo sistema, não foi possível de ser implementada no *Pocket PC*. Apesar da reprodução de áudio através do JMF funcionar no PDA, a captura de voz não era possível pois o JMF não era capaz de detectar o microfone do *Pocket PC*. Para resolver esse problema, seria necessário desenvolver uma biblioteca nativa capaz de acessar o microfone do *Pocket PC*, uma vez que não existe disponível nenhuma no mercado. Somente por meio desta biblioteca, o JMF conseguiria capturar o áudio do microfone. Como o desenvolvimento de código nativo demandaria um esforço e tempo fora do escopo do projeto, decidiu-se por implementar a comunicação por voz somente em *notebooks* e *tablet PCs*.

O acesso aos outros serviços (Autenticação, Localização e Arquivos) é feito via *Web Services* e foi implementado na camada cliente utilizando o KSOAP. O KSOAP é uma API feita em Java que permite o envio e recepção de envelopes SOAP para acesso a *Web Services*. Esta API foi utilizada por ser compatível com J2ME e funcionar no *Pocket PC*. O KSOAP permite construir e enviar chamadas de métodos via SOAP e converter as mensagens SOAP recebidas em objetos Java. Para encapsular o acesso aos *Web Services*, foi desenvolvida uma classe *ServerStub* responsável pela criação das

mensagens XML a partir das chamadas de métodos nos servidores e retornar os resultados de forma transparente para o cliente.

Os serviços de Arquivos e Mapas disponibilizados no cliente do sistema DWAM são aplicações simples que através de acesso *Web Services* recuperam informações que são utilizadas para fazer a interface com o usuário por meio de telas geradas dinamicamente.

O sistema de Arquivos utiliza-se de uma API específica para transporte de arquivos chamada JavaFTP API. Esta API foi escolhida dada sua facilidade de implementação para disponibilizar download e upload de arquivos e pelo seu correto funcionamento nos *Pockets*.

O sistema de Mapa baseou-se no uso da API Swing do Java para construir uma interface de um mapa dinâmico onde pudessem ser indicadas as informações de localização dos usuários. Para o correto funcionamento desta aplicação foi necessário a inclusão do pacote “rt.jar” do Java na aplicação cliente do DWAM.

A figura 20 pode exemplificar o sistema de mapa. Essa gravura representa o andar térreo do prédio de engenharia elétrica da poli, onde a região de formato oval, do lado esquerdo, representa o auditório e o grande retângulo à direita, a biblioteca. O pequeno ponto vermelho do mapa indica a posição do próprio usuário.



FIGURA 20: Mapa gerado dinamicamente com posição do usuário

4.2.4 SIP Proxy Server

O *SIP Proxy Server* é o módulo responsável por gerenciar a comunicação por mensagens de um usuário para outro. É um elemento essencial da arquitetura SIP, uma vez que é ele quem realiza a interpretação das mensagens, gera respostas e encaminha as requisições para os respectivos destinatários.

Como o escopo deste projeto não prevê a re-implementação da arquitetura do SIP, optou-se por utilizar uma implementação pronta do *SIP Proxy Server*. Existem no mercado diversos *proxy servers* disponíveis para diferentes aplicações. Desta forma, para o sistema DWAM, foi utilizado o *Proxy Server* do JAIN, escolhido por possuir código aberto e por utilizar a mesma API para SIP que o *Applet Phone*.

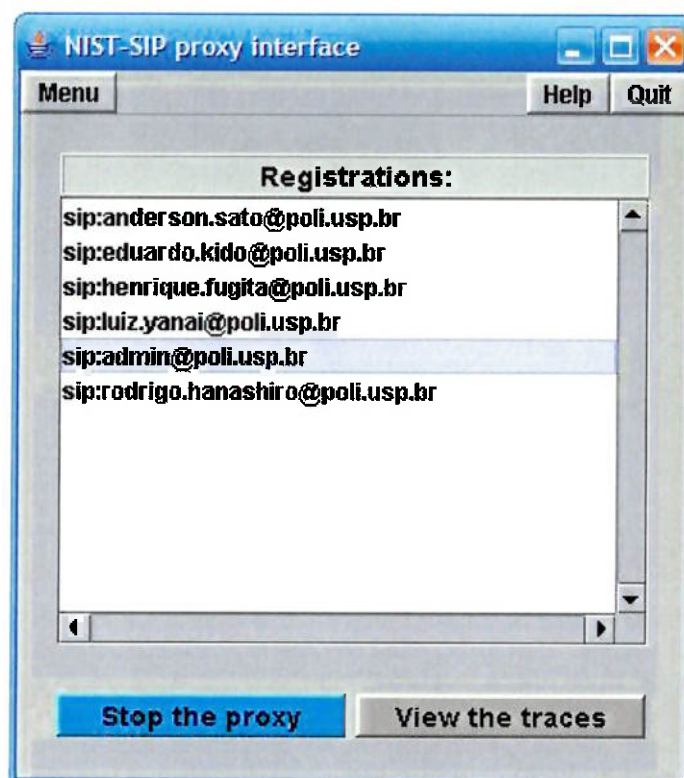


FIGURA 21: Interface do JAIN SIP Proxy

Durante o *login* de usuário, o *DWAM Client* acessa o *SIP Proxy* e realiza o registro do usuário, de maneira que este esteja disponível para ser contactado pelos outros usuários.

Originalmente, o *SIP Proxy* utilizava um arquivo XML para armazenar a lista de usuários permitidos. Entretanto, seu código foi alterado de modo que estes usuários fossem recuperados a partir do cadastro no banco de dados. Desta maneira, o *proxy* passou a acessar via *Web Services* o módulo de persistência para obter a lista de usuários cadastrados.

4.2.5 Web Services

Para integrar os módulos do sistema DWAM, a camada de *middleware* foi implementada usando a tecnologia *Web Services*, escolhida por permitir interoperabilidade entre aplicações em diversas plataformas, ser independente de fabricante e garantir desacoplamento entre os módulos. Dessa maneira, ganha-se flexibilidade, pois cada módulo pode ser implementado de modo independente, utilizando a plataforma mais adequada.

Apesar de haver a possibilidade de implementar cada serviço com tecnologias diferentes, optou-se por desenvolver todos os serviços seguindo o mesmo *framework*, que é descrito a seguir. Entretanto, foi deixada aberta a possibilidade de possíveis adições e implementações futuras em qualquer outra plataforma.

O *framework* desenvolvido para a criação dos *Web Services* em Java baseou-se na ferramenta Apache Axis e no *design pattern Stub*.

O Apache Axis é composto por uma API que implementa o protocolo SOAP e um container para disponibilização de *Web Services*. Através da API, é possível realizar

acesso aos serviços, pois ela é responsável por realizar a conversão das chamadas de métodos e dos objetos Java para mensagens SOAP e vice-versa. Já o container consiste em um *servlet* que é executado dentro do Jakarta Tomcat, onde devem ser instaladas as classes Java que servirão de implementação dos *Web Services*. Uma vez configurado, o container encarrega-se de tornar esses serviços disponíveis para serem acessados via SOAP. Assim, para disponibilizar os serviços do DWAM, o servidor necessita apenas possuir o Tomcat com o container do Axis.

O *design pattern Stub* foi utilizado para tornar transparente para o código cliente o acesso aos serviços remotos. Cada serviço possuía uma classe *stub* correspondente, de modo que caso algum módulo precisasse acessar outro via *Web Services*, bastaria chamar um método na classe *stub* localmente ao invés de construir e enviar uma chamada de método SOAP. O *stub* é o responsável por realizar esta chamada, tornando a implementação do acesso remoto transparente para o cliente.

Desta forma, foram construídas as classes *stubs* que realizavam as chamadas de métodos SOAP por meio da API do Axis. Cada serviço, assim, poderia estar rodando em um servidor separado, mas seria capaz de acessar todos os outros remotamente, bastando para isso acessar as classes *stubs*.

4.2.6 Serviço de Autenticação

O serviço de Autenticação é bastante simples, sendo responsável por realizar o *login* dos usuários no sistema. O método *login*, disponível via *Web Services*, autentica o nome do usuário e senha, além de verificar se este possui permissão para acessar o serviço desejado.

Para realizar a autenticação, o serviço acessa o banco de dados, realizado através do serviço de Persistência, e compara o nome e senha passados com aqueles cadastrados

no banco. Em caso de *login* para serviços específicos, é verificado se o usuário possui a permissão necessária.

Para futuras implementações, pode-se utilizar criptografia neste serviço, de modo a tornar o login mais seguro.

4.2.7 Serviço de Localização

A grande evolução da tecnologia WLAN tornou possível o grande desenvolvimento de aplicações e serviços para este tipo de rede. Devido ao alcance restrito das células, determinado pelo alcance dos *Access Points* nessas redes, torna-se possível o desenvolvimento do serviço de localização, através da identificação do *Access Point* onde o dispositivo móvel do usuário está conectado.

Existem diversas maneiras de se implementar o serviço de localização através da identificação do *Access Point* na qual o usuário está conectado. Uma delas é através de mensagens SNMP transmitidas do *Access Point* para o servidor de localização. Porém este método necessita de funcionalidades específicas, encontradas apenas em *Access Points* de alto custo no mercado. Devido a essas restrições, para este presente estudo foi utilizado *Access Points* que utilizam plataformas abertas e que sejam encontrados a baixo custo no mercado. Além disso, foi desenvolvida uma metodologia de localização de usuários, aproveitando a plataforma aberta do *Access Point* que possibilita desenvolver novas funcionalidades no equipamento de acordo com as necessidades da aplicação.

A implementação do sistema de localização pode ser dividida em duas partes:

a) Hardware – descreve todo o desenvolvimento necessário para a configuração dos *Access Points* e as funcionalidades necessárias para a implementação do serviço de localização;

b) Software – descreve o desenvolvimento do servidor para a captura das informações e o acesso ao banco de dados.

O Access Point é configurado de forma a listar os usuários conectados no dispositivo e exibi-los em uma página web, através de um pequeno servidor web implementado no próprio dispositivo. O servidor então, é responsável em acessar essas páginas conforme a demanda e descobrir onde o usuário requisitado pelo sistema está conectado.

4.2.7.1 Hardware

O Access Point escolhido para a implementação do sistema foi o WRT54g v4 da Linksys, que além da funcionalidade de *Access Point*, possui também as funcionalidades de roteador e de *switch* para uma rede local. O WRT54g utiliza *chipset* da Broadcom que é controlado por uma versão especialmente desenvolvida do sistema operacional Linux instalado no WRT54g. A utilização de Linux embarcado torna possível o desenvolvimento de novos *firmwares* com novas facilidades especialmente desenvolvidas para o dispositivo. Existem diversos *firmwares* desenvolvidos para o dispositivo, sejam eles oficiais e extra-oficiais, como por exemplo o da Svesoft, Hyper-WRT e o DD-WRT.

Para a implementação do sistema de localização, foi escolhido o DD-WRT, um *firmware* que possibilita configurar diversas funcionalidades, dentre elas o controle da potência do sinal de transmissão do sinal do *Access Point* (dado pela configuração de uma parâmetro da NVRAM do dispositivo) e a listagem dos dispositivos conectados no *Access Point*, através do comando “*wl assoclist*” do chipset WLAN do aparelho. As

informações são exibidas através do httpd um micro-servidor *web* implementado no WRT54g. Ele serve de interface para a configuração de parâmetros do WRT54g e para as suas funcionalidades diferenciadas.

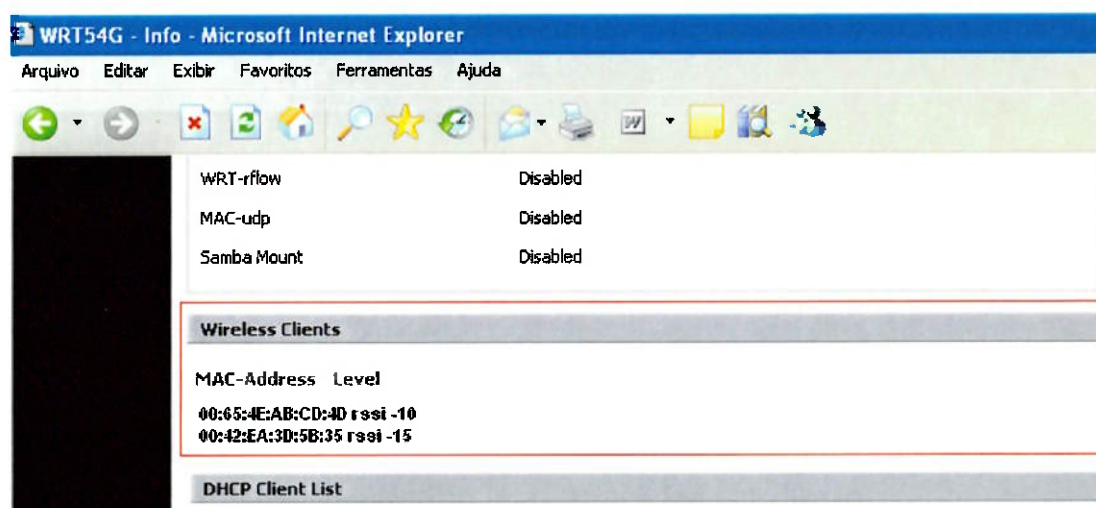


FIGURA 22: Lista dos Endereços MAC conectados

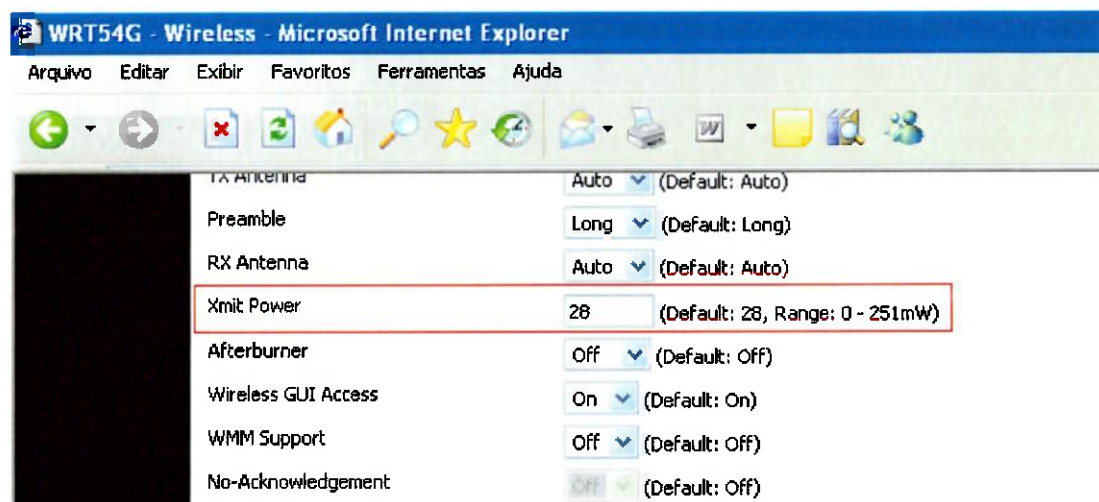


FIGURA 23: Regulagem da potência de transmissão

4.2.7.2 Software

O sistema de localização foi desenvolvido em Java, utilizando como camada de persistência o Hibernate e o *framework* Axis para o oferecimento do serviço de *Web Service*. O servidor oferece o serviço de localização através de *Web Service* que possibilita a localização através do endereço IP do usuário conectado. A partir da demanda do sistema de localização de um usuário, o servidor de localização executa um processo que é tratado de acordo com a ordem descrita abaixo:

1) Mapeamento do IP para MAC

O sistema de localização passa para o servidor de localização o endereço IP do usuário a ser localizado. Porém, a lista de usuários conectados, armazenada nos *Access Points*, contém apenas o endereço MAC dos usuários. Sendo assim o módulo de mapeamento IP-MAC executa um *ARP Request* na rede, obtendo assim o endereço MAC correspondente do usuário, possibilitando assim, a sua localização. O *ARP Request* foi desenvolvido em C++, devido a restrições da linguagem Java na implementação desse serviço. O Java não possui suporte a estes tipos de serviço em linguagem nativa, necessitando o auxílio de bibliotecas C/C++ acessadas através de interfaces especializadas. No servidor de localização, a aplicação desenvolvida em C++ é acessada diretamente pelo código Java, como uma aplicação externa, retornando o endereço MAC do IP solicitado.

2) Módulo de Busca dos Access Points

O módulo de Busca do *Access Points* é responsável por acessar e extrair as informações das páginas referentes à exibição das listas de usuários de cada *Access Point* cadastrado na tabela de *Access Points* do banco de dados do sistema. Os *Access Points* são acessados de maneira seqüencial de acordo com a tabela, e o endereço MAC do usuário

é procurado nas listas de MACs conectados. Quando o sistema encontra o MAC do usuário em uma das listas, o servidor de localização retorna o nome cadastrado do *Access Point* para o sistema, terminando assim o processo de localização.

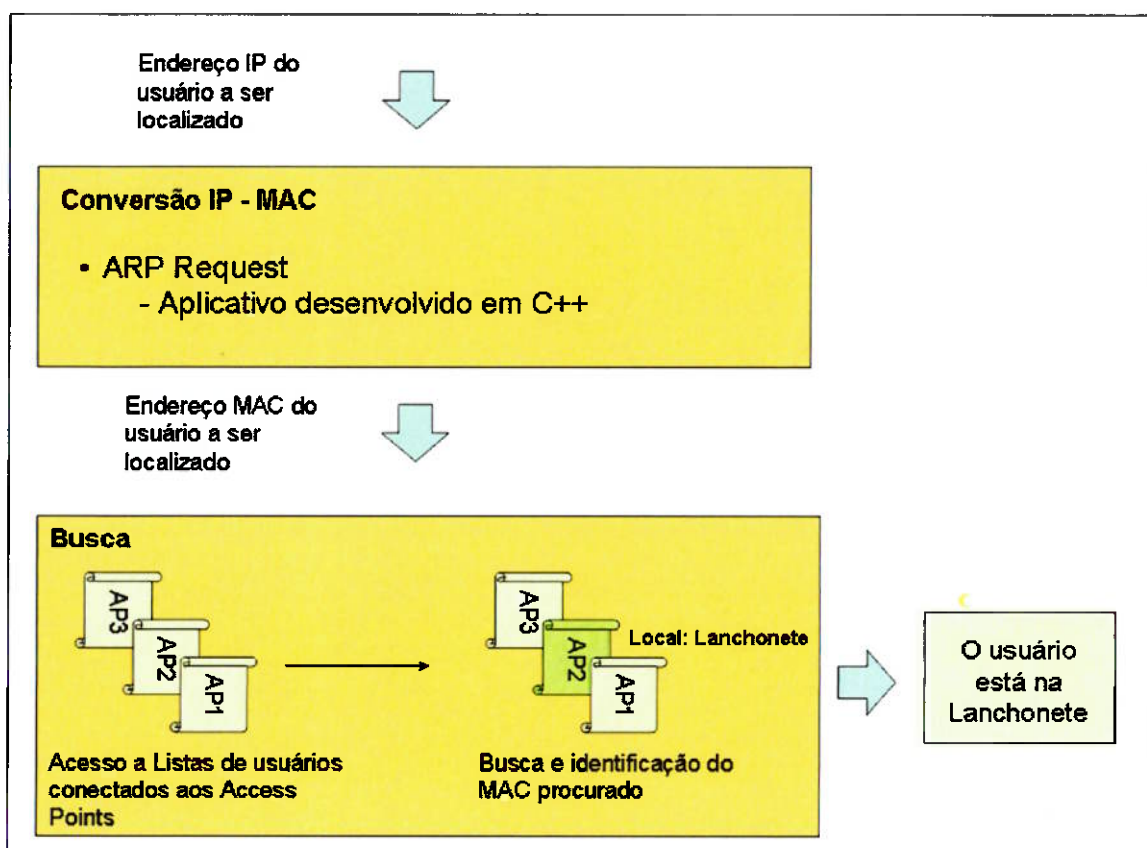


FIGURA 24: Diagrama do tratamento das requisições no servidor de localização

4.2.8 Serviço de Arquivos

Os serviços de Arquivos e Mapas disponibilizados no cliente do sistema DWAM são aplicações simples que através de acesso WebServices recuperam informações que são utilizadas para interfacear com o usuário por meio de telas geradas dinamicamente.

O sistema de Arquivos utiliza-se de uma API específica para transporte de arquivos chamada JavaFTP API. Esta API implementa o protocolo de transferência de arquivos

FTP. Esta API foi escolhida dada sua facilidade de implementação para disponibilizar download e upload de arquivos e pelo seu correto funcionamento nos Pockets. A API foi desenvolvida por **Florent Cueto** e é um código aberto.

O sistema de Mapas baseou-se no uso da API Swing do Java para construir uma interface de um mapa dinâmico onde pudessem ser indicadas as informações de localidade dos usuários. Uma vez que o usuário esteja logado no sistema, ele pode requisitar a localização de alguém com que esteja conversando. Através de acesso ao WebServices do Serviço de Localização ele recupera as informações associadas a cada localidade no mapa. No mapa construído pelo Swing existem pontos clicáveis associado as localidades, onde o usuário pode clicar e receber informações sobre a localidade específica tais como nome do local, arquivos disponíveis na localidade, etc. Para o correto funcionamento desta aplicação foi necessário a inclusão do pacote “rt.jar” do Java na aplicação cliente do DWAM devido a falta de algumas classes no CrE-ME.

4.2.9 Módulo de Cadastro Web

O módulo de cadastro tem o intuito de permitir aos administradores do sistema e de localidades uma fácil manipulação dos usuários, grupos de usuários, dispositivos e localidades. Em função disso, o módulo foi desenvolvido como uma aplicação *web* composta por páginas JSP e seguindo o *Struts Framework*.

O *Struts Framework* é um padrão *open-source* desenvolvido e mantido pela *Apache Software Foundation* e por uma comunidade de voluntários com o objetivo de fornecer um *framework* para facilitar o desenvolvimento de aplicações *web*.

Este *framework* segue o padrão MVC (Model-View-Controller) para aplicações Java para Internet. O padrão MVC separa de forma clara a camada de apresentação da camada de negócio.

Para um melhor entendimento do Struts Framework segue abaixo o diagrama de seu funcionamento e as devidas explicações.

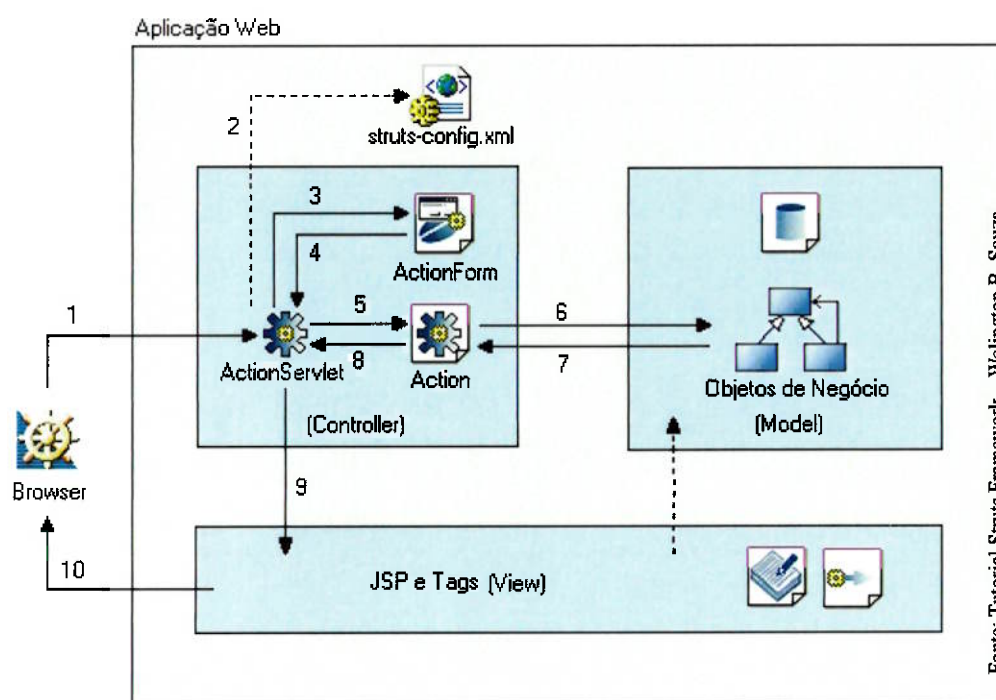


FIGURA 25: Diagrama de Funcionamento do *Struts Framework*

1. O usuário faz uma solicitação através de uma *url* no *browser*. Por exemplo, no nosso caso <http://localhost:8080/DWAM.Web/welcome.do>.
2. Se for a primeira solicitação que o container recebeu para esta aplicação, será invocado o método *init()* do *ActionServlet* e carregado as configurações do arquivo “*struts-config.xml*”. Este arquivo possui todo o mapeamento e o fluxo de componentes e páginas a serem utilizadas pela aplicação.

3. Baseado neste arquivo de configuração, o *ActionServlet* invoca a respectiva classe *ActionForm* para validação dos dados recebidos da solicitação que vem do *browser*.
4. Se não for validado, o usuário recebe novamente o formulário com o motivo da falha para que seja feita novamente a solicitação, mas agora de forma correta. Caso seja validado, retorna-se para o *ActionServlet*.
5. No *ActionServlet* é invocada uma classe *Action* que será responsável pela comunicação com a camada de negócio.
6. A classe *Action* irá delegar tarefas para camada de negócio através de chamadas de métodos de suas classes.
7. A camada de negócio pode tanto persistir dados no banco de dados como também popular um objeto ou coleção e devolvê-lo para a *Action*.
8. Ao receber as respostas da camada de negócio, a *Action* irá tomar as medidas necessárias e adicionar os objetos recebidos na sessão do usuário. E retorna ao *ActionServlet*.
9. Baseado no mapeamento da *Action* no arquivo de configuração, o *Action Servlet* redireciona para uma página JSP correspondente.
10. Como a página JSP é dinâmica, esta é compilada na camada de apresentação (no servidor Tomcat) e enviada de volta ao *browser* que fizera a solicitação. Os objetos contidos na sessão do usuário são utilizados no momento da compilação da página.

Seguindo esta estrutura o módulo de cadastro foi desenvolvido. No entanto, foi feita uma simplificação devido às características da aplicação. Basicamente, a estrutura ficou dividida em:

- Camada de Apresentação: páginas JSP. Nestas páginas ao invés de trabalhar com *Scriptlets* (código Java dentro das páginas) foram utilizadas *tags* customizadas que fazem parte do próprio *Struts Framework*.

- Camada de Controle: arquivo de configuração, as classes *Action* e *ActionForm* (o conceito de *ActionServlet* está implícito nas outras duas classes). Se houver necessidade, a *Action* é a responsável por manipular os dados vindos da camada de negócio.
- Camada de Negócio: resume-se a classes para chamadas do serviço de *Web Services* para uma consulta ou uma inserção de dados no banco.

A implementação do módulo de cadastro seguiu a especificação dos casos de uso do sistema DWAM. Foram levados em consideração apenas os casos de uso relacionados de alguma forma com o cadastro: 1, 2, 3 e 4. A descrição dos mesmos encontra-se no Anexo A.

Para o desenvolvimento da camada de controle e de negócio foi utilizada a plataforma Eclipse SDK, uma poderosa ferramenta gratuita de desenvolvimento de aplicações Java.

Já as páginas *web* foram escritas em um editor de texto normal e foram publicadas em um servidor *web* também gratuito, o Apache Jakarta Tomcat Servlet/JSP Container. Este servidor não só é compatível com o *Struts Framework*, como também foi utilizado nos próprios exemplos dados pelo *Apache Struts Project*.

A interface homem-máquina foi desenvolvida visando a alcançar ao mesmo tempo simplicidade (uma interface não muito poluída) e boa usabilidade (uma interface amigável) que não divirja do padrão encontrado no mercado. Pode-se visualizar uma das telas do módulo de cadastro na figura 26.

Gerenciamento de Usuários

Esta é a página de Gerenciamento de Usuários do DWAM. A partir desta página o administrador do sistema pode adicionar, alterar, remover ou consultar um usuário no cadastro do DWAM.

Lista de Usuários

[Adicionar Usuário](#) primeiro | anterior | próximo | último

ID	Usuário	Nome Completo	Tipo	Editar	Apagar
1	usuario1@poli.usp.br	Usuario1	1	Editar	Apagar
2	usuario2@poli.usp.br	Usuario2	1	Editar	Apagar
3	usuario3@poli.usp.br	Usuario3	1	Editar	Apagar
4	usuario4@poli.usp.br	Usuario4	1	Editar	Apagar
5	usuario5@poli.usp.br	Usuario5	1	Editar	Apagar
6	usuario6@poli.usp.br	Usuario6	1	Editar	Apagar
7	usuario7@poli.usp.br	Usuario7	1	Editar	Apagar
8	usuario8@poli.usp.br	Usuario8	1	Editar	Apagar
9	usuario9@poli.usp.br	Usuario9	1	Editar	Apagar
10	usuario10@poli.usp.br	Usuario10	1	Editar	Apagar

FIGURA 26: Página *web* de Gerenciamento de Usuários no sistema DWAM.

4.2.10 Serviço de Persistência

Este é o *Web Service* que encapsula o acesso ao banco de dados. Todo mecanismo de persistência é implementado neste módulo e exposto para os outros na forma de

métodos SOAP. Para a criação do *Web Service* foi utilizado o mesmo *framework* que nos demais, composto pelo Apache Axis e pelo design *pattern Stub*.

Para a implementação da persistência dos dados no banco, foi empregado o Hibernate, que é uma ferramenta de código aberto para conversão de objetos Java em entradas tabelas relacionais. Com o Hibernate, é possível criar classes persistentes cujos atributos são mapeados diretamente para colunas de tabelas. Podem ser mapeados inclusive relacionamentos entre classes persistentes. Assim, instâncias destas classes podem ser armazenadas como registros em um banco de dados.

A manipulação dos objetos persistentes se dá por meio da classe Session do Hibernate, que se encarrega de abrir uma conexão JDBC com o banco de dados e realizar os acessos, inclusive com controle de transações. Através da Session, pode-se armazenar objetos no banco, alterá-los, removê-los e realizar buscas. As buscas por objetos persistentes são realizadas utilizando-se o HQL (Hibernate Query Language), que é bastante semelhante ao SQL, porém faz referências a objetos e atributos ao invés de tabelas e colunas.

O Hibernate também é compatível com a grande maioria dos bancos de dados existentes no mercado, podendo ser facilmente configurado para qualquer um deles. Como o acesso aos dados persistentes se dá apenas por meio do Hibernate, para aplicação é transparente qual gerenciador de banco de dados está sendo utilizado, criando desacoplamento entre as camadas de aplicação de dados.

Desta maneira, o serviço de persistência oferece via *Web Services* para os outros módulos do sistema os seguintes métodos para acesso ao banco de dados:

- Salvar: Recebe como parâmetro uma instância de uma classe persistente e a insere na respectiva tabela no banco de dados.

- Atualizar: Recebe como parâmetro um objeto já existente no banco, porém com atributos modificados, persistindo as modificações.
- Remover: Remove do banco de dados o objeto persistente recebido como parâmetro.
- BuscaTodos: Retorna todos os objetos de uma classe persistente, isto é, todas as linhas de uma determinada tabela.
- BuscaPorParâmetros: Realiza uma busca parametrizada em HQL, retornando uma coleção de objetos como resultado.

4.2.11 Banco de Dados

O gerenciador de banco de dados utilizado nesta implementação do DWAM foi o PostgreSQL, por ser um software livre e atender aos requisitos de volume de dados do sistema.

A criação das tabelas foi realizada a partir das configurações de mapeamento do Hibernate. Devido ao fato de todo o acesso ao banco ser feito através do Hibernate, uma eventual migração para outro gerenciador de banco de dados pode ser feita de maneira transparente para o restante da aplicação, bastando apenas alterar a configuração do Hibernate.

4.3 Testes e Resultados

Depois realizada toda a instalação física, iniciamos os testes dos softwares que compõem o DWAM. Cada módulo foi, inicialmente, testado individualmente e, depois de validado, testou-se o sistema integrado.

Foram testadas tanto as sessões estabelecidas entre equipamentos semelhantes como entre equipamentos diferentes, ou seja, a comunicação entre dois notebooks, dois

PDA, dois tablets e entre um notebook e um PDA, um notebook e um tablet e entre um PDA e um tablet.

4.3.1 Envio de mensagens instantâneas

O módulo do IM foi testado em três partes:

- Dois equipamentos no mesmo domínio local: não houve nenhum problema de comunicação.
- Equipamentos em domínios locais diferentes: Assim como o item acima, não houve problemas com os equipamentos localizados em domínios diferentes.
- Um equipamento no domínio local e outro no domínio global: Conforme esperávamos, as funcionalidades do IM também funcionaram adequadamente.

4.3.2 Realização de chamadas de voz

As chamadas de voz foram, da mesma forma que o módulo de IM, testado em três etapas:

- Dois equipamentos no mesmo domínio local: as sessões de VoIP foram estabelecidas com sucesso. Apesar de a qualidade da comunicação não ter sido como a da telefonia tradicional, as falhas ocorridas não comprometeram a comunicação, permitindo que os usuários pudessem se comunicar normalmente.
- Equipamentos em domínios locais diferentes: novamente não encontramos problemas.
- Um equipamento no domínio local e outro no domínio global: esse teste também indicou que o módulo de voz estava adequado.

4.3.3 Serviço de localização

Para o serviço de localização foi realizado testes em supondo somente duas situações já que não existe, do ponto de vista de aplicação, necessidade de se procurar por alguém que esteja na mesma sala que o usuário.

Assim, testaram-se as seguintes situações:

- Equipamentos em domínios locais diferentes;
- Um equipamento no domínio local e outro no domínio global.

Apesar de o sistema ter reconhecido o AP ao qual o usuário havia se conectado, foi detectado um problema que estávamos esperando. Como o manuseio de um equipamento sem fio é ainda bastante difícil no que diz respeito ao controle do sinal, muitas vezes, nas áreas de intersecção, não tínhamos o controle de qual AP o usuário se conectaria. Isso limitou o uso dessa aplicação.

4.3.4 Disponibilidade de arquivos

Depois de conectado ao AP local adequado, verificamos se realmente conseguíamos agendar o horário de disponibilidade dos arquivos. Conseguimos fazer com que apenas o usuário presente na sala pudesse ter acesso ao material eletrônico. Porém, ainda devido ao pouco controle sobre redes sem fio, alguns usuários conseguiram se conectar ao AP da sala mesmo não estando dentro dela. Notamos que se nos aproximássemos do domínio local, já conseguíamos ter acesso a ele.

5 Considerações finais

Apresentamos a seguir as conclusões que o grupo obteve com o projeto. Além disso, também sugerimos idéias de melhorias e avanços para projetos que sigam a mesma linha de estudo do DWAM.

5.1 Conclusão do Projeto

Desde o início do projeto, o objetivo do grupo foi desenvolver aplicações para o segmento de equipamentos móveis não só por acreditar no potencial de seu crescimento mas também por se mostrar uma área de bastante interesse à todos os integrantes.

Conforme mostravam as tendências, o crescimento do segmento de tecnologia explorado pelo DWAM tem cada vez mais correspondido ao nosso estudo inicial. Não só pela tecnologia de VoIP, que as empresas já estão procurando implantar para reduzir custos, mas também o fator de convergência e mobilidade, assuntos que têm sido bastante discutidos.

As empresas de tecnologia estão investindo muito nessas áreas, o que faz desse projeto bastante atual e contextualizado com a situação vigente do mundo tecnológico.

A utilização de tecnologias convergentes amplia as possibilidades de desenvolvimento de aplicações, oferecendo um mundo de aplicações totalmente novo aos usuários, com novas facilidades e tecnologias.

5.2 Trabalhos Futuros

O sistema DWAM foca na convergência de serviços voltados para dispositivos móveis. Porém, existem funcionalidades que foram simplificadas devido à duração do projeto. Assim, listamos algumas dessas funcionalidades para que futuros estudos possam aperfeiçoar o sistema DWAM. Acreditamos que alguns dos trabalhos futuros sugeridos possuem complexidade tal que podem ser utilizados até mesmo como outros projetos de formatura.

- **Manutenção de sessão:** no sistema DWAM, se um usuário está se locomovendo entre os domínios, sua conexão é perdida. Assim, para voltar a utilizar esses serviços ele deve, novamente, se conectar ao sistema. Com isso, se o usuário estiver, por exemplo, enviando um e-mail, a sessão é perdida e os dados não são enviados. Contudo, para que essa funcionalidade seja implementada, é necessário a implementação de uma solução diferente, o que muda bastante a arquitetura física do nosso projeto. Essas modificações resultariam em um projeto de complexidade bastante elevada.
- **Interligação com a rede de telefonia pública:** as sessões de voz do DWAM só podem ser estabelecidas entre dois dispositivos conectados ao sistema. Porém, existe uma tendência bastante forte de a telefonia pública (PSTN) se integrar à rede de dados, tornando-se uma única rede. Portanto, essa aplicação poderia contribuir bastante no que se refere à oferta de serviços para a nova rede.
- **Correio de Voz e *Unified Messaging*:** o e-mail tem se tornado um dos mais importantes meios de comunicação da era moderna. Com ele, ganhou-se agilidade e precisão, não somente da comunicação informal como também na troca de informações e dados do mundo corporativo. Assim, o *Unified Messaging* seria uma solução convergente entre as funcionalidades da secretária eletrônica e do e-mail. Dessa forma, seria possível o recebimento de mensagens do correio de voz (*Voice Mail*) previamente gravadas em um servidor de correio de voz, na caixa de e-mail do usuário. Para isso seria necessário a

implementação do serviço de Correio de Voz, além da facilidade de *Unified Messaging* no sistema.

Como se pode notar claramente, o grupo defende a idéia de projetos que se utilizem da convergência. Acreditamos que essa tendência vai ganhar forte presença de mercado num curto prazo.

Anexo A – Casos de uso

Caso de uso 1: Autenticação de Usuário no Sistema

Descrição: Este caso de uso descreve a autenticação de um usuário no sistema.

Atores: Usuário do sistema

Pré-condição: Usuário cadastrado no sistema

Seqüência de eventos:

1. O usuário digita seu *login* e senha;
2. O sistema faz a verificação das informações;
3. O usuário entra no sistema.

Pós-condição: Usuário autenticado no sistema.

Extensões:

1. *Login* e/ou senha incorreta(s): sistema exibe mensagem de erro. (Passo 2)
-

Caso de uso 2: Manipular Usuário

Caso de uso 2.1: Cadastro de Usuário

Descrição: Este caso de uso descreve o cadastro de um usuário no sistema.

Atores: Administrador do sistema

Pré-condição: Administrador autenticado no sistema

Seqüência de eventos:

1. O administrador clica em *Adicionar Usuário*;
2. O sistema exibe formulário de cadastro de usuário;
3. O administrador preenche os campos do formulário com as informações básicas do usuário e confirma a adição de usuário;
4. O sistema grava as informações no banco de dados e exibe mensagem de confirmação.

Pós-condição: Novo usuário cadastrado no sistema.

Extensões:

1. Dados do usuário inconsistentes: sistema exibe mensagem de erro. (Passo 4)
 2. Usuário já existente no sistema: sistema exibe mensagem de erro. (Passo 4)
-

Caso de uso 2.2: Remoção de Usuário

Descrição: Este caso de uso descreve a remoção de um usuário do sistema.

Atores: Administrador do sistema

Pré-condição: Administrador autenticado no sistema.

Seqüência de eventos:

1. O administrador clica em *Buscar Usuário*;
2. O sistema exibe formulário de busca de usuário;
3. O administrador preenche com o usuário a ser removido e executa a busca;
4. O sistema exibe o usuário a ser removido;
5. O administrador confirma a remoção;
6. O sistema remove as informações no banco e exibe mensagem de confirmação.

Pós-condição: Cadastro de usuário removido.

Inclusão: Busca de Usuário (Passos 1)

Caso de uso 2.3: Alteração de Usuário

Descrição: Este caso de uso descreve a alteração das informações de um usuário do sistema.

Atores: Administrador do sistema

Pré-condição: Administrador autenticado no sistema.

Seqüência de eventos:

1. O administrador clica em *Buscar Usuário*;
2. O sistema exibe formulário de busca de usuário;
3. O administrador preenche com o usuário a ser alterado e executa a busca;
4. O sistema exibe as informações do usuário a ser alterado;

5. O administrador edita as informações necessárias e confirma a alteração das informações;
6. O sistema atualiza as informações no banco e exibe mensagem de confirmação.

Pós-condição: Cadastro do usuário alterado.

Inclusão: Busca de Usuário (Passos 1)

Caso de uso 2.4: Busca de Usuário

Descrição: Este caso de uso descreve a busca de um usuário no sistema.

Atores: Administrador do sistema

Pré-condição: Administrador autenticado no sistema.

Seqüência de eventos:

1. O administrador preenche os dados do usuário a ser localizado e executa a busca;
2. Sistema exibe tela com informações do usuário encontrado.

Pós-condição: Usuário encontrado.

Extensões:

1. Usuário não encontrado: sistema exibe mensagem de erro. (Passo 1)
-

Caso de uso 3: Manipular Grupo de Usuário

Caso de uso 3.1: Cadastro de Grupo de Usuários

Descrição: Este caso de uso descreve o cadastro de um grupo de usuário no sistema.

Atores: Usuário do sistema

Pré-condição: Usuário autenticado no sistema

Seqüência de eventos:

1. O administrador clica em *Criar Grupo de Usuário*;
2. O sistema exibe um formulário de grupo de usuário;

3. O administrador preenche os campos com as informações básicas de um grupo de usuário;
4. O administrador busca os usuários a serem adicionados no grupo de usuário;
5. O administrador efetua o cadastro do grupo de usuário;
6. O sistema grava as informações no banco e exibe mensagem de confirmação.

Pós-condição: Novo grupo de usuários criado.

Extensões:

1. Dados do grupo de usuários inconsistentes: sistema exibe mensagem de erro. (Passo 5)
 2. Grupo de usuários já existente: sistema exibe mensagem de erro. (Passo 5)
-

Caso de uso 3.2: Remoção de Grupo de Usuário

Descrição: Este caso de uso descreve a remoção de um grupo de usuário do sistema.

Atores: Usuário do sistema

Pré-condição: Usuário autenticado no sistema

Seqüência de eventos:

1. O administrador seleciona *Buscar Grupo de Usuários*;
2. O sistema retorna formulário de busca de grupo de usuários;
3. O administrador preenche com o grupo de usuários a ser removido e executa a busca;
4. O sistema exibe o grupo de usuários a ser removido;
5. O administrador efetua a remoção do grupo de usuários;
6. O sistema remove as informações no banco e exibe mensagem de confirmação.

Pós-condição: Grupo de usuários removido.

Extensões:

1. Grupo de usuário não encontrado: sistema exibe mensagem de erro. (Passo 1)

Inclusão: Busca de Grupo de Usuários (Passo 1)

Caso de uso 3.3: Alteração de Grupo de Usuário

Descrição: Este caso de uso descreve a alteração das informações de um grupo de usuário do sistema.

Atores: Usuário do sistema

Pré-condição: Usuário autenticado no sistema

Seqüência de eventos:

1. O administrador clica em *Buscar Grupo de Usuários*;
2. O sistema retorna formulário de busca de grupo de usuários;
3. O administrador preenche com o grupo de usuários a ser alterado e executa a busca;
4. O sistema exibe as informações do grupo de usuários a ser alterado;
5. O administrador edita as informações necessárias e confirma a alteração das informações;
6. O sistema atualiza as informações no banco e exibe mensagem de confirmação.

Pós-condição: Grupo de usuários alterado.

Extensões:

1. Grupo de usuário não encontrado: sistema exibe mensagem de erro. (Passo 1)

Inclusão: Busca de Grupo de Usuários (Passo 1)

Caso de uso 3.4: Busca de Grupo de Usuários

Descrição: Este caso de uso descreve a busca de um grupo de usuários no sistema.

Atores: Usuário do sistema

Pré-condição: Usuário autenticado no sistema.

Seqüência de eventos:

1. O administrador preenche os dados do grupo de usuário a ser localizado e executa a busca;
2. Sistema exibe tela com o grupo de usuários encontrado;

Pós-condição: Grupo de usuário encontrado.

Extensões:

1. Grupo de usuário não encontrado: Sistema exibe mensagem de erro. (Passo 1)
-

Caso de uso 4: Manipular Localidade

Caso de uso 4.1: Cadastro de Localidade

Descrição: Este caso de uso descreve o cadastro de uma localidade no sistema.

Atores: Administrador do sistema

Pré-condição: Administrador autenticado no sistema.

Seqüência de eventos:

1. O administrador seleciona em *Criar Localidade*;
2. O sistema exibe formulário de cadastro de localidade;
3. O administrador preenche os campos com as informações básicas da localidade a ser criada:
4. O administrador seleciona as restrições a serem adicionadas na localidade;
5. O administrador efetua o cadastro de localidade;
6. O sistema grava as informações no banco e exibe mensagem de confirmação.

Pós-condição: Nova localidade criada.

Extensões:

1. Dados da localidade inconsistentes: sistema exibe mensagem de erro. (Passo 5)
 2. Localidade já existente: sistema exibe mensagem de erro. (Passo 5)
-

Caso de uso 4.2: Remoção da localidade

Descrição: Este caso de uso descreve a remoção de uma localidade do sistema.

Atores: Administrador do sistema

Pré-condição: Administrador autenticado no sistema.

Seqüência de eventos:

1. O administrador seleciona *Buscar localidade*;
2. O sistema exibe formulário de busca de localidade;
3. O administrador preenche com a localidade a ser removida e executa a busca;
4. O sistema exibe a localidade a ser removida;
5. O administrador efetua a remoção da localidade;
6. O sistema remove as informações no banco e exibe mensagem de confirmação.

Pós-condição: Localidade removida.

Extensões:

1. Localidade não encontrada: sistema exibe mensagem de erro. (Passo 3)

Inclusão: Busca de Localidade (Passo 1)

Caso de uso 4.3: Alteração de Localidade

Descrição: Este caso de uso descreve a alteração das informações de uma localidade do sistema.

Atores: Administrador do sistema ou Administrador de localidade

Pré-condição: Administrador autenticado no sistema.

Seqüência de eventos:

1. O administrador clica em *Buscar Localidade*;
2. O sistema exibe formulário de busca de localidade;
3. O administrador preenche com a localidade a ser alterada e executa a busca;
4. O sistema exibe as informações da localidade a ser alterada;
5. O administrador edita as informações necessárias e confirma a alteração das informações;
6. O administrador edita as informações necessárias e confirma a edição das informações;
7. O sistema atualiza as informações no banco e exibe mensagem de confirmação.

Pós-condição: Propriedades da localidade editada.

Extensões:

1. Localidade não encontrada: sistema exibe mensagem de erro. (Passo 3)

Inclusão: Busca de localidade (Passo 1)

Caso de uso 4.4: Busca de localidade

Descrição: Este caso de uso descreve a busca de uma localidade no sistema.

Atores: Administrador do sistema

Pré-condição: Administrador autenticado no sistema

Seqüência de eventos:

1. O administrador preenche os dados da localidade a ser localizada;
2. Sistema exibe tela com as localidades encontradas;
3. O administrador seleciona uma localidade e confirma seleção;
4. Sistema exibe tela com informações da localidade.

Pós-condição: Localidade encontrada, sistema exibe tela com informações da localidade para o administrador.

Extensões:

1. Localidade não encontrada: Sistema exibe mensagem de erro. (Passo 3)
-

Caso de uso 5: Enviar Mensagem Instantânea

Descrição: Este caso de uso descreve o processo de envio de mensagens instantâneas entre dois usuários, estabelecendo uma conversa.

Evento iniciador: Seleção do *usuário receptor* com o qual estabelecerá a troca de mensagens.

Atores: Usuário do sistema (iniciador)

Usuário do sistema (a ser receptor)

Pré-condição: Ambos os usuários autenticados no sistema.

Seqüência de eventos:

1. O *usuário iniciador* seleciona o *usuário a ser receptor* com quem estabelecerá a conversa e abre uma sessão;
2. O *usuário iniciador* digita uma mensagem e envia ao *usuário a ser receptor*.
3. O *usuário a ser receptor* recebe a mensagem e digita uma outra mensagem e a envia ao *usuário iniciador*.
4. A troca de mensagens se repete até que um dos usuários encerre a conversa.

Pós-condição: Conversa entre usuários encerrada.

Extensões:

1. Usuário é desconectado do sistema durante a conversa: o sistema notifica que não é possível enviar mensagens ao destinatário e encerra a conversa. (Passo 3 ou 4)

Caso de uso 6: Fazer Chamada de Voz

Descrição: Este caso de uso descreve o processo de fazer uma chamada de voz para outro usuário do sistema.

Evento iniciador: Seleção do usuário com o qual estabelecerá a conversa de voz.

Atores: Usuário do sistema (iniciador)

Usuário do sistema (receptor)

Pré-condição: Ambos os usuários autenticados no sistema.

Seqüência de eventos:

1. O *usuário iniciador* seleciona o *usuário (receptor)* com quem estabelecerá a conversa e seleciona a opção *Fazer Chamada de Voz*.
2. O *usuário iniciador* aguarda até que o *usuário receptor* atenda a chamada.
3. O *usuário receptor* atende a chamada e uma conversa de voz é estabelecida.
4. Quando a conversa chegar ao fim, um dos usuários encerra a chamada.

Pós-condição: Chamada encerrada.

Extensões:

1. *Usuário receptor* não atende a chamada: Após um intervalo de tempo de *timeout*, o sistema exibe uma mensagem de erro e encerra a chamada. (Passo 3)

Inclusões:

1. Receber Chamada de Voz (Passo 2)
-

Caso de uso 7: Receber Chamada de Voz

Descrição: Este caso de uso descreve a recepção de uma chamada de voz e o estabelecimento de uma conversa de voz.

Evento iniciador: Sinal sonoro e visual avisando que há uma chamada de voz para o *usuário receptor*.

Atores: Usuário do sistema (iniciador)

Usuário do sistema (receptor)

Pré-condição: Ambos os usuários autenticados no sistema. O *usuário iniciador* deve estar fazendo uma chamada para o *usuário receptor*.

Seqüência de eventos:

1. O sistema notifica o *usuário receptor* que há uma chamada através de um sinal sonoro ou visual.
2. O *usuário receptor* atende a chamada e uma conversa de voz é estabelecida.
3. Quando a conversa chegar ao fim, um dos usuários encerra a chamada.

Pós-condição: Chamada encerrada.

Extensões:

1. *Usuário receptor* não atende a chamada: Após um intervalo de tempo de *timeout*, o sistema exibe uma mensagem de erro e encerra a chamada. (Passo 2)
-

Caso de uso 8: Disponibilizar Documento

Descrição: Este caso de uso descreve o processo de disponibilização de documentos para outros usuários do sistema.

Evento iniciador: Usuário seleciona opção Disponibilizar Documento.

Atores: Usuário do sistema

Pré-condição: Usuário autenticado no sistema.

Seqüência de eventos:

1. O usuário seleciona o documento que deseja disponibilizar.
2. O usuário seleciona a localidade onde o documento estará disponível.
3. O usuário seleciona os grupos de usuário que terão acesso ao documento.
4. O usuário seleciona o período de tempo em que o documento estará disponível.
5. O usuário realiza o *upload* do documento para o sistema.

Pós-condição: Documento disponibilizado no sistema.

Caso de uso 9: Download de Documento

Descrição: Este caso de uso descreve o processo de *download* de documentos disponíveis em uma localidade.

Evento iniciador: Usuário seleciona para *download* um documento disponível na localidade onde se encontra.

Atores: Usuário do sistema

Pré-condição: Usuário autenticado no sistema. O usuário deve estar na localidade onde o documento estiver, no horário disponível e pertencer a um grupo de usuário com permissão de acesso ao documento.

Seqüência de eventos:

1. O usuário seleciona o documento que deseja realizar *download*.
2. O sistema envia o documento para o usuário, que o salva em seu dispositivo móvel.

Pós-condição: Documento salvo no dispositivo móvel do usuário.

Caso de uso 10: Manipular Lista de Contatos

Caso de uso 10.1: Adicionar Usuário à Lista de Contatos

Descrição: Este caso de uso descreve a adição de outro usuário do sistema à lista de contatos de um usuário.

Evento iniciador: *Usuário iniciador* seleciona outro usuário para adicionar à lista de contatos.

Atores: Usuário do sistema (iniciador)

Usuário do sistema (a ser adicionado)

Pré-condição: Usuário iniciador autenticado no sistema.

Seqüência de eventos:

1. O *usuário iniciador* clica em *Adicionar Usuário à Lista de Contatos*;
2. O sistema exibe formulário de adição de usuário;
3. O *usuário iniciador* preenche com o *usuário a ser adicionado* a sua lista de usuários do sistema e executa adição.
4. O sistema requisita a autorização do *usuário a ser adicionado* para adicioná-lo a lista de contatos do *usuário iniciador*.
5. O *usuário a ser adicionado* autoriza o sistema a adicioná-lo e o contato é adicionado à lista do *usuário iniciador*.

Pós-condição: *Usuário a ser adicionado* é adicionado à lista de contatos do *usuário iniciador*.

Extensões:

1. *Usuário a ser adicionado* não autoriza a adição: o sistema avisa o *usuário iniciador* e cancela a adição. (Passo 4)

Inclusões:

1. Autorizar Adição à Lista de Contatos. (Passo 3)
-

Caso de uso 10.2: Autorizar Adição à Lista de Contatos

Descrição: Este caso de uso descreve o processo de um usuário autorizar outro a adicioná-lo à sua lista de contato.

Evento iniciador: *Usuário iniciador* seleciona o *usuário a ser adicionado* para adicionar à lista de contatos.

Atores: Usuário do sistema (iniciador)

Usuário do sistema (a ser adicionado)

Pré-condição: *Usuário a ser adicionado* autenticado no sistema. E *usuário iniciador* ter efetuado uma requisição de autorização ao usuário adicionado.

Seqüência de eventos:

1. O sistema notifica o *usuário a ser adicionado* que o usuário iniciador deseja adicioná-lo a sua lista de contatos.
2. O *usuário a ser adicionado* autoriza a adição à lista de contatos.
3. O *usuário iniciador* é adicionado a sua lista do *usuário a ser adicionado*.

Pós-condição: *Usuário a ser adicionado* adiciona o *usuário iniciador* a sua lista de contatos e avisa autorizar o *usuário iniciador* a adicioná-lo.

Extensões:

1. *Usuário a ser adicionado* não autoriza a adição: o sistema avisa o *usuário iniciador* e cancela a adição.

Caso de uso 10.3: Remover Usuário da Lista de Contatos

Descrição: Este caso de uso descreve o processo de remoção de um usuário da lista de contato.

Evento iniciador: *Usuário iniciador* seleciona o usuário a remover.

Atores: Usuário do sistema (iniciador)

Usuário do sistema (a ser removido)

Pré-condição: *Usuário iniciador* autenticado no sistema.

Seqüência de eventos:

1. O *usuário iniciador* seleciona em sua lista de contatos o outro usuário e executa a remoção.
2. O sistema remove este usuário da lista de contatos do *usuário iniciador*.

Pós-condição: *Usuário a ser removido* é removido da Lista de Contatos.

Caso de uso 11: Localizar Usuário da Lista de Usuário

Descrição: Este caso de uso descreve a busca (física) de um usuário no sistema, identificando sua localidade.

Atores: Usuário do sistema (iniciador)
Usuário do sistema (a ser localizado)

Pré-condição: Usuário estar autenticado no sistema e se você não está bloqueado

Seqüência de eventos:

1. O *usuário iniciador* seleciona a opção *Localizar Usuário*;
2. Sistema exibe o formulário de localização de usuários;
3. O usuário iniciador preenche com o *usuário a ser localizado* e executa a localização;
4. Sistema busca a localidade atual do usuário no servidor de localização e exibe a informação.

Pós-condição: Localidade do usuário a ser encontrado é exibida.

Referências

- [1] KIYOSAKI, R. T.; LECHTER, S. L. Pai Rico Pai Pobre. 57ª edição. Rio de Janeiro. Editora Campus, 2000. 186 P. Página 105.
- [2] TANENBAUM, A. S. Redes de Computadores. Tradução da quarta edição. Rio de Janeiro. Editora Campus. 2003, 945 P.
- [3] TELECO. Página com informações sobre tecnologias e conceitos de redes de telefonia e de computadores. Disponível em: <http://www.teleco.com.br/tutoriais/tutoriallbs/pagina_1.asp> Acesso em: 20 de julho de 2005.
- [4] CISCO NETACADEMY. Material do curso on line para certificação CCNA (Cisco Certified Network Associate) versão 2.9. “Camada 2: enlace de dados”. Divulgado somente para alunos em: <<http://www.cisco.com/web/learning/netacad/index.html>> Acesso em: 7 de agosto de 2005
- [5] THE ELECTRONICS RESEARCH GROUP (ERG). Endereço do grupo de pesquisa eletrônicas do departamento de engenharia da Universidade de Aberdeen. Disponível em: <<http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html>> Acesso em: 8 de agosto de 2005
- [6] TELECO. Página com informações sobre tecnologias e conceitos de redes de telefonia e de computadores. Disponível em: <http://www.teleco.com.br/tutoriais/tutorialwlanx/pagina_2.asp> Acesso em: 13 de agosto de 2005

[7] TELECO. Página com informações sobre tecnologias e conceitos de redes de telefonia e de computadores. Disponível em:

<http://www.teleco.com.br/tutoriais/tutorialwlan/pagina_3.asp> Acesso em: 13 de agosto de 2005

[8] COMPUTER SCIENCE AT COLUMBIA UNIVERSITY. Site do departamento de Ciência da Computação da Universidade de Columbia. Disponível em <<http://www.cs.columbia.edu/sip>> Acesso em: 3 de setembro de 2005

[9] SILVEIRA, R. M. "Multimídia e Hiperimídia – Voz sobre IP". Trabalho apresentado à alunos de graduação da Escola Politécnica de São Paulo, 2005. Não publicado.

[10] ESTADÃO. Site do Jornal "O Estado de São Paulo". Disponível em: <<http://www.estadao.com.br/tecnologia/colunas/sociedade/2000/jun/19/159.htm>> Acesso em: 13 de novembro de 2005

[11] LEONEL, J.; PEDINI, P.; "Voz sobre IP – A Revolução na Telefonia". Matéria da Revista Promon Business & Technology Review, 2005.

[12] ABE, A. Y.; YANAI, E. R.; ONISHI, F. J. "Portal de Comércio Eletrônico Baseado em Dispositivos Móveis e na Personalização da IHC: Sistema TANAMÃO". 2004, 105p. Dissertação (Graduação) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2004.

[13] NSI.COM. Site de um fornecedor de Máquinas Virtuais Java. Disponível em: <<http://www.nsicom.com/Default.aspx?tabid=138>> Acesso em: 7 de maio de 2005

[14] SUN DEVELOPER NETWORK. Endereço da Sun na Internet no qual podemos encontrar informações sobre o JAIN. Disponível em: <<http://www.jcp.org/en/jsr/detail?id=035>> Acesso em: 8 de maio de 2005

[15] SUN DEVELOPER NETWORK. Endereço da Sun na Internet no qual podemos encontrar informações sobre o JMF. Disponível em: <<http://java.sun.com/products/java-media/jmf>> Acesso em: 27 de maio de 2005

[16] UNIVERSIDADE FEDERAL DE PERNAMBUCO. Citação sobre o JMF. Disponível em: <<http://www.di.ufpe.br/~jhcf/TheGraphEditor/ImgMapData/javaapis.gif.data.19.html>> Acesso em: 27 de maio de 2005

