

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

MURILO GARCIA BENTO

APLICAÇÃO VOIP POR MEIO DA TECNOLOGIA WIRELESS

CRICIÚMA, DEZEMBRO DE 2010

MURILO GARCIA BENTO

APLICAÇÃO VOIP POR MEIO DA TECNOLOGIA WIRELESS

Trabalho de Conclusão de Curso,
apresentado para obtenção de Grau de
Bacharel em Ciência da Computação da
Universidade do Extremo Sul Catarinense.

Orientador: Prof. MSc. Paulo João

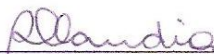
Martins

CRICIÚMA, DEZEMBRO DE 2010

MURILO GARCIA BENTO

Aplicação VoIP por meio da tecnologia Wireless

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

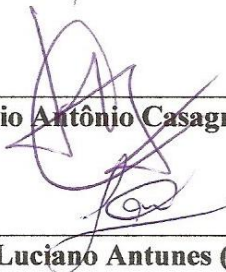


Prof. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:



Prof. MSc. Paulo João Martins (UNESC)
Orientador



Prof. MSc. Rogério Antônio Casagrande (UNESC)

Prof. Esp. Luciano Antunes (UNESC)

Aos meus pais, minha família, amigos,
e a todos que me apoiaram e estiveram
ao meu lado em todo o período de estudo.

AGRADECIMENTOS

Agradeço primeiramente a Deus.

Aos meus pais por acreditarem e mim e fornecer toda a estrutura necessária para meu desenvolvimento, dando-me apoio necessário sempre que precisei.

Aos professores, pela paciência e dedicação na explicação da matéria, passando assim um pouco de seus conhecimentos ao longo de toda a trajetória percorrida.

Aos meus amigos por ajudarem quando precisei, pelos momentos inesquecíveis que passamos neste curso.

A empresa Ema Software por ter me proporcionado apoio e as mais diversas oportunidades de aprendizagem.

O único homem que nunca comete erros é aquele que nunca faz coisa alguma. Não tenha medo de errar, pois você aprenderá a não cometer duas vezes o mesmo erro.

(Theodore Roosevelt)

RESUMO

VoIP está ganhando destaque atualmente, um prova disso são os *smartphones* mais atuais que já estão vindo com suporte a tal recurso, porém, sua configuração ainda é toda manual exigindo um certo grau de conhecimento do usuário. Baseado nestes fundamentos é que foi alavancada a idéia de desenvolver um aplicativo de fácil utilização de que tenha a mesma funcionalidade. O aplicativo até então batizado de JVoIP, desenvolvido em Java para dispositivos móveis utilizando o protocolo *Datagram* em conjunto com as bibliotecas nativas de multimídias, consegue estabelecer uma conexão entre dois dispositivos por meio de um ponto de acesso sem fio, evitando principalmente gastos com linha telefônica e criando assim uma alternativa de fácil utilização e a baixo custo.

Palavras-chaves: VoIP; *Wireless*; J2ME; Dispositivo móvel; Smartphone.

ABSTRACT

VoIP is gaining prominence nowadays, a proof of this is the most current smartphones that already come with support for such an appeal, but its configuration is still all manual requiring a degree of knowledge from its user. Based on these assumptions, emerged the idea to develop an easy to use application that has the same functionality. The application previously named JVoIP, developed in Java for mobile devices using the Datagram protocol in conjunction with the native libraries of multimedia, can establish a connection between two devices via a wireless access point, avoiding spending mainly with telephone line and thus creating an easy-use and low cost alternative.

Keywords: VoIP; Wireless; J2ME; Mobile device; Smartphone.

LISTA DE ILUSTRAÇÕES

Figura 1. Exemplo de rede em células na cidade de Criciúma, dados fictícios.....	26
Figura 2. A esquerda o Nokia Mobira Cityman 900 de 1987 e a direita o novo Nokia	29
Figura 3. Arquitetura Java.	42
Figura 4. Arquitetura J2ME.....	44
Figura 5. Comunicação SIP.....	51
Figura 6. Exemplo de MIDlet.....	59
Figura 7. Classe OlaForm, tela que será visualizada pelo usuário.	60
Figura 8. MIDlet rodando no emulador.....	61
Figura 9. Criando um servidor <i>Datagram</i> para mensagens de texto.....	63
Figura 10. Lado cliente do aplicativo.	64
Figura 11. Respectivamente, lista de contatos no <i>sotryboard</i> e como	66
Figura 12. Tela principal do aplicativo.....	67
Figura 13. Início do processo de gravação do áudio.	68
Figura 14. Fim do processo de gravação de áudio.	70
Figura 15. Aplicativo VoIP rodando em um N95.	75

LISTA DE TABELAS

Tabela 1 – Análise de custos.	71
Tabela 2 – Comparativo entre o aplicativo Skype e VoIP.....	75

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CDC	Connected Device Configuration
CDMA	Code Division Multiple Access
CLDC	Connected Limited Device Configuration
EDGE	Enhanced Data rates for Global Evolution
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU	International Telecommunications Union
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
JRE	Java Runtime Environment
JVM	Java Virtual Machine
KVM	K Virtual Machine

MID	Mobile Information Device
MMS	Multimedia Message Service
MP	Módulo Processadore
P2P	Peer-To-Peer
PDA	Personal Digital Assistants
PTT	Push-To-Talk
RMS	Record Management System
SDK	Software Development Kit
SIP	Session Initiation Protocol
SMS	Short Message Service
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UMTS	Universal Mobile Telecommunication System
USB	Universal Serial Bus
VoIP	Voice over IP
WAP	Wireless Application Protocol
Wi-Fi	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVO GERAL.....	18
1.2 OBJETIVOS ESPECÍFICOS	18
1.3 JUSTIFICATIVA	19
1.4 ESTRUTURA DO TRABALHO	22
2 COMUNICAÇÃO	24
2.1 SISTEMA TELEFÔNICO	24
2.2 COMUNICAÇÃO MÓVEL.....	25
2.3 EVOLUÇÃO DOS APARELHOS CELULARES	28
2.4 MODELOS DE COMUNICAÇÃO MÓVEL EXISTENTES	30
2.4.1 TDMA	32
2.4.2 CDMA.....	32
2.4.3 GSM	33
2.5 SERVIÇOS.....	34
2.5.1 Short Message Service.....	35
2.5.2 Wireless Access Protocol.....	36
2.5.3 General Packet Radio Service	37
2.5.4 Enhanced Data for GSM Evolution	38
2.5.5 Multimedia Messaging Service.....	39
2.5.6 Universal Mobile Telecommunication System.....	39
3 TECNOLOGIAS DE DESENVOLVIMENTO	41

3.1 JAVA	41
3.2 JAVA2 MICRO EDITION.....	44
4 REDES DE COMUNICAÇÃO	47
4.1 VOZ SOBRE IP	47
4.2 REDES SEM FIO – WIRELESS	49
4.3 PROTOCOLO SIP	50
5 TRABALHOS CORRELATOS	53
5.1 COMUNICAÇÃO ENTRE DISPOSITIVOS DE BAIXO PODER COMPUTACIONAL: BLUETOOTH COMO PROTOCOLO DE COMUNICAÇÃO PARA JOGOS PARA CELULAR.....	53
5.2 VOIP PARA COMPUTAÇÃO MÓVEL.....	53
6 APLICATIVO MÓVEL PARA COMUNICAÇÃO DE VOZ	55
6.1 METODOLOGIA.....	55
6.2 ETAPAS DO DESENVOLVIMENTO.....	57
6.2.1 A linguagem de programação.....	58
6.2.2 Funcionamento da comunicação	61
6.2.2.1 <i>Datagram</i>	62
6.2.3 Framework de persistência – Floggy	64
6.2.4 A estrutura do aplicativo	65
6.2.5 Gravação e execução de áudio.....	66
6.3 ANÁLISE DE CUSTOS	71
6.4 RESULTADOS OBTIDOS	72

CONCLUSÃO.....	76
REFERÊNCIAS	80
REFERÊNCIAS COMPLEMENTARES	83
APÊNDICE A – DIAGRAMA DE CLASSES	86
APÊNDICE B – DIAGRAMA DE ATIVIDADE DE CONEXÃO	88
APÊNDICE C – ARTIGO	99
ANEXO A – ATIVIDADES REALIZADAS PELO TELEFONE CELULAR	98
ANEXO B – PROPORÇÃO DE INDIVÍDUOS QUE POSSUEM TELEFONE CELULAR COM ACESSO À INTERNET	99
ANEXO C – CONFIGURANDO O NETBEANS PARA USAR FLOGGY	100

1 INTRODUÇÃO

A comunicação é um dos fatores mais fortes no marco do desenvolvimento da sociedade, e por vez, é o objeto fundamental para o relacionamento interdependente entre sistemas, tendo em vista que grandes empresas do ramo têm suas tecnologias voltadas para esta área. Neste sentido, algumas tecnologias aumentam uma série de vantagens e facilidades nativas de sua abordagem, como é o caso do telefone celular, um excelente aparelho tecnológico para conversação que permite realizar ligações para qualquer lugar do mundo (desde que a operadora cubra a região desejada), com rapidez e facilidade.

Segundo Colcher, *et al.* (2005) o processo de convergência das tecnologias de telecomunicações e processamento de informações (via sistemas computacionais) tem promovido profundas alterações na organização do trabalho na indústria, no comércio, na prestação de serviços, nas pesquisas, na vida particular do cidadão, na saúde e na educação. Dentre os maiores representantes desses processos figuram os serviços multimídia, hoje presentes em dispositivos móveis e celulares, e a proliferação de serviços variados sobre a infraestrutura de redes de computadores como a Internet.

Porém, mesmo com a expansão do mercado consumidor e a sua gradativa evolução, ainda existem custos de pesquisas, testes e divulgação, além de muitas outras variantes existentes no processo de desenvolvimento de novas

tecnologias. A somatória de todos estes custos gera valores de comércio, muitas vezes, inacessíveis a maioria das pessoas.

O telefone celular, mesmo que adquirido a um baixo custo, ainda há a necessidade de aderir a um chip da operadora desejada e um plano de tarifa. Atualmente, o plano mais comum é o chamado pré-pago, onde se paga as ligações antes de utilizá-las, sendo considerado o meio mais barato e econômico para quem realiza poucas ligações, também há os planos pré-definidos e personalizados de cada operadora para quem realiza ligações em quantidades medianas ou exorbitantes.

De forma geral, é possível dizer que existem muitos casos onde ocorre a comunicação a curtas distâncias e baixo período de tempo, porém, independentemente da distância ou do tempo necessário para transmitir a informação, haverá um custo associado a este canal de comunicação para o cliente final. Para suprir este problema de custo, particularmente em transmissões de informações curtas e rápidas, as operadoras oferecem o SMS, serviço de mensagem curta, onde é possível encaminhar ao destinatário a informação desejada por meio de texto, o qual é acessível em qualquer aparelho celular que disponibilize este recurso, e com custo relativamente mais baixo que uma ligação; e como este recurso se tornou mais acessível ao público, é notável que possua limitações, como quantidade de caracteres.

Além do mais existe a opção de aderir a um plano de dados com a operadora, sendo que este permite (caso o aparelho celular suporte) a navegação

na Internet, incluindo o envio e o recebimento de e-mails, e para dispositivos mais modernos, a utilização do *Voice over IP* (VoIP), ou voz sobre IP, por meio de algum aplicativo que funciona pela *web*, como o *Skype*®, porém a aquisição deste plano também está ligada a pagamentos mensais e/ou tarifação de uso, sendo que os valores também variam por plano/operadora.

Baseado neste contexto esta pesquisa visa reunir a utilização de recursos existentes às necessidades do cotidiano. Um recurso que vem ganhando mercado e a cada dia se tornando mais comum nos dispositivos móveis é o *Wi-Fi*, um protocolo de comunicação sem fio simples que pode ser encontrado em vários modelos de aparelhos celulares e *smartphones*. Outra tecnologia que também está se desenvolvendo de forma rápida e *agressiva* é o J2ME, tecnologia Java para dispositivos móveis. Em virtude de sua simplicidade e semelhanças com o J2SE e principalmente sua portabilidade, é cada vez mais comum encontrar aparelhos que suportem esta tecnologia.

A união destas tecnologias deverá resultar na implementação de um aplicativo para dispositivos móveis com suporte a J2ME, que tem como principal função permitir a comunicação a um custo mais acessível às pessoas via VoIP por meio de *Wireless* quando houver um ponto de acesso a Internet sem fio dentro do alcance.

Como diz Cliconnect.com (2009) uma das principais vantagens da implantação de uma rede VoIP em uma rede corporativa é a possível diminuição de custos com ligações telefônicas locais, interurbanas e até mesmo

internacionais. Tal implantação, em alguns casos, é bastante simples e torna-se muito mais vantajosa caso a empresa já possua uma rede de dados. O uso de um protocolo padrão já bastante conhecido e utilizado no mundo todo como o IP só faz aumentar a confiança de utilizá-lo também para a transmissão da voz humana.

1.1 OBJETIVO GERAL

Desenvolver uma aplicação VoIP para dispositivos móveis em cima do protocolo de redes sem fio *Wireless*.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos são:

- a) analisar e utilizar os conceitos de VoIP;
- b) estudar conexão dos dispositivos móveis por meio de *Wireless*;
- c) analisar redução de custo na realização de uma conexão local por meio da tecnologia *Wireless* já existente nos dispositivos móveis, aproveitando pontos de acesso sem fio disponíveis;
- d) estudar possibilidade de realizar conexão ponto-a-ponto entre dispositivos móveis;
- e) utilizar tecnologia J2ME para a implementação da aplicação.

1.3 JUSTIFICATIVA

No dia-a-dia, a utilização da tecnologia é rotineira e até invisível à percepção de quem já está acostumado. Usar um telefone celular para executar uma ligação para o trabalho ou falar com um familiar, acessar a Internet e pagar as contas, enviar SMS para amigos, não só estão se tornando cada vez mais comum, como também estão atingindo faixas etárias mais diversificadas. Conforme IBGE (2007) mais de seis milhões de jovens entre 10 e 14 e idosos com 60 ou mais anos de idade no Brasil já tem acesso a um aparelho celular para uso pessoal. Isso demonstra o quão acessível estes aparelhos estão ficando a cada dia às pessoas.

Segundo Muchow (2004) os telefones móveis modificam as nossas vidas, pois nos permitem comunicar independente do local que estamos, seja em casa, no escritório ou longe deles. Também é possível navegar na Internet ou executar aplicativos diversos com os assistentes digitais pessoais - Personal Digital Assistants (PDA).

O Brasil ainda tem os custos de ligações e envio de SMS mais caros da América Latina, podendo chegar a 80% a mais, quando comparado ao valor de um país vizinho. Por exemplo, no estado de Santa Catarina, para enviar um SMS, que atualmente é o recurso mais acessível do celular e amplamente utilizado por estudantes segundo estudos recentes, pela operadora Vivo para um celular de outra operadora o custo é de R\$ 0,35 para o plano pós-pago e R\$ 0,36 para o plano pré-pago. Pela operadora Claro, o custo reduz para R\$ 0,30 para ambos os

casos. Em contrapartida para a operadora TIM, o custo é de R\$ 0,39 em ambos os casos. Já os valores das ligações têm uma variação muito maior, pois, depende do estado/operadora origem e estado/operadora destino da ligação, porém os menores preços podem ser encontrados nos planos pós-pagos, mas mesmo com o valor da ligação mais acessível, ainda existem taxas mensais que o tornam mais caros que os outros planos, já nos planos pré-pagos os valores das ligações podem chegar ao dobro de uma ligação no plano pós (UCEL, 2009).

Isso demonstra a necessidade do desenvolvimento de novas tecnologias com o objetivo de minimizar os custos gerais. Atualmente, existe a tecnologia VoIP, que, consiste em digitalizar e codificar a voz em bits, criando assim pacotes de dados que são enviados por meio de um protocolo de rede como UDP, TCP e IP, onde a comunicação é feita pela Internet. Nela a voz é convertida em sinais digitais e enviada ao receptor responsável pela transformação em sinais analógicos. Mas para poder utilizar destes recursos em um aparelho celular ou *smartphone*, há a necessidade de se adquirir um plano de dados com a operadora, o que cria mais um custo em cima de outros como a aquisição do *chip*, aparelho e a tarifa do plano escolhido. O valor dos dados trafegados varia de acordo com a operadora e o plano, podendo-se inclusive aderir a um plano ilimitado, onde é pago uma taxa mensal fixa e se pode usar livremente. Assim é retornado ao problema do alto custo de utilização, onde poucos têm acesso e a gastos desnecessários que podem ser supridos com a utilização de recursos muitas vezes já disponíveis.

Perante a todas essas necessidades, é que surgiu a idéia de desenvolver um aplicativo para dispositivos móveis que permita realizar transmissão de voz a baixo custo. Com a utilização da tecnologia Java para dispositivos móveis, é possível, a partir de um único código fonte desenvolver um aplicativo flexível a um baixo custo de manutenção e com um alto grau de portabilidade direcionado a dispositivos com pouco poder computacional. Conforme Muchow (2004) J2ME é destinado diretamente aos dispositivos consumidores com poder limitado. Muitos destes dispositivos (por exemplo, celular ou pager) não têm a opção de *download*¹ e *software*² de instalação, além daquele que já foi configurado durante o processo de fabricação. Com a introdução desta tecnologia, os dispositivos “micros” não precisam mais ter natureza “estática”.

A portabilidade é uma das principais vantagens da utilização desta tecnologia, pois, torna possível que o aplicativo seja adaptado e executado em diversos tipos de dispositivos, como celulares, PDA, *smartphones*, e sistemas operacionais, como *Palm OS*, *Symbian*, bastando apenas que estes tenham uma máquina virtual K – K Virtual Machine (KVM).

O recurso Wireless, já embutido em muitos aparelhos existentes no mercado atualmente, justifica-se pela possibilidade de criar uma aplicação que possa conectar dois ou mais dispositivos móveis diferentes permitindo que haja a troca de informações, no caso desta pesquisa, voz. Segundo Higdon (2007) as

¹ Download, é o ato de requisitar e transferir algum conteúdo digital de um local remoto a outro.

² Software, é qualquer aplicativo utilizado por um computador.

novas gerações de celulares e PDA estão trazendo a tecnologia *Wi-Fi* padrão IEEE 802.11x para suas arquiteturas. Com estas tecnologias podemos conectar estes aparelhos móveis em uma rede Internet facilmente, ganhando com isso, mobilidade.

Assim, pode haver comunicação entre dispositivos aproveitando as redes sem fio disponíveis, com custo zero (desconsiderando gastos com energia – como carregamento do celular – e aquisição de um aparelho que suporte as tecnologias Java e *Wireless*). Outra grande vantagem do *Wireless* é a sua presença em muitos tipos de aparelhos eletrônicos, possibilitando a expansão para a utilização de outros recursos, como um micro computador.

1.4 ESTRUTURA DO TRABALHO

O primeiro capítulo visa demonstrar de forma geral a realidade do sistema de comunicação atualmente, falando sobre valores cobrados pelas operadoras e planos para aquisição, além de algumas tecnologias em ascensão que serão utilizadas no projeto para o desenvolvimento do aplicativo proposto.

Em seguida, é entrado na questão da comunicação propriamente dita, como surgiu, evoluiu e como funciona atualmente, meios mais utilizados e principalmente a comunicação móvel, tema deste projeto, além de explanar os padrões já criados e suas evoluções e serviços oferecidos.

No capítulo três será visto a tecnologia de desenvolvimento optada para a realização do projeto, falando do por que de sua escolha e qualidades oferecidas e limitações encontradas.

Após a definição da linguagem de programação que será utilizada, é necessário escolher o melhor modelo de transmissão e recepção que será estudado, para isso, o capítulo quatro, fala sobre redes de comunicação, tecnologia sem fio, voz sobre IP e protocolos.

A seguir serão vistos alguns trabalhos existentes na área que utilizam a tecnologia de desenvolvimento escolhida em outras áreas e aplicações.

No capítulo seguinte é explicado e exemplificado o aplicativo desenvolvido, falando de cada uma de suas etapas, dificuldades, soluções e quais decisões tiveram que ser tomadas durante o processo de estudo e desenvolvimento.

Para fechar estão disponíveis as referências bibliográficas utilizadas para o estudo e desenvolvimento além de apêndices que visam facilitar e auxiliar no estudo deste projeto e alguns anexos que foram de vital ajuda no desenvolvimento.

2 COMUNICAÇÃO

Atualmente é uma importante característica da sociedade, a partir dela, por exemplo, podemos saber o país de origem ou cultura de uma determinada pessoa. A existência de sociedades, seja a nível global como um país ou local como uma cidade, se dá em virtude da comunicação, principal ferramenta usada na integração entre seres possibilitando que trabalhem e convivam em conjunto.

Derivado do latim *comunicare*, cujo significado seria “tornar comum”, “repartir, associar”, “trocar opiniões”, “conferenciar”. Porém a comunicação de hoje conhecida, assim como muitas outras atividades passou por um processo evolutivo. Segundo Perles (2007) as primeiras formas de diálogo eram expressas por gestos, imagens e sons, somente mais tarde foi desenvolvido a capacidade de utilizar sinais gráficos, neste ponto houve uma evolução tanto na conversação como na cultura.

A compreensão de comunicação abrange todo o processo, desde o envio, recebimento e entendimento de uma informação.

2.1 SISTEMA TELEFÔNICO

A evolução do sistema telefônico é muito comumente confundida com a evolução do sistema de comunicação, porém, em 1844, Samuel Morse enviara a primeira mensagem utilizando sua criação, um sistema de telegrafia. A

transmissão de sinais elétricos, só veio a acontecer mais tarde em 1875 por Alexander Graham Bell e de forma inesperada. Em testes no seu sistema de telegrafia, Bell descobriu por acaso que era possível enviar e receber de tal forma que se poderiam produzir sons diferentes do esperado, conforme Colcher, *et al.* (2005, p. 2) “[...], ele conseguira, produzir uma corrente elétrica cuja variação acontecia na mesma intensidade que o ar variava de densidade junto ao transmissor.”.

Com isso, após diversas melhorias em 1876, Bell submeteu a patente do chamado telefone, o aparelho capaz de transmitir voz e outros sons.

2.2 COMUNICAÇÃO MÓVEL

A transmissão por rádio frequência é a tendência que mais evoluiu nas últimas décadas e que atualmente tem maior destaque no ramo da comunicação. Utilizado inicialmente para assuntos marítimos e militares, a conversação se dava por meio de um transmissor de alta potência situado em um local alto cobrindo uma ampla região. Este desempenhava o papel de transmissor e receptor ao mesmo tempo dentro de um único canal de envio, conhecido como PTT, pressione para falar, neste sistema, para falar o usuário pressionava um botão para habilitar a o canal de conversação ao mesmo tempo em que o canal de recepção é desabilitado.

O surgimento do conceito de célula mudou esta realidade, trocando de um único transmissor de alta potência para vários outros menores, este modelo de sistema permite melhor qualidade de transmissão/recepção de sinais além de permitir a utilização de mais canais de difusão, ou seja, vários usuários utilizando o mesmo meio ao mesmo tempo, além de permitir a utilização da técnica *handoff*, onde um usuário pode no meio de uma comunicação, sair da área de cobertura de uma célula e ir para outra sem a interrupção da conversação, isso, desde que, estas células estejam adjacentes.

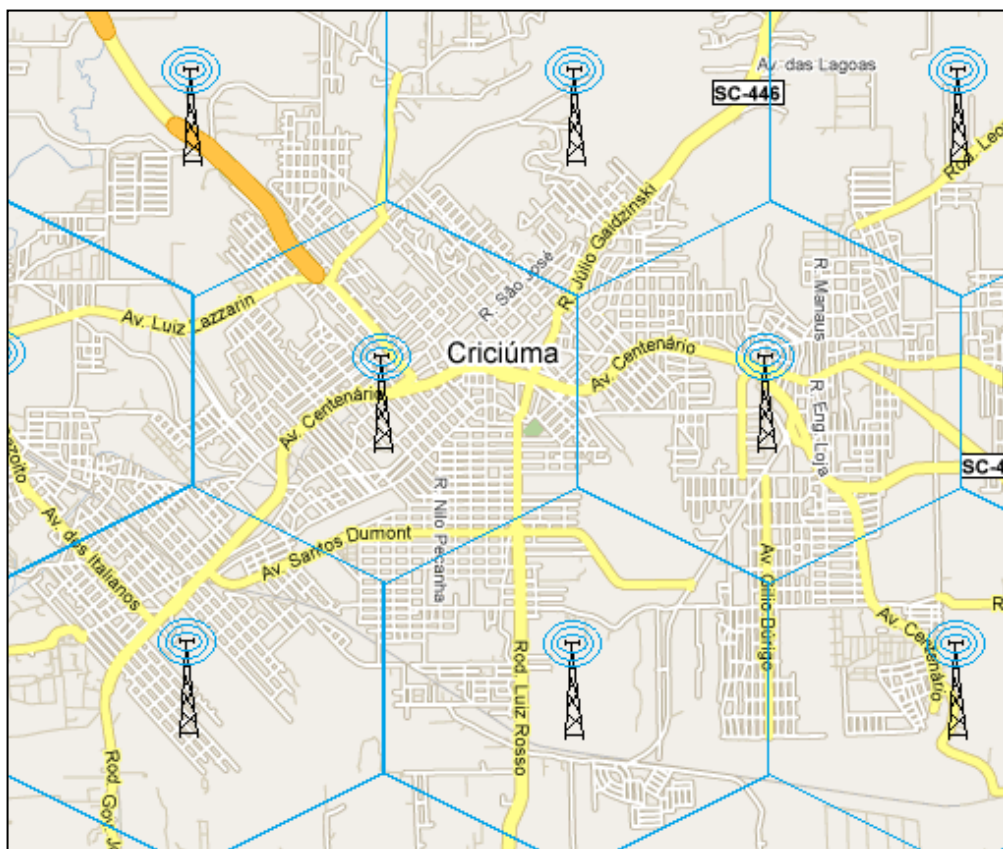


Figura 1. Exemplo de rede em células na cidade de Criciúma, dados fictícios.
Fonte: Adaptado de Google Maps (2010).

A evolução no modelo de comunicação também permitiu a evolução dos aparelhos sem fio. Celular, dispositivo eletrônico muito utilizado atualmente, teve seu nome originado no novo modelo de células (célula → celular). Dentre as diversas causas que permitiram que este aparelho se tornasse algo tão famoso e comum na sociedade atual, é possível citar o inconveniente da troca de endereço ou emprego, onde pode ocasionar na troca do número fixo. Outra característica dos aparelhos celulares é o fato de poder ser levado a qualquer lugar junto ao usuário, considerando que o local tenha cobertura da empresa de telefonia escolhida, é possível fazer e receber ligações onde estiver.

Os preços de uma ligação entre dispositivos móveis também não pode ser desconsiderado. De acordo com Cairncross (2000) no início da década de 90 uma ligação feita de um celular chegava a ser três vezes mais cara que uma ligação feita de uma linha fixa. Porém, no quadro atual, os preços entre os serviços de uma linha fixa e uma linha móvel estão equivalentes, já é notável a diminuição da aquisição de linhas fixas, ao contrário das linhas móveis, em particular com os planos e formas de cobranças hoje disponibilizados pelas operadoras é muito mais viável a grande massa ter um celular pré-pago ou uma conta controle do que uma linha fixa em casa, isso sem considerar o fato de que muita gente passa maior parte fora de casa, deixando assim a linha fixa por muito tempo ociosa.

2.3 EVOLUÇÃO DOS APARELHOS CELULARES

Já faz muito tempo que os aparelhos celulares deixaram de servir apenas para transmitir voz. A evolução em nível de *hardware* e *software* dos aparelhos celulares tem ampliando o seu poder computacional. A gama de serviços oferecidos pelas operadoras telefônicas também tem aumentado. Como diz Gugelmin (2010), nos dias atuais, é comum ouvir que os celulares fazem a mesma coisa que um computador, e de certa forma, realmente fazem. Não apenas os celulares, mas os PDA, ou *smartphones*, possuem o que é chamado de sistema aberto, ou seja, é possível a adição de programas no seu sistema operacional, de acordo com Smartphone Celular (2010) esta característica também é a principal diferença entre ele e um celular comum, uma vez que os sistemas operacionais de celulares normalmente não permitem a adição de outros programas, exceto quando este tiver suporte à tecnologia Java.



Figura 2. A esquerda o Nokia Mobira Cityman 900 de 1987 e a direita o novo Nokia N8 de 2010.

Fonte: Ross (2009) e Nokia (2010).

Essa possibilidade de adição de programas ao aparelho permite que sejam desenvolvidos os mais diversos tipos de aplicativos, permitindo segundo Cairncross (2000) assim que um usuário possa utilizar em seu aparelho, por exemplo, programas como agenda de contatos, jogos, acesso ao banco de dados local para a efetuação de vendas e outros. Um exemplo prático que pode ser citado são as empresas de desenvolvimento de software ter um módulo de PDA, que serve para instalar nos *smartphones* dos representantes de um cliente que adquiriu o sistema para a efetuação de vendas descartando a necessidade de utilizar papel para anotações de vendas, ou notebooks para a consulta de preços de produtos, em um único aparelho móvel de fácil utilização é possível ter um programa que tenha cadastrado toda a pauta de preços do mix de produtos, quantidade em estoque,

tabelas de preços além da possibilidade da efetuação de vendas no próprio dispositivo e posteriormente a sincronização³ com o sistema principal.

2.4 MODELOS DE COMUNICAÇÃO MÓVEL EXISTENTES

Os modelos de comunicação móvel também passaram por diversas evoluções, estas evoluções são classificadas em gerações:

- a) 1G, analógica, foram os primeiros dispositivos disponíveis no mercado, altamente suscetível a falhas de transmissão;
- b) 2G, esta já conta com recursos digitais, melhorando a qualidade dos sinais enviados;
- c) 2,5G, intermediária conta com melhorias significativas no modelo de transmissão melhorando ainda mais a qualidade dos sinais;
- d) 3G, atualmente em adoção no Brasil, já existe em outros países a um bom tempo, segundo Bazotti (2007) a esta geração é dotada de uma alta qualidade de transmissão com interferência mínima, além da capacidade de envio de sinais multimídia como, imagens, músicas e vídeo;
- e) 4G é uma realidade em desenvolvimento e testes em outros países da Europa que trará inovações no sentido de qualidade e principalmente velocidade.

³ Sincronizar, dentro do contexto, pode-se dizer que é determinar a diferenças entre os dados de diferentes dispositivos ou base de dados e juntar as diferenças em uma única informação.

Cada uma das gerações descritas tinha a sua disposição uma gama de serviços compatíveis com o modelo de comunicação, considerando-se velocidade de transmissão e tecnologia utilizada tanto nos terminais (com antenas e satélites), como nos próprios aparelhos. A primeira geração por se tratar de um modelo analógico e muito ultrapassado para os padrões atuais não é mais interessante ao projeto de pesquisa, porém, vale destacar que a já continha o serviço SMS ou serviço de mensagens curtas. Já a segunda trouxe modelos que são utilizados até os dias de hoje, mesmo com a existência das novas tecnologias como a 3G.

Dentre todos os sistemas e serviços de conversação disponíveis na segunda geração, os principais são:

- a) TDMA;
- b) CDMA;
- c) GSM;
- d) SMS;
- e) WAP.

Já na segunda e meia, temos a evolução de alguns sistemas da segunda geração, levando em consideração aspectos como velocidade de transmissão, confiabilidade nos serviços e qualidade. Dentre todos, os mais conhecidos incluem GPRS, EDGE e MMS.

Na terceira geração existe uma grande evolução e difusão dos serviços de dados. A migração para outra geração não é algo simples para as operadoras, porém o forte crescimento dos serviços via *Web*, implicam que as operadoras

ofereçam os serviços que os clientes desejam ter. Não é certo que os celulares irão substituir os computadores convencionais, porém, é uma questão de tempo até que as pessoas utilizem serviços no celular que antes só estavam disponíveis em desktops, como por exemplo, videoconferência. A tecnologia disponível na terceira geração é a UMTS que será discutido mais a frente.

2.4.1 TDMA

Em sua primeira versão em 1990, apresentou um modelo um pouco mais evoluído que seus ancestrais analógicos, na sua segunda, lançada em 1991, foram adicionados alguns serviços importantes, como, autenticação, privacidade de voz e identificador de chamadas.

A principal desvantagem deste sistema conforme Fleury (2004) é a falta de evolução para outros sistemas, logo, não é possível um sistema deste migrar para uma geração 2,5 ou superior.

2.4.2 CDMA

Desenvolvido nos EUA, foi utilizado inicialmente com finalidade militar, suas principais características incluem alta privacidade e facilidade de expansão para outros sistemas mais modernos como o sistema da terceira geração.

De forma geral pode-se dizer que é o sistema mais flexível, pois trabalha com uma faixa de taxa de transmissão de sinais ampla, segundo Bazotti (2007) isso permite que uma operadora possa escolher entre trabalhar com uma baixa taxa de transmissão e conseqüentemente uma maior capacidade do sistema ou então o contrário, uma taxa mais alta e menos capacidade do sistema,

2.4.3 GSM

Na década de 80 na Europa, eram utilizados muitos sistemas de telefonia móvel diferentes, de forma que estes não funcionavam entre si. Mas com o constante crescimento destes, viu-se a necessidade de integrar estes sistemas. Ainda na Europa, no início da década de 90, foi criado um grupo chamado *Groupe Spéciale Mobile*, este grupo foi responsável pela criação do padrão GSM que tem como uma das principais características da integração dos sistemas de comunicação móvel, a chamada interoperabilidade (FLEURY, 2004). Atualmente a sigla é utilizada para representar *Global System for Mobile Communications* ou Sistema Global de Comunicações Móveis.

Neste sistema de comunicação existe uma série de especificações que devem ser seguidas, estas especificações vão desde necessidades comerciais, como a existência de várias operadoras dentro de um país, com o intuito de garantir a competitividade e redução de custos, até usabilidade, como, ter qualidade de voz e um bom nível de segurança nas informações transmitidas. Sua principal desvantagem em

relação ao sistema CDMA é o fato de não permitir expansão nativa ao 3G, para isso, a operadora deverá fazer investimento em mais antenas e aumento da frequência de operação. Porém, em contra partida, de acordo com Fleury (2004) as principais vantagens oferecidas pelo sistema GSM incluem muitos benefícios aos usuários, como:

- a) mais qualidade na transmissão de voz;
- b) roaming internacional, na teoria é possível realizar uma ligação para qualquer país que tenha um sistema GSM;
- c) segurança nas transmissões, a voz é criptografada;
- d) suporte a transmissão de dados;
- e) alto nível de concorrência, conseqüentemente preços mais baixos, porém interoperabilidade entre operadoras.

2.5 SERVIÇOS

Além da comunicação por voz propriamente dita, existem atualmente diversos serviços oferecidos pelas operadoras telefônicas de celular. Estes serviços visam principalmente uma alternativa a difusão por voz, seja por uma questão de custo, rapidez ou lazer. Cada serviço tem suas próprias características e funções bem específicas, além de cobrança diferenciada por operadora, plano adotado e pacotes personalizados ou promocionais adquiridos.

2.5.1 Short Message Service

O primeiro serviço de SMS ou Serviço de Mensagens Curtas foi desenvolvido na Europa em 1991, desde então vem ganhando cada vez mais mercado. É um dos recursos que mais ganhou espaço dentro dos meios de comunicação. Trata-se de um serviço oferecido pelas operadoras de envio e recebimento de pequenos textos (até 160 caracteres no sistema GSM e 255 no CDMA) digitados nos aparelhos de telecomunicação, e não só celulares como costumava ser, atualmente até mesmo em telefones fixos podem ser encontrados dispositivos de envio e recebimento de mensagens, além de envio pela Internet.

O principal objetivo do surgimento do serviço foi à substituição dos *paggers*, pois, ao contrário deste mecanismo, uma vez enviado um SMS, tem-se a garantia que será entregue. Um dispositivo destinatário irá receber a mensagem enviada, mesmo que este esteja desligado, pois, a mensagem fica armazenada na operadora até que esteja novamente disponível para o recebimento.

Outra característica do SMS é o de trabalhar em canais de comunicação separados dos canais de voz e dados, logo, é possível utilizar em paralelo a outros recursos do dispositivo como uma conversação.

Dentre todos os serviços atualmente oferecidos pelas operadoras, este é o que tem maior destaque, por se tratar de um serviço simples, interoperável e mais acessível aos usuários finais no quesito custo, principalmente para quem utiliza um plano pré-pago. Os adolescentes são os que mais utilizam deste recurso,

na Finlândia em uma pesquisa realizada pela Nokia, sabe-se que um adolescente chega a enviar mais de 100 mensagens por mês (BAZOTTI, 2007).

De acordo com Comitê Gestor da Internet no Brasil (2010), atualmente o SMS também está ganhando mercado na área comercial e interatividade, isso pode ser observado no anexo a, por exemplo, é facilmente visível aos usuários de Internet e espectadores de meios de comunicação como rádio e televisão, a venda de algum produto ou serviço ou até o ingresso em alguma promoção por meio do envio de uma mensagem com características pré-determinadas (como o corpo da mensagem).

2.5.2 Wireless Access Protocol

Ao contrário de um computador desktop, os dispositivos móveis tem uma capacidade de processamento muito pequena além do inconveniente tamanho da tela⁴, então em 1997 um grupo de empresas criou o WAP Forum, determinou um protocolo padrão de comunicação para funcionar em dispositivos móveis com pouca capacidade. O protocolo WAP utiliza a linguagem *Wireless Markup Language* (WML), pois páginas desenvolvidas em outras linguagens para Web como HTML são consideradas inóspitas aos dispositivos.

⁴ Comumente, dispositivos móveis têm uma tela muito pequena por causa do espaço necessário para o teclado do aparelho, porém, atualmente este quadro está em mudança com a evolução da tecnologia *Touch Screen* para os dispositivos portáteis.

As redes de dados disponíveis aos dispositivos móveis também utilizam uma banda de transmissão relativamente baixa se comparado a outros padrões sem fio, além de menos estabilidade e mais latência. Outras características do protocolo WAP incluem interoperabilidade e sua correta funcionalidade na grande maioria dos dispositivos móveis existentes no mercado evitando que programadores fiquem escrevendo aplicações diferentes para cada modelo de dispositivo existente.

Outro fato considerado no desenvolvimento do protocolo é que o usuário não tem a disposição mouse e teclado como nos computadores, logo o WAP possui limitações diversas que tendem a tornar as aplicações desenvolvidas rápidas e simples, porém, de fácil utilização pelos teclados dos celulares.

2.5.3 General Packet Radio Service

O WAP em sua forma nativa na segunda geração não alcançou o sucesso esperado, suas limitações, principalmente no quesito velocidade, deixam muito a desejar, e com a gradativa evolução dos aparelhos móveis e pela quantidade de dispositivos ativos no mercado que crescem de forma rápida, houve a necessidade da evolução da tecnologia utilizada. Alterar a tecnologia existente é inviável, pois, os equipamentos da segunda geração trabalhavam com baixas taxas de transmissão além do pequeno leque de opções de segurança, personalização e aplicações úteis oferecidos aos usuários finais, então, viu-se a necessidade do

desenvolvimento de uma nova tecnologia. O GPRS foi desenvolvido com o objetivo de expandir a utilização de outros serviços ofertados pelas operadoras. Até então, o serviço de voz representavam aproximadamente 98% do faturamento das operadoras, ficando apenas 2% para outros serviços (BAZOTTI, 2007).

Com o seu surgimento, muita coisa mudou, a cobrança da utilização mudou de tempo conectado para quantidade de dados trafegados, também surgiu ofertas de pacotes de dados separados do plano de voz adotados pelo cliente. O principal objetivo é oferecer possibilidade de acessar as redes de dados padrão, como TCP/IP, somado isso à característica de velocidade que é superior a de seu “irmão” mais velho, os serviços de dados começaram a ganhar mais mercado com o surgimento de novas aplicações como chat e e-mail.

2.5.4 Enhanced Data for GSM Evolution

Nada mais é do que uma evolução do GPRS no sentido de banda, pois conta com velocidades de transmissão superior. É o padrão de comunicação mais comum no Brasil, que atualmente, está em fase de transição para a terceira geração.

2.5.5 Multimedia Messaging Service

Considerado uma evolução do SMS. Originalmente criado dentro da tecnologia 2,5G seu objetivo é tirar proveito dos recursos ofertados desta geração em diante. Conforme explica Bazotti (2007) a expectativa do MMS é a gradual substituição do SMS na medida em que o mercado vai aderindo a idéia.

As principais características incluem a *não* limitação de 160 caracteres por mensagem e a possibilidade de se incluir recursos como imagens ou vídeos na mesma, por exemplo, por meio deste serviço também é possível enviar e receber e-mails diretamente no celular sem a necessidade de ser abrir o browser e navegar na página do e-mail, teoricamente menos dados trafegados e conseqüentemente menor a tarifação.

2.5.6 Universal Mobile Telecommunication System

A tecnologia, introduzida na terceira geração visa principalmente o desenvolvimento dos serviços oferecidos pela *Web* e altas taxas de transmissão. Seu objetivo é atender a demanda cada vez maior de aplicativos que funcionam pela Internet. Os novos serviços que estão disponíveis com esta nova tecnologia incluem vídeo-chamada, possibilidade de navegação por sites HTTP, *stream*⁵, vídeo-conferência, jogos multi-jogador online, além de muitos outros.

⁵ Stream, fluxo de dados, objeto que permite realizar entradas e saídas organizados em bytes.

Conseqüentemente a todos estes serviços, existe a necessidade da comunicação de dados (transmissão e recepção), logo, atualmente apenas um plano de voz não é suficiente, a contratação de um plano de dados se torna indispensável para que o usuário possa usufruir de todos estes serviços, por isso existe no mercado uma ampla gama de ferramentas e recursos de desenvolvimento permitindo que a cada dia surjam novos aplicativos com diferentes funções.

3 TECNOLOGIAS DE DESENVOLVIMENTO

Para o desenvolvimento de aplicativos para dispositivos móveis existem as mais diversas ferramentas, *IDE* ou Ambiente Integrado de Desenvolvimento, e *SDK*, Kit de Desenvolvimento de Software. Muitas fornecidas ou recomendadas pelo próprio fabricante do aparelho e outras optadas por escolha individual. Cada sistema operacional dos *smartphones* tem as suas especificações e restrições quanto à linguagem que será utilizada para o desenvolvimento, dentro de um *Windows Mobile*, por exemplo, temos .NET ou Java, já para um *Symbian*, sistema operacional da Nokia, os aplicativos podem ser desenvolvidos na linguagem Symbian C++ ou em Java.

3.1 JAVA

Desenvolvida na década de 90, pela Sun Microsystem, o principal objetivo de seus desenvolvedores foi a criação de uma linguagem simples e que permitisse ser escrita apenas uma vez e pudesse ser executada em qualquer plataforma disponível no mercado.

Segundo Damasceno e Rego (2007, p. 44) “a JavaSoft, empresa do grupo Sun Microsystems que [...] controla a compatibilidade e o desenvolvimento das sucessivas versões do Java, [...]”. Essas distribuições contêm desde ferramentas até conjuntos de *API*, também conhecido como interface de

programação de aplicativos que servem para o desenvolvimento e são subdivididas em três grandes grupos:

- a) J2SE, Java2 Standard Edition, visa o desenvolvimento de aplicativos cliente/servidor;
- b) J2EE, Java2 Enterprise Edition, este grupo tem como objetivo fornecedor *API* para a integração entre sistemas e o desenvolvimento corporativos;
- c) J2ME, Java2 Micro Edition, é grupo que contém toda a gama de *API* para o desenvolvimento de aplicativos para dispositivos de pouco poder computacional (MUCHOW, 2004).

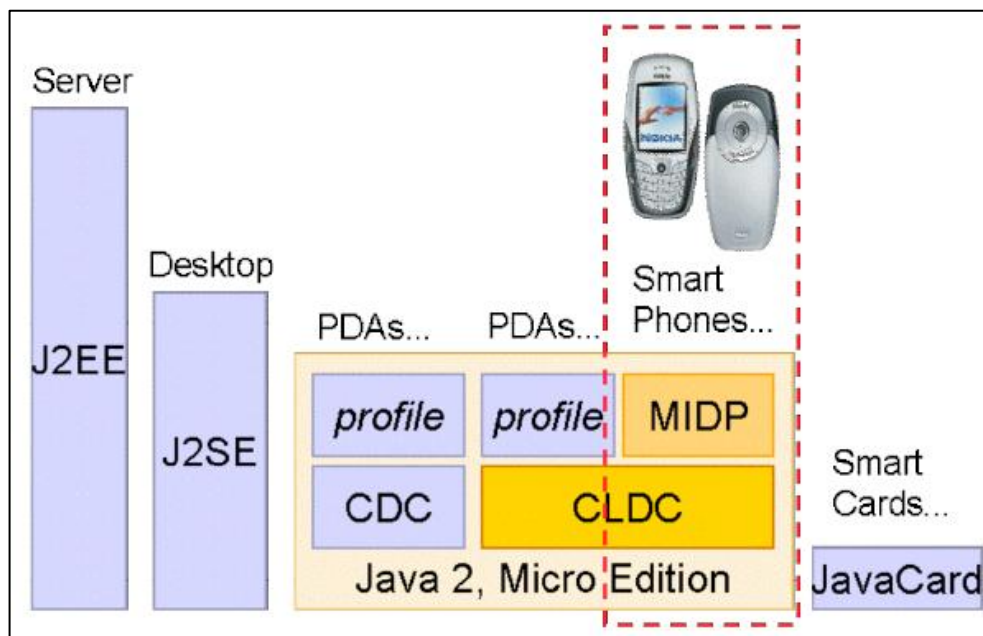


Figura 3. Arquitetura Java.
Fonte: Java Móvel (2009).

Além dos três grupos de *API* de desenvolvimento disponíveis, há pelo menos mais três tecnologias disponibilizadas pelo grupo Sun que devem ser compreendidas:

- a) JVM, Java Virtual Machine, é o software responsável pela emulação⁶ e execução do código Java na plataforma escolhida;
- b) JRE, Java Runtime Environment, é o ambiente onde será executado o código Java, de modo que deve estar instalado no dispositivo destino do aplicativo Java. A JRE é composta pelo conjunto de *API* e pela JVM;
- c) SDK, Software Development Kit, é todo o conjunto de ferramentas e bibliotecas necessárias para o desenvolvimento de aplicativos Java. Ela é composta pela JRE e aplicativos de compilação, documentação e depuração do código fonte.

Algumas das principais características da linguagem Java, incluem simplicidade, com poucas palavras reservadas. Orientação ao objeto, o modelo de programação mais adotado atualmente, pois permite uma boa estruturação de qualquer projeto, facilitando e agilizando o desenvolvimento. Portabilidade, principal características que torna o Java o centro das atenções quando se está escolhendo qual linguagem de programação utilizar em novos projetos.

⁶ Emulação, derivado de emulador, é o software responsável por simular ser algum outro aplicativo ou dispositivo.

3.2 JAVA2 MICRO EDITION

A versão micro do Java, nasceu da necessidade do desenvolvimento de aplicativos para dispositivos de pouco poder computacional, como *paggers* e até celulares. Caracterizado pela necessidade mínima de recursos para a sua execução. A máquina virtual do J2ME, a KVM, Máquina Virtual K, também exige pouco poder de processamento, podendo, por exemplo, ser executada em processadores de 16 bits com apenas de 25MHz de frequência.

A arquitetura do Java para dispositivos móveis pode ser visualizada na Figura 4.



Figura 4. Arquitetura J2ME.

Fonte: Adaptado de Muchow (2004).

- a) Perfil MID, são extensões opcionais para o desenvolvedor. Estas extensões fornecem bibliotecas específicas para os dispositivos que se deseja desenvolver. Segundo Muchow (2004) é importante

salientar que cada dispositivo possui uma configuração de hardware particular, de modo que, o que pode ser muito para um, pode ser pouco para outro, é inviável o desenvolvimento específico para um aplicativo *touchscreen*⁷ e posteriormente a utilização deste mesmo software em outro que não seja;

- b) conforme Muchow (2004) as bibliotecas básicas são divididas em dois grupos, um conhecido como CDC, Configuração de Dispositivos Conectados, e o outro como CLDC, Configuração de Dispositivos Conectado Limitado, basicamente a diferença entre um e outro está na configuração mínima necessária para sua utilização, por exemplo, um aplicativo Java feito com o primeiro necessitaria de no mínimo 512 kilobytes de memória para ser executado, já um aplicativo feito com o segundo, precisaria de apenas 128 kilobytes para a sua execução;
- c) máquina Virtual K, já comentada anteriormente, atualmente existe várias disponíveis de fabricantes distintos. A escolha de qual utilizar depende da necessidade do desenvolvedor. Alguns sistemas operacionais de *smartphones* já vêm com uma máquina virtual implantada, como o *Symbian* da Nokia, dispensando a

⁷ Touchscreen, sensível ao toque, são os dispositivos que dispensam (não é regra, um aparelho sensível ao toque pode ter teclado também) a utilização de teclados, por sua tela aceitar comandos de toque.

instalação de um de terceiro, salvo alguma exceção do desenvolvedor (MUCHOW, 2004);

- d) sistema operacional hospedeiro, é o sistema operacional do dispositivo em si, aqui vai depender exclusivamente para qual dispositivo o desenvolvedor vai criar suas aplicações. Alguns sistemas no mercado bastante conhecidos incluem, *Windows Mobile* da Microsoft, *Symbian* da Nokia e *Android* da Google.

Com o entendimento da linguagem de programação escolhida, é necessário ter em mente antes de começar a programar, como o aplicativo deverá ficar quando pronto, assim, cria-se um trilha a ser seguida pelo desenvolvedor. Os estudos agora são direcionados a comunicação no meio computacional, tendo em vista a necessidade do entendimento dos protocolos existentes para escolher qual irá melhor se adaptar ao sistema proposto.

4 REDES DE COMUNICAÇÃO

De acordo com Colcher, *et al.* (2005), pode-se dizer que uma rede, é dois ou mais dispositivos que trocam informações por meio de um sistema de comunicação, neste caso, existe a necessidade de ambos enviarem e receberem as informações de modo entendível. Estes dispositivos são chamados de Módulos Processadores (MPs), pois ambos têm a capacidade de trocar informações através de um sistema de comunicação, logo, um MPs pode ser, por exemplo, desde um computador até um telefone.

Essas redes são formadas sobre uma arquitetura física (Topologia⁸) e um protocolo⁹.

4.1 VOZ SOBRE IP

Este conceito não é algo novo, ele vem se desenvolvendo desde o final da década de 90 quando surgiu o primeiro aplicativo comercial de transmissão de voz sobre pacotes IP. Chamado de *Internet Phone*, desenvolvido pela *VocalTec Communications*, este aplicativo possibilitava o envio de voz entre computadores pessoais, porém, em virtude das baixas taxas de transmissão e tecnologia precárias

⁸ Topologia é o modelo físico de como os MPs e os cabos (quanto aplicado) estão distribuídos na rede, exemplos de arquitetura incluem, estrela, anel e barra.

⁹ Protocolo são as regras de comunicação, para que exista uma rede, os MPs da mesma devem se comunicar, para haver esta comunicação, os MPs devem falar a “mesma língua”, esta linguagem é o protocolo utilizado.

ao modelo proposto, a qualidade da voz transmitida não tinha potencial para concorrer com os sistemas telefônicos existentes.

Proposto inicialmente como um modo de comunicação entre computadores, mais tarde, na medida em que foi evoluindo, sua área de atuação também aumentou. Segundo Minoli (2006) em meados do ano 2000 é que começaram a surgir os primeiro sistemas comerciais VoIP em cima do modelo 1G, funcionamento de forma interligada ao serviço telefônico comercial. A partir de então com a expansão da Internet e aumento das taxas de transmissão, começaram a surgir empresas produzindo em larga escala equipamentos específicos para o VoIP, como roteadores, aparelhos telefônicos (telefones IP), adaptadores e outros. Como resultado desta evolução, ele também sofreu grandes melhorias em sua qualidade, assim com também começaram a surgir os primeiros padrões definidos pela ITU, *International Telecommunications Union* e IETF, *Internet Engineering Task Force*.

Basicamente, consiste na codificação da voz e seu envio por meio de uma rede IP. A comunicação entre um sistema de telefonia móvel tradicional e um sistema VoIP é possível por meio de um *gateway*, este seria responsável pela interpretação e conversação entre as duas redes distintas.

Para que exista a comunicação multimídia entre dois dispositivos, tem-se a necessidade da utilização de um chamado *protocolo de inicialização* estes, são os responsáveis pela ligação e transmissão multimídia entre dispositivos. Desenvolvido pela ITU em conjunto com a IETF, o protocolo de

inicialização H.323 foi um dos primeiros a ser desenvolvido. Graças a esta padronização é que os mais diversos aplicativos a equipamentos de fornecedores distintos pudessem estabelecer uma comunicação. Mas, conforme Vaz e Dinau (2006), o protocolo H.323 é considerado por muitos como complexo e de difícil implementação, por isso tende a cair em desuso, em contrapartida surgiu o protocolo SIP, que aos poucos vem ganhando força e tomando o lugar dos protocolos antigos.

4.2 REDES SEM FIO – WIRELESS

As redes sem fio se tornaram popular, principalmente pela necessidade de colocar computadores em uma rede de dentro de um escritório e a inviabilidade da passagem de cabos. As redes sem fio, como o próprio nome sugere, dispensa a necessidade de se espalhar cabos pelo ambiente, é fácil de utilizar e seu custo acaba sendo muito mais acessível uma vez que não há necessidade da aquisição de cabos ou roteadores, mesmo quando se deseja aumentar o número de máquinas na rede, basta configurar o ponto de acesso e pronto. Segundo RISCHPATER (2001), além da facilidade de acesso, as redes sem fio permite que outros dispositivos além de computadores sejam conectados, como *Smartphones*.

Atendendo as normas 802.11 definidas pela IEEE. Atualmente as redes sem fio são vistas como algo muito maior que apenas interligar computadores dentro de um escritório, já é realidade, por exemplo, a existência de

redes sem fio em lugares públicos, como universidades, aeroportos, shoppings e até centros de cidades. A lista de aparelhos que utilizam esta tecnologia também aumentou consideravelmente, hoje temos notebooks, netbooks, celulares e outros dispositivos de uso pessoal.

4.3 PROTOCOLO SIP

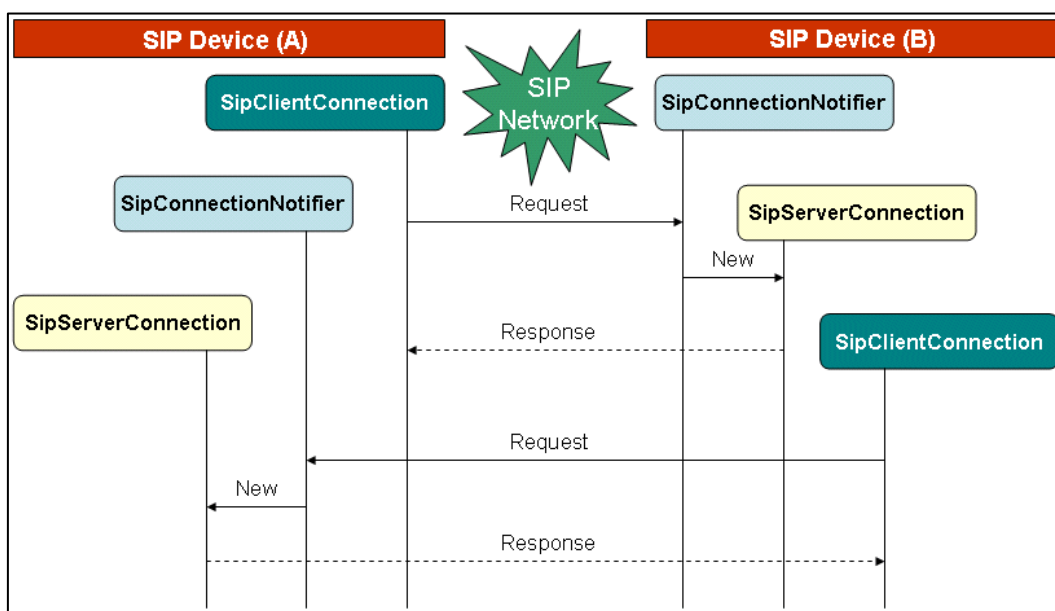
Atuante no nível da aplicação é mais flexível que o H.323, pois foi desenvolvido para trabalhar em conjunto com outros protocolos, de modo que o SIP pode ser utilizado para iniciar, gerenciar e finalizar sessões ao mesmo tempo em que para a transmissão do conteúdo multimídia é feita por outro protocolo distinto.

Este protocolo especifica uma arquitetura cliente/servidor, e sua estrutura é formada pelos seguintes componentes:

- a) agente usuário, formado por duas sub partes, a primeira usuário cliente, que realiza a requisição SIP e a segunda, usuário servidor, que recebe e responde as requisições;
- b) servidor Proxy, atua com cliente e servidor ao mesmo tempo, executando as requisições de clientes que não podem executar diretamente ao servidor;
- c) servidor de redirecionamento, responsável pelo mapeamento dos endereços associados aos clientes;

- d) servidor de registro trabalha em conjunto com o servidor de proxy e redirecionamento, armazenando os dados necessários.

Conforme Colcher, *et al.* (2005) a flexibilidade deste protocolo também permite a comunicação *Peer-to-Peer*¹⁰ (P2P) sem a necessidade de intermediários, isso pode ser implementado para dispositivos se comunicarem diretamente, porém, dentro de uma realidade mais ampla como um sistema de telefonia baseado em SIP, a existência de todos os componentes citados é indispensável e vital para um pleno funcionamento do sistema.



61

Figura 5. Comunicação SIP.

Fonte: Mahmoud (2004).

¹⁰ Peer-to-peer: Segundo Pelegrina (2010), peer-to-peer ou P2P em sua forma diminutiva, significa, a conexão direta entre dispositivos, de modo que não seja necessário um servidor central, pois, os próprios dispositivos se revezam na tarefa cliente/servidor.

Este protocolo de comunicação é um entre muito existentes, conforme Martins (2009), é visível que a quantidade de aplicativos existentes no mercado atualmente que utilizam este protocolo para a comunicação VoIP. Porém, para o desenvolvimento em curto prazo, precisa-se conhecer outras ferramentas e trabalhos desenvolvidos para verificar qual o melhor método que pode ser utilizado.

5 TRABALHOS CORRELATOS

5.1 COMUNICAÇÃO ENTRE DISPOSITIVOS DE BAIXO PODER COMPUTACIONAL: BLUETOOTH COMO PROTOCOLO DE COMUNICAÇÃO PARA JOGOS PARA CELULAR

O trabalho final de estágio foi desenvolvido por Álvaro Ramos de Medeiros Raposo, em junho de 2005, no Instituto de Ensino Superior FUCAPI – CESF e orientado pelo Professor Rodrigo Carlos Silva e Silva no curso de Engenharia de Comunicações.

O trabalho tem como principal objetivo demonstrar como funciona tecnologias como Bluetooth, infravermelho e J2ME, além de demonstrar como é possível juntar essas tecnologias no desenvolvimento de um jogo, com o intuito de entrar no mercado de trabalho atendendo a demanda de jogos para dispositivos móveis solicitada.

5.2 VOIP PARA COMPUTAÇÃO MÓVEL

O trabalho de conclusão de curso foi desenvolvido por Ezequiel Bazotti, em dezembro de 2007, na Universidade de Cruz Alta - UNICRUZ e orientado pelo Professor Luis Claudio Gubert no curso de Ciência da Computação.

O trabalho tem como objetivo o desenvolvimento de uma aplicação na plataforma J2ME e que por meio das tecnologias Bluetooth e Wireless, estabeleça uma conexão ponto-a-ponto entre os clientes permitindo que haja a transmissão e recepção de voz neste canal aberto.

6 APLICATIVO MÓVEL PARA COMUNICAÇÃO DE VOZ

O sistema faz o aproveitamento de tecnologias já existentes e que estão se tornando cada vez mais comuns nos dispositivos móveis atuais. Por meio de um protocolo de comunicação, será criado um canal de conversação entre dois dispositivos celulares que tenha o recurso de rede sem fio disponível. Após ser criado, este então irá, trafegar dados de voz entre os dois pontos permitindo assim que haja transmissão e recepção entre usuários distantes entre si, mas dentro da rede sem fio.

Para o desenvolvimento inicial serão utilizados intermediários como *routers*, repetidores ou pontos de acesso. A conexão não será direta, tornando necessária a utilização de equipamentos de terceiros, porém, a partir do momento que cada usuário obtiver um endereço IP associado ao seu aparelho quando conectado a Internet, é possível que sejam realizadas conexões diretas se o intermédio de outros equipamentos.

6.1 METODOLOGIA

A necessidade do desenvolvimento do projeto proposto surgiu da necessidade de comunicação a custos mais acessíveis, como dentro de uma organização setorizada, isso dispensaria a compra de aparelhos telefônicos como ramais e a necessidade de linhas digitais ou centrais telefônicas.

Inicialmente foi realizado o levantamento bibliográfico a respeito das tecnologias disponíveis atualmente, como J2ME e Wireless. Também foi feito um levantamento histórico sobre a comunicação e os principais sistemas e serviços oferecidos em cada geração da comunicação.

Após esses levantamentos foram pesquisados trabalhos semelhantes com o intuito de analisar e auxiliar no desenvolvimento.

Para a concretização do projeto pretendido, as ferramentas de desenvolvimento necessárias foram instaladas no computador pessoal para o início do desenvolvimento. A tecnologia J2ME já comprovou oferecer os recursos necessários para o desenvolvimento, possuindo bibliotecas para a comunicação Wireless e o tratamento de sinais de voz.

Para o desenvolvimento foi utilizado a IDE *NetBeans* versão 6.9.1, em sua versão completa, esta ferramenta possui todas as bibliotecas para dispositivos móveis incluídas, dispensando um longo tempo em configuração. Também foram utilizados os kits Sun Java Wireless Toolkit 2.5.2_01 for CLDC e Java (TM) ME Platform SDK 3.0, ambos contém bibliotecas e emuladores de dispositivos móveis além de ferramentas de desenvolvimento, porém, foram instalados apenas para ter acesso a seus emuladores que são mais completos. Como os dispositivos para testes práticos são aparelhos N95 e N8, ambos rodando o sistema operacional *Symbian*, versão *S60 FPI* e *S^3* respectivamente, também houve necessidade de instalar as bibliotecas referente aos dispositivos, a *Nokia Developer Tools* na versões *3rd Edition FPI SDK* e *Symbian3 SDK* respectivamente. Sem estas

bibliotecas não tem como gerar o aplicativo Java para os dispositivos *Symbian*. As instalações são simples e podem ser baixadas direto do site da Nokia.

Uma vez com as bibliotecas de desenvolvimento prontas também foi instalado o *framework* Floggy para persistência, facilitando o trabalho de armazenamento de dados nos dispositivos, no anexo C é demonstrando como configurar na IDE *NetBeans*.

O computador de desenvolvimento é um notebook *Dual Core* com 3GB de memória RAM, também foi necessário pelo menos o mesmo tamanho de espaço no HD para instalar todas as ferramentas. Para a transmissão do aplicativo para os dispositivos móveis foi utilizada uma conexão *Bluetooth*.

Depois de todo instalado, configurado e testado, deu-se início ao processo de desenvolvimento, sendo este dividido em etapas para se atingir os objetivos desejados.

6.2 ETAPAS DO DESENVOLVIMENTO

O desenvolvimento do projeto foi dividido em cinco etapas. Em cada uma foram efetuados estudos, desenvolvimento e testes separados, somente após a conclusão de uma etapa é que a mesma foi unida ao projeto principal.

6.2.1 A linguagem de programação

Optou-se por J2ME principalmente por sua versatilidade multiplataforma e disponibilidade de todos os recursos necessários para o desenvolvimento. A linguagem em si é relativamente simples, e fácil compreensão para alguém que já tenha tido algum contato com outra versão do Java (J2SE ou J2EE), pois seguem o mesmo princípio. Para a aprendizagem, talvez a maior dificuldade seja encontrar livros em português, porém, é notória a grande quantidade de tutoriais e exemplos disponíveis na Internet e até mesmo na IDE de desenvolvimento utilizada.

A estrutura de programação que um aplicativo J2ME irá ter segue um padrão para que seja reconhecido como válido e funcione corretamente pelo dispositivo destino, é indispensável à criação de uma classe herdada da classe *MIDlet*. Esta será a classe principal do aplicativo responsável pelo gerenciamento do mesmo e principalmente como intermediário com o dispositivo, recebendo solicitações de inicialização, parada e finalização, como diz Muchow (2004) é um aplicativo, desenvolvido para ser executado em dispositivos móveis.

```
1 import javax.microedition.lcdui.Display;
2 import javax.microedition.midlet.*;
3
4 /**
5  * @author Murilo
6  */
7 public class MidletOla extends MIDlet {
8
9     public void startApp() {
10         OlaForm olaForm = new OlaForm();
11         Display.getDisplay(this).setCurrent(olaForm);
12     }
13
14     public void pauseApp() {
15     }
16
17     public void destroyApp(boolean unconditional) {
18     }
19 }
```

Figura 6. Exemplo de MIDlet.

Fonte: Autoria própria.

Na Figura 6 está ilustrado um exemplo de aplicativo herdado da classe *MIDlet*, ele sobreescreve três métodos essenciais para que o aplicativo funcione. O primeiro *startApp()*, é executado quando o aplicativo for colocado em modo ativo, dentro dele está sendo criado uma interface que será vista mais adiante. Após a criação desta, é efetuado uma chamada a classe *Display*, estática, esta classe é responsável pelo gerenciamento, controlando o que será exibido na tela do dispositivo, a função *getDisplay* que recebe o *MIDlet* como parâmetro, tem um método chamado *setCurrent*, responsável por definir o que será exibido para o usuário pelo aplicativo. As funções *pauseApp()* e *destroyApp()* são chamadas respectiva quanto o dispositivo faz uma solicitação de parada e finalização.

```

1 import javax.microedition.lcdui.Command;
2 import javax.microedition.lcdui.CommandListener;
3 import javax.microedition.lcdui.Displayable;
4 import javax.microedition.lcdui.Form;
5 import javax.microedition.lcdui.StringItem;
6
7 /**
8  *
9  * @author Murilo
10  */
11 public class OlaForm extends Form implements CommandListener{
12
13     private StringItem olaJ2ME = null;
14     private Command cmSair = null;
15
16     public OlaForm() {
17         // Construtor da tela.
18         super("Olá J2ME!");
19         // Cria um objeto do tipo StringItem, passando 2 parâmetros do tipo texto.
20         olaJ2ME = new StringItem("Olá J2ME", "Meu aplicativo no smartphone!!!");
21         // Adiciona o objeto criado nesta tela.
22         this.append(olaJ2ME);
23         // Cria um comando de saída.
24         cmSair = new Command("Sair", Command.EXIT, 0);
25         // Adiciona o comando de saída à tela.
26         this.addCommand(cmSair);
27     }
28
29     public void commandAction(Command c, Displayable d) {
30     }
31 }

```

Figura 7. Classe OlaForm, tela que será visualizada pelo usuário.

Fonte: Autoria própria.

Com o *MIDlet* para gerenciar o aplicativo pronto, falta apenas a classe com a interface que usuário irá visualizar e interagir. A Figura 7, representa um exemplo, que não possui nenhum objeto de entrada de dados, apenas um componente do tipo *StringItem* utilizado para exibir uma mensagem ao usuário e uma ação do tipo sair. Como o projeto não visa à aprendizagem da linguagem, algumas explicações estão disponíveis no próprio código-fonte exibido. O resultado pode ser visualizado na Figura 8, onde o aplicativo está rodando em um emulador de celular. Vale salientar que o modo como cada objeto será exibido depende inteiramente do dispositivo em questão, porém, a funcionalidade será a mesma.



Figura 8. MIDlet rodando no emulador.
Fonte: Autoria própria

6.2.2 Funcionamento da comunicação

Para o sucesso do desenvolvimento foi essencial à pesquisa e o estudo de qual protocolo de comunicação utilizar. Dentre os protocolos existentes, foram estudados *SIP*(protocolo), *Socket* (interface) e *Datagram* (serviço), todos disponíveis no J2ME e com os recursos necessários para o aplicativo. Abaixo uma breve descrição de cada um:

- a) *SIP*, apesar da facilidade de utilização, seu funcionamento ficou restrito apenas ao emulador, não funcionando como esperando no

dispositivo, um problema encontrado foi em relação ao IP, onde sempre era atribuído o de *loopback*¹¹ impedindo a comunicação entre aparelhos distintos dentro de uma rede local. Outro foi em relação ao suporte dos dispositivos ao *SIP*, tornando limitado a gama de aparelhos compatíveis com o aplicativo;

- b) *Socket* apresentou inicialmente o mesmo problema do *SIP* em relação ao IP, porém, mesmo encontrando formas de contornar essa questão, esse protocolo trabalha de forma TCP, criando um gargalo na transmissão dos dados, conforme diz VoipBra (2010), como o protocolo TCP garante a entrega dos pacotes, por isso, atrasos são inevitáveis.

6.2.2.1 *Datagram*

Dos três protocolos, foi o escolhido para o desenvolvimento, uma vez que se trata de dispositivos com um poder de processamento reduzido e que serão enviados grandes quantidades de pacotes binários. Segundo Bazotti (2007), o *Datagram* funciona sobre o protocolo UDP, portanto não existe a preocupação de verificar se todos os pacotes chegaram ao destino. A programação foi dividida nas classes cliente e servidor para facilitar, sendo toda estruturada em *Threads* para

¹¹ *Loopback*, canal de comunicação onde os dados enviados retornam ao transmissor, sua principal função é a realização de testes.

permitir o envio e recebimento de informações de forma independentes, porém a maior diferença entre ambos está na forma como é criada a conexão, enquanto no cliente e utiliza o IP do servidor, no lado do servidor apenas a porta de comunicação é utilizada para indiciar a abertura de um canal de comunicação.

```

56     public void run() {
57         try {
58             mensagemRecebida.setText("Aguardando conexão...");
59             DatagramConnection datagramConexao =
60                 (DatagramConnection)Connector.open("datagram://:5555");
61             sender = new Sender(datagramConexao);
62             while (true) {
63                 Datagram datagram = datagramConexao.newDatagram(100);
64                 datagramConexao.receive(datagram);
65                 enderecoIP = datagram.getAddress();
66                 mensagemRecebida.setText("Recebido: " +
67                     new String(datagram.getData(), 0, datagram.getLength()));
68             }
69         } catch (IOException ioe) {
70             Alert alErro = new Alert("VoIP Servidor", "Porta 5555 já está em uso!",
71                 null, AlertType.ERROR);
72             alErro.setTimeout(Alert.FOREVER);
73             alErro.setCommandListener(this);
74             VoIP.setCurrent(alErro);
75         } catch (Exception e) {
76             VoIP.showException(e);
77         }
78     }

```

Figura 9. Criando um servidor *Datagram* para mensagens de texto.

Fonte: Adaptado do código-fonte exemplo do *NetBeans*.

É possível observar na Figura 9 um exemplo de servidor *Datagram* para receber mensagens de texto. A criação do servidor por meio da função *open* tem como parâmetros o protocolo utilizado e a porta e escuta, isso indica que o aplicativo ficará escutando na porta indicada. O loop¹² serve para indicar que a *Thread* irá ficar funcionando ininterruptamente, dentro dela um objeto do tipo *Datagram* é criado, passando como parâmetro o tamanho máximo que este poderá

¹² *Loop*, laço de repetição, na Figura 9 é representado pela função *while()*.

ter, logo em seguida é executado uma chamada a função *receive()*, colocando o servidor no modo de espera por um pacote de algum cliente.

```
public void run() {  
54     try {  
55         DatagramConnection dc =  
56             (DatagramConnection)Connector.open("datagram://127.0.0.1:5555");  
57         mensagemRecebida.setText("Conectado ao servidor VoIP");  
58         sender = new Sender(dc);  
    }
```

Figura 10. Lado cliente do aplicativo.

Fonte: Adaptado do código fonte exemplo do *NetBeans*.

Já no lado cliente do aplicativo, como demonstrado na Figura 10, na função *open* também é passado o IP do servidor, neste caso foi utilizado o IP local para exemplificação. Uma vez estabelecido à conexão e transmitindo os dados como esperado, viu-se a necessidade do aplicativo armazenar informações referentes às configurações de conexão, assim cada vez que o aplicativo fosse iniciado não seria necessário configurá-lo novamente.

6.2.3 Framework de persistência – Floggy

Como o aplicativo trabalha com telas de configurações e contatos para facilitar a usabilidade, fez-se necessário a utilização de um recurso para armazenar estes dados. O J2ME por padrão possui um banco de dados nativo e robusto chamado RMS ou sistema de gerenciamento de registro. Mas, em virtude de sua

programação extensa e trabalhosa, optou-se pela utilização de um *framework*¹³ para auxiliar na persistência dos dados.

Desenvolvido como um projeto de conclusão de curso por alunos da UFSC, segundo o fórum de discussão que os desenvolvedores participam, GUI (2007), o *framework* foi desenvolvido para auxiliar no desenvolvimento e facilitar manutenções na parte de persistência de dados.

Como sua utilização foi opcional no projeto, não será entrado em detalhes quanto a sua programação. Até este momento do projeto, já se tem um aplicativo que envia e recebe mensagens de texto e com capacidade armazenar as configurações facilitando a vida do usuário final, porém, ainda está muito pouco usual para alguém que nunca tenha entrado em contato com o aplicativo, criando dúvidas quanto o seu uso e funcionalidades.

6.2.4 A estrutura do aplicativo

Um dos pontos mais simples do desenvolvimento, porém, de grande importância, é a forma que o aplicativo terá. A definição de suas telas é importante pela questão de usabilidade do aplicativo por usuários iniciantes.

¹³ *Framework*, pacote de bibliotecas já programadas que visam facilitar alguma operação da programação convencional.

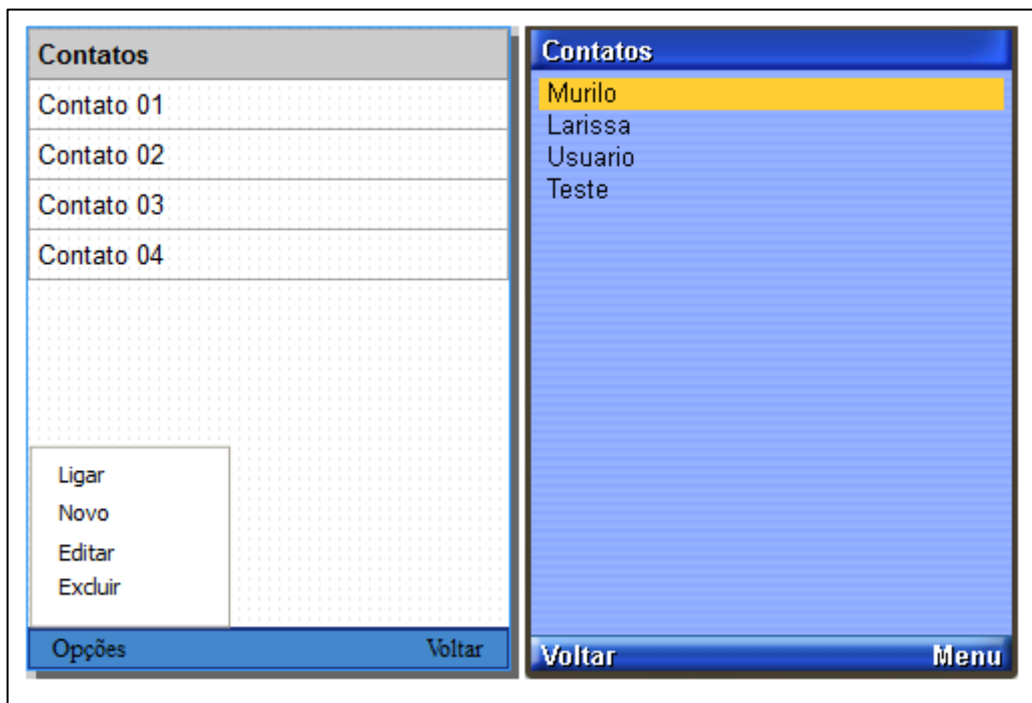


Figura 11. Respectivamente, lista de contatos no *sotryboard*¹⁴ e como ficou no emulador.

Fonte: Autoria própria.

Para se chegar ao modelo atual do aplicativo foi primeiramente criado um diagrama de atividades referente ao fluxo de como deverá ocorrer à conexão, este diagrama disponibilizado no Apêndice B é importante para tentar criar um modelo que seja o mais simples possível evitando muitas iterações do usuário final para concluir a operação. Já no Apêndice está disponível o diagrama de classes do aplicativo quando ainda em estágio de desenvolvimento. Na Figura 11, é visível um esboço previsto de como ficaria a tela de listagem de contatos e como ficou na prática rodando em um emulador. Para a facilidade de manuseio, foi

¹⁴ *Storyboard*, técnica utilizada para desenhar telas, prevendo como deverão ficar antes de serem desenvolvidas na prática.

adotada a abordagem de listas, deixando os menus laterais apenas para casos de real necessidade, isso pode ser observado na Figura 12 que exibe a tela principal do aplicativo e pode ser observada sua semelhança com a tela de contatos.

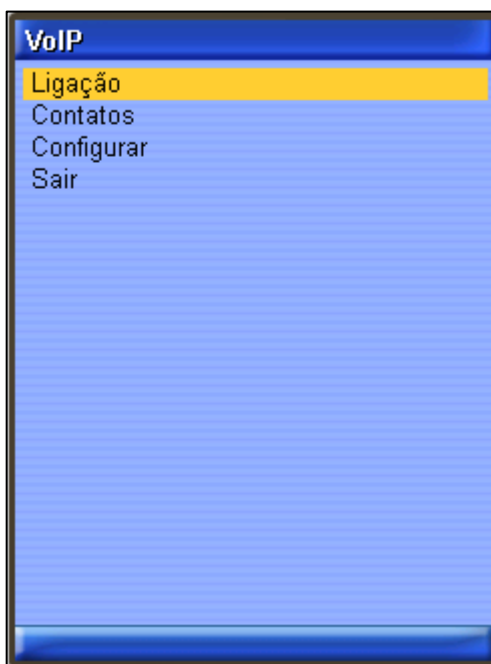


Figura 12. Tela principal do aplicativo no emulador.

Fonte: Autoria própria.

Com o término da estruturação do aplicativo, a transmissão de mensagens de texto e todos os dados necessários devidamente armazenados, o desenvolvimento do aplicativo chegou a sua etapa final e com maior âmbito de pesquisa, a conversação.

6.2.5 Gravação e execução de áudio

O J2ME possui bibliotecas específicas e simples para a utilização dos recursos multimídias. Com elas, é possível gravar e reproduzir áudio e vídeo, isso, desde que o dispositivo tenha suporte.

```

100     System.out.println("1 - Iniciando Grava");
101     try {
102         player = Manager.createPlayer("capture://audio?encoding=pcm&rate=" +
103             Constantes.rate + "&bits=" + Constantes.bits +
104             "&channels=" + Constantes.canais);
105         System.out.println("1.2 - Realize");
106         player.realize();
107         player.start();
108         rc = (RecordControl) player.getControl("RecordControl");
109     } catch (Exception ex) {
110         VoIP.showException(ex);
111     }
112
113     ByteArrayOutputStream output = new ByteArrayOutputStream();
114     rc.setRecordStream(output);
115     rc.startRecord();

```

Figura 13. Início do processo de gravação do áudio.

Fonte: Autoria própria.

Na Figura 13 está ilustrado o traço inicial do processo de captura do áudio quando o aplicativo ainda estava sendo desenvolvido. É importante visualizar a criação do objeto *player*, é nele que são definidos os parâmetros da gravação, como *codec*¹⁵ que será utilizado, com ele também obtemos um objeto do tipo *RecordControl*, responsável pelo controle das entradas multimídia do dispositivo, como o microfone, iniciando e finalizando a captura do áudio. Com

¹⁵ *Codec*, biblioteca que irá codificar/descodificar os dados, neste caso de áudio.

este controle também pode ser definido o destino da gravação que será efetuada, como salvar para um arquivo, no caso do aplicativo, foi criado um *Array* de *Bytes* como saída. O método *setRecordStream()*, que recebe um *Array* como parâmetros, está definindo o destino de todo o áudio capturado para a variável *output*, em contrapartida também existe a possibilidade de ser utilizado o método *setRecordLocation* para salvar o áudio capturado para arquivo, porém, em testes no desenvolvimento a escrita em local físico é mais demorada que o armazenamento temporário na memória.

A variável *output* será enviada à classe *Enviar.java* por meio da função *enviaAudio()* como pode ser observado na Figura 14, responsável pelo envio dos dados capturados ao outro ponto da conexão.

```

55     while (true) {
56         try {
57             //         System.out.println("2 - Sleep");
58             Thread.sleep(Constants.tempoCaptura);
59         } catch (InterruptedException ex) {
60             VoIP.showException(ex);
61         }
62         try {
63             //         System.out.println("3 - Commit");
64             //         rc.stopRecord();
65             rc.commit();
66         } catch (Exception ex) {
67             VoIP.showException(ex);
68         }
69         if (!VoIP.isPaused()) {
70             //         System.out.println("4 - Envio");
71             enviaAudio.enviaAudio(Constants.endereco, output.toByteArray());
72         }
73         //         System.out.println("5 - Finalizando");
74         output.reset();
75         rc.setRecordStream(output);
76         rc.startRecord();
77         //         System.out.println("6 - Fim");
78     }
79 }
80 }

```

Figura 14. Fim do processo de gravação de áudio.
 Fonte: Autoria própria.

A *Thread* de captura também foi codificada dentro de um *loop*, assim a captura será constante. Dentro do laço, existe uma chamada a da função *sleep*, nesta será definido o tempo de captura de cada pacote do *Datagram* para enviar ao destino e o método *commit()* do *RecordControl* é o que finaliza o processo de gravação, enviando efetivamente os dados para a variável *output* e parando a gravação, porém, após a chamada deste método, existe a necessidade de definir o destino do áudio novamente, isso é executado mais abaixo quando é feita uma nova chamada da função *setRecordStream()*. Note que a variável *output* também

precisa ser limpa para evitar o acúmulo de informações desnecessárias, por isso é efetuado um *reset()* na mesma.

6.3 ANÁLISE DE CUSTOS

Realizando uma pesquisa via Internet no site das operadoras, foram obtidos os valores tanto do SMS quanto de ligações dos planos pré-pagos. A Tabela 1 apresenta um breve comparativo entre a utilização do aplicativo VoIP e os planos oferecidos.

Tabela 1 – Análise de custos.

Operadora	Para a mesma operadora		Para outra operadora	
	R\$ SMS ¹ (cada)	R\$ ligação (min.)	R\$ SMS ¹ (cada)	R\$ ligação (min.)
Vivo	0,39	1,15	0,39	1,25
Claro	0,30	1,29	0,30	1,29
Tim	0,39	0,75 ²	0,39	1,39
Oi/BrT	0,39	1,39	0,39	1,39
JVoIP	Sem suporte	Gratuito ³	Sem suporte	Gratuito ³

Fonte: Adaptado de UCEL (2009).

¹ Para dentro do país.

² Plano Infinity Pré, pagando apenas o primeiro minuto.

³ Dentro da área de cobertura do *access point*.

*Não foram contabilizados os planos pós-pago.

**Somente para uso dentro do país.

***Promoções foram desconsideradas.

6.4 RESULTADOS OBTIDOS

Após o desenvolvimento da aplicação, deu-se início a realização de testes práticos. Neles o aplicativo demonstrou estabilidade de funcionamento, tanto na parte do servidor quanto na parte do cliente. Todos foram realizados dentro de uma rede local e contaram com a utilização de um *access point* para a distribuição do sinal Wireless. Os dispositivos móveis A e B foram adicionados à rede e iniciaram a conexão, o primeiro como servidor e o segundo como cliente passando o IP do servidor A como parâmetro.

A conexão e o envio dos dados tiveram bons resultados, pois o tempo de resposta do aplicativo é hábil. Mesmo a execução do áudio teve qualidade considerável. Foram detectados dois grandes problemas, o primeiro, foi em relação à gravação do áudio para posterior envio. Como a programação delimita um tempo de gravação, para depois enviar e somente depois iniciar a nova gravação, esse desgaste causa perda de informações durante a fala enviando apenas metade do que deveria. A solução encontrada foi à customização do tempo de gravação na tela de configuração, outro fator importante é o tamanho do *Datagram* que será recebido, quando nas primeiras versões era utilizado um valor fixo, na versão mais atual é possível configurar no aplicativo o tamanho desejado, assim o pacote enviado sofrerá o mínimo possível de cortes. O segundo problema é em relação ao áudio recebido, que apesar de bom, existe um eco de causa não identificada, causando a repetição da fala por alguns momentos.

A realização da conexão ponto-a-ponto entre os dispositivos não foi executada por dois motivos, primeiro que quando configurado o aparelho reiniciava sem causa aparente e segundo, a configuração ponto a ponto ficou por conta da configuração no próprio aparelho, dificultando a instalação e configuração de todo o aplicativo. Porém se utilizado dados de pacote para a realização da conexão, basta passar o IP atual do dispositivo servidor para os clientes realizarem a conexão.

Outro ponto positivo observado foi em relação à utilização do J2ME com *Datagram*. O aplicativo ficou altamente portátil para diversas plataformas, alguns sistemas operacionais testados incluem *Palm OS*, *Windows Mobile* e *Symbian 9.x*, mas é importante ressaltar que apenas em aparelhos *Symbian* foram realizados testes de conversação, nos outros o aplicativo foi instalado e configurado apenas. Quando o *Datagram* deu a possibilidade do aplicativo funcionar perfeitamente mesmo em dispositivos que não tenham o recurso Wireless, mas, podem efetuar uma conexão por outros meios como compartilhamento de Internet via *USB* ou *Bluetooth*. O aparelho com o sistema operacional *Palm OS* em questão é um caso, pois, trata-se de um *Treo 650* que não dispõe de rede sem fio, mas permite navegação por outros meios.

Apesar de todos os aspectos positivos apresentados, o aplicativo demonstrou alguns pontos negativos que deixam a desejar, dentre eles os principais inconvenientes são:

- a) A questão do IP que, a menos que seja tratado no roteador para mapear os endereços dos dispositivos, os usuários terão que repassar o IP servidor aos clientes sempre que forem realizar uma conexão;
- b) necessidade de deixar o aplicativo rodando o tempo inteiro. Nem todo dispositivo móvel possui um sistema operacional multitarefa, o que inviabiliza deixar o programa rodando em segundo plano em quanto utiliza outras funções do smartphone.

Em relação aos programas semelhantes no mercado atualmente como *Skype* e *Fring*, aplicativo ainda é muito imaturo para ser comercializado ou utilizado o dia-a-dia. Algumas diferenças que podem ser notadas incluem o nível de complexibilidade dos aplicativos, onde a codificação reduz a quase zero a perda de informações durante o processo de gravação e envio, principal problema encontrado na finalização do desenvolvimento, a diferença de estrutura, onde os aplicativos citados possuem uma lista de contatos com status que é mantida atualizada automaticamente, um identificador único que torna desnecessário o cliente descobrir o IP do dispositivo destino e a disponibilização de um servidor de registro para ter controle das conexões e cadastros dos usuários, além de login e senha para acesso, isso permite, por exemplo, de um usuário do *Skype* acesse sua conta de outra máquina e possa se comunicar com seus contatos sem a necessidade de recadastrá-los. Já no sistema desenvolvido, todo este cadastro fica armazenado localmente, inviabilizando o acesso em outros dispositivos, porém,

também fornecendo mais privacidade dos dados armazenados. A Tabela 2 demonstra um comparativo de alguns dos principais recursos dos aplicativos.

Tabela 2 – Comparativo entre o aplicativo Skype e VoIP

Aspecto / Aplicativo	Skype	VoIP
Facilidade de uso	Razoável	Bom
Qualidade da chamada	Bom	Bom
Número fixo	Sim	IP, não
Configuração	Automático	Manual, apenas na primeira vez
Padrão utilizado	Proprietário	Datagram
Extras	Vídeo chamada e IM*	-
Ligação para fixo	Sim	Não

Fonte: Adaptado de Martins (2009).

* IM, Instant messaging ou mensagem instantânea, conversação por mensagens de texto.

Na Figura 15 estão algumas das telas do aplicativo VoIP, da esquerda para a direita, a primeira representa a tela inicial, onde o usuário pode optar pela realização de uma ligação direta, consultar os contatos, configurar ou sair do aplicativo. Ao meio a lista de contatos com algumas opções como cadastrar novos, editar, excluir ou realizar uma ligação ao mesmo e por fim, a última representa a tela de conversação entre aplicativos, exibindo em cima o status da comunicação, o texto restante na tela não é importante, serve apenas para a realização de testes.



Figura 15. Aplicativo VoIP rodando em um N95.
Fonte: Autoria própria.

CONCLUSÃO

A grande finalidade deste projeto é demonstrar como é possível utilizar dos recursos existentes para criar aplicativos que podem ser úteis no dia-a-dia. A necessidade de comunicação é uma constante na vida das pessoas, porém, não são todas que possuem de recursos financeiros para contemplar esta necessidade em longo prazo. O aplicativo desenvolvido visa principalmente sua utilização dentro de organizações onde as pessoas trabalham dispersas, porém necessitam de comunicação constante sem a necessidade de gasto algum. Seja, por exemplo, uma empresa de consultoria onde, cada consultor cuida de um setor de seu cliente separadamente ou um campus de uma universidade onde os funcionários necessitam se comunicar. Até mesmo empresas que trabalham com representantes podem utilizar deste recurso como um canal de comunicação com a matriz ou com outros representantes independente de onde estejam (desde que haja uma rede sem fio disponível com conexão a Internet) sem necessidades de gastos de celular, linha telefônica, interurbano e outros.

Estudando os conceitos de VoIP e transmissão de dados, foi possível desenvolver um aplicativo em Java para dispositivos móveis que crie um canal de comunicação entre dois pontos e transmita voz permitindo assim a comunicação. A conexão realizada utilizando o *Datagram*, foi além do esperado, permitindo a utilização de outros recursos de conexão, como Internet compartilhada, no caso de dispositivos sem Wireless, ou dados de pacote. Apenas a realização de conexões

ponto-a-ponto não foi efetuada em virtude de problemas encontrados, o primeiro referente ao dispositivo que reiniciava durante o estabelecimento da conexão e em segundo a dependência do usuário de configurar a conexão.

Durante todo o desenvolvimento foram encontrados diversos obstáculos, os primeiros relacionados ao conhecimento e domínio da linguagem trabalhada foram sanados com horas de estudo e consultas ao livro de Muchow (2004) e a própria documentação da linguagem. Depois vieram os problemas com o protocolo de comunicação, a dificuldade de escolha e principalmente como utilizar cada um e qual que se adaptaria melhor. A consulta em trabalhos relacionados e pesquisas em fóruns ajudaram na base de conhecimento pessoal para poder implementar e testar cada um dos protocolos disponíveis encontrados, assim foi possível escolher aquele que demonstrou os melhores resultados. E por fim veio o maior dos problemas encontrados e mais difíceis solução, o áudio, a gravação e execução não foi de difícil implementação pela existência de diversos exemplos disponíveis na Internet, porém a sua transmissão foi realmente complicado. Em diversos testes o áudio não era enviado ou quando era, escutavam-se apenas ruídos no lado receptor, depois vieram os erros de rede no dispositivo sem causa aparente. Em pesquisas não foram encontrados registros nem exemplos de como enviar um *stream* de um dispositivo a outro, em consultas a fóruns também não foi apresentado ajuda. Então após uma árdua leitura da documentação existente e diversos testes realizados com configuração diferentes, obteve-se êxito na transmissão do áudio. O problema do envio estava relacionado

a duas questões, o tempo de gravação que não deve ser muito curto para gravar o suficiente e nem muito extenso, pois isso irá ultrapassar o tamanho do *Datagram*, enviando apenas metade ou menos das informações necessárias. A segunda questão é o tamanho do *Datagram* em si, que não deve extrapolar o máximo que o dispositivo comporta e nem menos que o tamanho do pacote que será enviado. Nos dispositivos de testes práticos a melhor configuração encontrada é um tempo de gravação de 2500 milissegundos com um *Datagram* de 40000 bytes, assim é possível um equilíbrio entre os dois parâmetros permitindo que haja uma boa transmissão dos dados com o mínimo de perda.

É claro, como já dito, o aplicativo ainda é muito imaturo para ser comercializado ou utilizado em campo. Por isso, é conhecido que alguns itens ainda tem que ser melhorados, alguns dos principais quesitos inclui:

- a) descobrir e resolver a causa do eco de comunicação existente na execução do áudio no lado receptor;
- b) desenvolver um meio de não precisar obter o IP dos contatos sempre que for realizar uma conexão, meios encontrados durante a pesquisa, porém não estudados a fundo incluem um servidor de registro para manutenção dos contatos ou a utilização da API *Wireless Messaging*, que permite a troca de informações utilizando o número do telefone destino;

- c) testes de consumo de bateria, como se trata de dispositivos de pouco poder computacional qualquer gasto excessivo representa uma grande desvantagem;
- d) melhorias gerais do aplicativo, como status de conexão para os contatos e conversação por mensagens de texto;
- e) configuração automática, identificando os recursos do aparelho em questão o aplicativo deverá se auto-configurar, evitando conhecimento técnico do usuário;
- f) utilização do protocolo *SIP* para comunicação, permitindo assim compatibilidade com outros aplicativos VoIP.

REFERÊNCIAS

BAZOTTI, Ezequiel. **VoIP para Computação Móvel**. 59 f. Dissertação (Trabalho de Conclusão de Curso) Departamento de Ciência da Computação - Universidade de Cruz Alta. Cruz Alta. 2007. Disponível em: <http://www.forum.nokia.com/piazza/wiki/images/5/51/TCC_EBazotti_FINAL_3.pdf>. Acesso em: 15 nov. 2009.

CAIRNCROSS, Frances. **O fim das distâncias**: como a revolução nas comunicações transformará nossas vidas. São Paulo: Nobel, 2000. 341 p.

CLICONNECT.COM. **Voz sobre IP em Redes Corporativas**. Disponível em: <<http://www.cliconnect.com/br/Artigos/VOIPRedesCorp/VoIPRedeCorp06.html>>. Acesso em: 7 dez. 2009.

COLCHER, Sérgio et al. **VoIP: Voz sobre IP**. 3. ed. Rio de Janeiro: Elsevier, 2005.

COMITÊ GESTOR DA INTERNET NO BRASIL. **Pesquisa Sobre o Uso das Tecnologias da Informação e da Comunicação no Brasil 2009**. São Paulo: [s.n.], 2010.

DAMASCENO, Nielsen C.; REGO, Joilson B. D. A. **Controlador Híbrido Neural no Controle de Nível do Protótipo de uma Planta Industrial Existente na Universidade Potiguar UnP**. Monografia Universidade Potiguar. Natal, p. 69. 2007. Disponível em: <<http://www.loopncd.hpg.com.br/Apostilas/Monografia1.pdf>>. Acesso em: 11 nov. 2010.

FLEURY, Vinícius V. **Avaliação Do Padrão GSM Como Solução Intermediária Na Migração Para Sistemas 3G (Terceira Geração)**. Dissertação (Trabalho de conclusão de curso) Escola de Engenharia Elétrica - Universidade Federal de Goiás. Goiânia, p. 88. 2004. Disponível em: <<http://www.ufg.br/this2/uploads/files/105/60278766.pdf>>. Acesso em: 20 nov. 2009.

GOOGLE MAPS. **Google Maps**. Disponível em: <<http://maps.google.com/>>. Acesso em: 22 jun. 2010.

GUGELMIN, Felipe. **Celulares: os serial killers do mundo da tecnologia.** Disponível em: <<http://www.baixaki.com.br/info/4346-celulares-os-serial-killers-do-mundo-da-tecnologia.htm>>. Acesso em: 02 out. 2010.

GUJ. Floggy: framework para persistência em J2ME/MIDP. Disponível em: <<http://www.guj.com.br/posts/list/67846.java>>. Acesso em: 12 nov. 2010.

HIGDON, Jim. **Coming Soon: Wifi VoIP.** Disponível em: <<http://www.voip-news.com/feature/coming-soon-wifi-voip-041707/>>. Acesso em: 9 dez. 2009.

IBGE. **Acesso à internet e posse de telefone móvel celular para uso pessoal.** Rio de Janeiro: [s.n.], 2007. 255 p.

JAVA MÓVEL. **Arquitetura J2ME.** Disponível em: <<http://www.javamovel.com/2009/05/arquitetura-j2me.html>>. Acesso em: 22 jun. 2010.

LIMA JÚNIOR, Almir W. **Tecnologias de redes & comunicação de dados.** Rio de Janeiro: Alta Books, 2002. 253 p.

MAHMOUD, Qusay H. **Getting Started with SIP API for J2ME (JSR 180).** Disponível em: <<http://developers.sun.com/mobility/apis/articles/sip/>>. Acesso em: 12 mar. 2010.

MARTINS, Vinicius. **Tabela comparativa de preços de algumas soluções VoIP nacionais.** Disponível em: <<http://vinicius.mobi/2009/02/28/tabela-comparativa-de-precos-de-algumas-solucoes-voip-nacionais/>>. Acesso em: 11 nov. 2010.

MINOLI, Daniel. **Voice Over IPv6.** [S.l.]: Elsevier, 2006. 366 p.

MUCHOW, John W. **Core J2ME: Tecnologia & MIDP.** 1. ed. São Paulo: Pearson Makron Books, 2004.

NOKIA. **Nokia N8.** Disponível em: <<http://www.nokia.com.br/produtos/celulares/nokia-n8>>. Acesso em: 22 jun. 2010.

PERLES, João B. **Comunicação: conceitos, fundamentos e história.** 17 f. Artigo - Faculdade de Selvíria. Selvíria. 2007. Disponível em: <<http://www.bocc.uff.br/pag/perles-joao-comunicacao-conceitos-fundamentos-historia.pdf>>. Acesso em: 11 nov. 2010.

PETERSON, Larry L.; DAVIE, Bruce S. **Redes de computadores:** uma abordagem de sistemas. Rio de Janeiro: Elsevier, 2004. 587 p.

RISCHPATER, Ray. **Desenvolvendo wireless para web:** como enfrentar os desafios dos projetos para web sem fio. São Paulo: Makron Books, 2001. 345 p.

ROSS, Catanzariti. **The Mobile Phone:** A History in Pictures. Disponível em: <http://www.pcworld.com/article/180678/first_car_phone_nokia_mobira_senator_1982.html>. Acesso em: 22 jun. 2010.

SMARTPHONE CELULAR. **Smartphones x Celulares:** Qual é a Diferença? Disponível em: <<http://smartphonecelular.com.br/smartphone-x-celular-qual-e-a-diferenca>>. Acesso em: 02 out. 2010.

VAZ, Igor; DINAU, Priscilla. **SIP:** Vantagens e Desvantagens. Disponível em: <http://www.gta.ufrj.br/grad/06_1/sip/VantagenseDesvantagens.html>. Acesso em: 02 out. 2010.

VOIPBRA. **Voip:** A Internet e o TCP/IP. Disponível em: <<http://www.voipbra.com.br/voip.htm>>. Acesso em: 10 nov. 2010.

REFERÊNCIAS COMPLEMENTARES

- G1. **Brasil tem torpedo mais caro da América Latina.** Disponível em: <[http://g1.globo.com/bomdiabrasil/0, MUL1122535-16020,00-BRASIL+TEM+TORPEDO+MAIS+CARO+DA+AMERICA+LATINA.htm](http://g1.globo.com/bomdiabrasil/0,MUL1122535-16020,00-BRASIL+TEM+TORPEDO+MAIS+CARO+DA+AMERICA+LATINA.htm)>. Acesso em: 12 nov. 2009.
- HORD, Jennifer. **Como funciona o SMS.** Disponível em: <<http://informatica.hsw.uol.com.br/sms.htm>>. Acesso em: 4 nov. 2009.
- JAVA MÓVEL. **Configurando o Netbeans para usar Floggy.** Disponível em: <<http://www.javamovel.com/2009/06/configurando-o-netbeans-para-usar.html>>. Acesso em: 22 jun. 2010.
- LI, Sing; KNUDSEN, Jonathan. **Beginning J2ME: From Novice to Professional.** 3. ed. Nova Iorque: Apress, 2005.
- MOREIRA, Thiago L.; ROSSATO, Thiago; LUGON, Priscila. **Floggy.** Disponível em: <<http://floggy.sourceforge.net/>>. Acesso em: 22 jun. 2010.
- PELEGRINA, José A. **Dicionário de informática:** Letra: P. Disponível em: <<http://www.dicweb.com>>. Acesso em: 24 maio 2010.
- PIROUMIAN, Vartan. **Wireless J2ME: Platform Programming.** Nova Iorque: Sun Microsystems Press, 2005. Nova Iorque: Sun Microsystems Press, 2002.
- RAPOSO, Álvaro R. D. M. **Comunicação entre dispositivos de baixo poder computacional: Bluetooth como protocolo de comunicação para jogos para celular.** 35 f. Dissertação (Trabalho de Conclusão de Curso) Curso de Engenharia de Comunicação - Instituto de Ensino Superior Fucapi. Manaus. 2005. Disponível em: <<http://portal.fucapi.edu.br/arquivos/biblioteca/ecm/alvaroramos.pdf>>. Acesso em: 15 dez. 2009.
- ROSS, Julio. **VoIP Voz Sobre IP.** Rio de Janeiro: Antenna Edições Técnicas Ltda, 2007.
- SANTOS, Daniel D. L. **Estudo de soluções VoIP baseadas em Software Livre.** 73 f. Dissertação (Trabalho de Conclusão de Curso) Departamento de Centro de Ciências Tecnológicas - Fundação Edson Queiroz, Universidade de Fortaleza.

Ceará. 2006. Disponível em:

<http://www.dnocs.gov.br/php/util/downloads_file.php?&dir=&file=/home/util/livres/ebooks/monografias/estudos_de_solucoes_voip_baseadas_em_software_livre.pdf&>. Acesso em: 15 nov. 2009.

SUN MICROSYSTEMS. **MIDP APIs for Wireless Applications: A Brief Tour for Software Developers**. Disponível em:

<<http://java.sun.com/products/midp/midp-wirelessapps-wp.pdf>>. Acesso em: 09 dez. 2009.

_____. **Sun Java™ Wireless Toolkit for CLDC: Basic Customization Guide**. Disponível em:

<<http://java.sun.com/products/sjwtoolkit/wtk2.5.2/docs/BasicCustomizationGuide.pdf>>. Acesso em: 09 dez. 2009.

_____. **Sun Java™ Wireless Client Software 2.2: Architecture and Design Guide**, 250 f. Disponível em: <<http://java.sun.com/javame/reference/docs/sjwc-2.2/pdf-html/architecture.pdf>>. Acesso em: 9 dez. 2009.

_____. **Sun Java™ Wireless Client Software 2.2: Build Guide**, 74 f.

Disponível em: <<http://java.sun.com/javame/reference/docs/sjwc-2.2/pdf-html/build.pdf>>. Acesso em: 09 dez. 2009.

_____. **Sun Java™ Wireless Client Software 2.2: Multitasking Guide**, 38 f.

Disponível em: <<http://java.sun.com/javame/reference/docs/sjwc-2.2/pdf-html/multitasking.pdf>>. Acesso em: 09 dez. 2009.

_____. **Sun Java™ Wireless Client Software 2.2: Porting Guide**, 164 f.

Disponível em: <<http://java.sun.com/javame/reference/docs/sjwc-2.2/pdf-html/porting.pdf>>. Acesso em: 09 dez. 2009.

_____. **Sun Java™ Wireless Client Software 2.2: Skin Author's Guide to Adaptive User Interface Technology**, 90 f. Disponível em:

<<http://java.sun.com/javame/reference/docs/sjwc-2.2/pdf-html/skin.pdf>>. Acesso em: 09 dez. 2009.

_____. **Sun Java™ Wireless Toolkit for CLDC: User's Guide**. Disponível

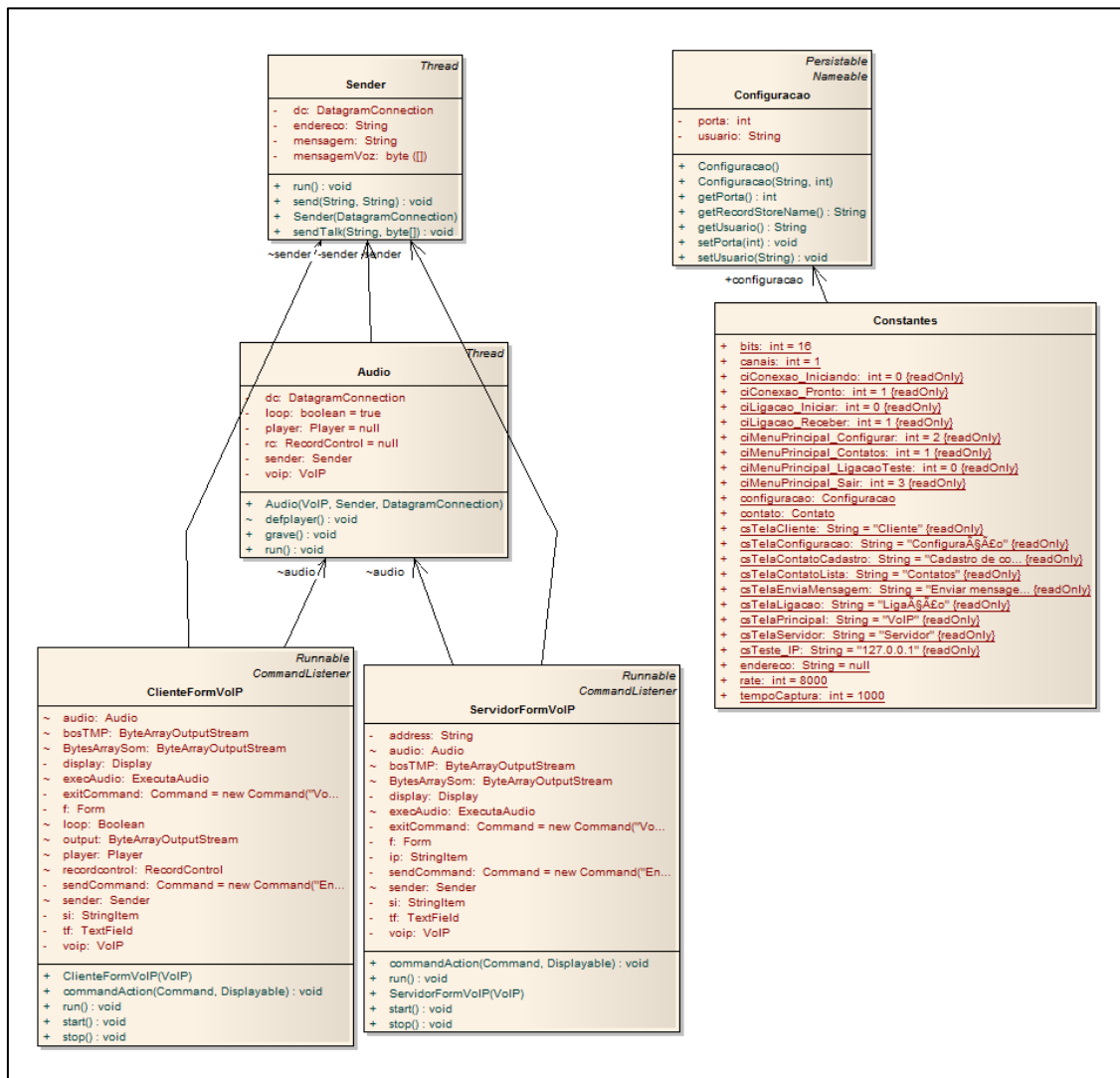
em: <<http://java.sun.com/products/sjwtoolkit/wtk2.5.2/docs/UserGuide.pdf>>. Acesso em: 09 dez. 2009.

UCEL. **Torpedo (SMS)**. Disponível em: <<http://www.ucel.com.br/sms.asp>>. Acesso em: 11 nov. 2009.

WANDERLEY, Márcio. **Vivo tem menor tarifa para pré-pago e a Claro, para pós-pago**. Disponível em: <<http://www.setimoceu.com.br/modules/planet/view.article.php/86732>>. Acesso em: 22 set. 2009.

WATANABE, Gerson T. **Aplicação de Voip Na Convergência de Dados e Voz Corporativos**. 113 f. Dissertação (Trabalho de Conclusão de Curso) Departamento Acadêmico de Eletrônica - Universidade Tecnológica Federal do Paraná. Curitiba. 2006. Disponível em: <<http://www.scribd.com/doc/22752483/APLICACAO-DE-VOIP-NA-CONVERGENCIA-DE-DADOS-E-VOZ-CORPORATIVOS>>. Acesso em: 9 dez. 2009.

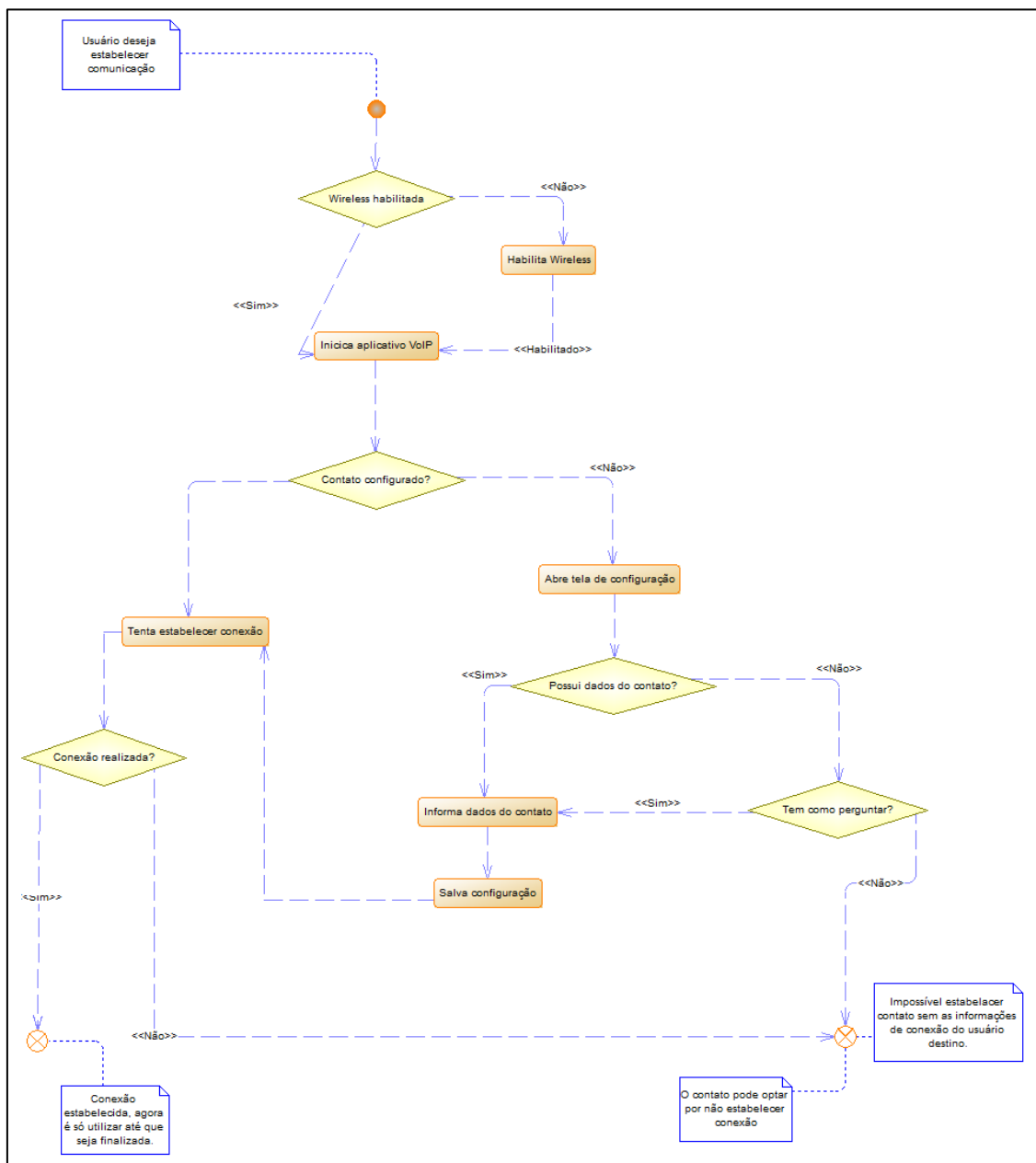
APÊNDICE A – DIAGRAMA DE CLASSES



APÊNDICE A - CONTINUAÇÃO



APÊNDICE B – DIAGRAMA DE ATIVIDADE DE CONEXÃO



APÊNDICE C – ARTIGO

Aplicação VoIP por meio da Tecnologia Wireless

Murilo Garcia Bento¹, Paulo João Martins²

¹Acadêmico do curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brasil

1 ²Professor(a) do curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brasil

2 jedizero@msn.com, pjm@unesc.net

Abstract. *VoIP is gaining prominence nowadays, a proof of this is the most current smartphones that already come with support for such an appeal, but its configuration is still all manual requiring a degree of knowledge from its user. Based on these assumptions, emerged the idea to develop an easy to use application that has the same functionality. Developed in Java for mobile devices using the Datagram protocol in conjunction with the native libraries of multimedia, can establish a connection between two devices via a wireless access point, avoiding spending mainly with telephone line and thus creating an easy-use and low cost alternative*

Resumo. *VoIP está ganhando destaque atualmente, um prova disso são os smartphones mais atuais que já estão vindo com suporte a tal recurso, porém, sua configuração ainda é toda manual exigindo um certo grau de conhecimento do usuário. Baseado nestes fundamentos é que foi alavancada a idéia de desenvolver um aplicativo de fácil utilização de que tenha a mesma funcionalidade. Desenvolvido em Java para dispositivos móveis utilizando o protocolo Datagram em conjunto com as bibliotecas nativas de multimídias, consegue estabelecer uma conexão entre dois dispositivos por meio de um ponto de acesso sem fio, evitando principalmente gastos com linha telefônica e criando assim uma alternativa de fácil utilização e a baixo custo.*

1. Introdução

Segundo Colcher, et al. (2005) o processo de convergência das tecnologias de telecomunicações e processamento de informações (via sistemas computacionais) tem promovido profundas alterações na organização do trabalho na indústria, no comércio, na prestação de serviços, nas pesquisas, na vida particular do cidadão,

na saúde e na educação. Dentre os maiores representantes desses processos figuram os serviços multimídia, hoje presentes em dispositivos móveis e celulares, e a proliferação de serviços variados sobre a infraestrutura de redes de computadores como a Internet.

Porém, mesmo com a expansão do mercado consumidor e a sua gradativa evolução, ainda existem custos de pesquisas, testes e divulgação, além de muitas outras variantes existentes no processo de desenvolvimento de novas tecnologias. A somatória de todos estes custos gera valores de comércio, muitas vezes, inacessíveis a maioria das pessoas, por isso, o estudo e utilização de tecnologias que vem ganhando mercado a cada dia, como a Wireless e o J2ME resultará na implementação de um aplicativo para dispositivos móveis com suporte a J2ME, que tem como principal função permitir a comunicação a um custo mais acessível às pessoas via VoIP por meio de Wireless quando houver um ponto de acesso a Internet sem fio dentro do alcance.

2. Comunicação

Derivado do latim *comunicare*, cujo significado seria “tornar comum”, “repartir, associar”, “trocar opiniões”, “conferenciar”. Porém a comunicação de hoje conhecida, assim como muitas outras atividades passou por um processo evolutivo. Segundo Perles (2007) as primeiras formas de diálogo eram expressas por gestos, imagens e sons, somente mais tarde foi desenvolvido a capacidade de utilizar sinais gráficos, neste ponto houve uma evolução tanto na conversação como na cultura.

A compreensão de comunicação abrange todo o processo, desde o envio, recebimento e entendimento de uma informação.

2.1. Comunicação móvel

A evolução no modelo de comunicação também permitiu a evolução dos aparelhos sem fio. Celular, dispositivo eletrônico muito utilizado atualmente, teve seu nome originado no novo modelo de células (célula → celular). Dentre as diversas causas que permitiram que este aparelho se tornasse algo tão famoso e comum na sociedade atual, é possível citar o inconveniente da troca de endereço ou emprego, onde pode ocasionar na troca do número fixo. Outra característica dos aparelhos celulares é o fato de poder ser levado a qualquer lugar junto ao usuário, considerando que o local tenha cobertura da empresa de telefonia escolhida, é possível fazer e receber ligações onde estiver.

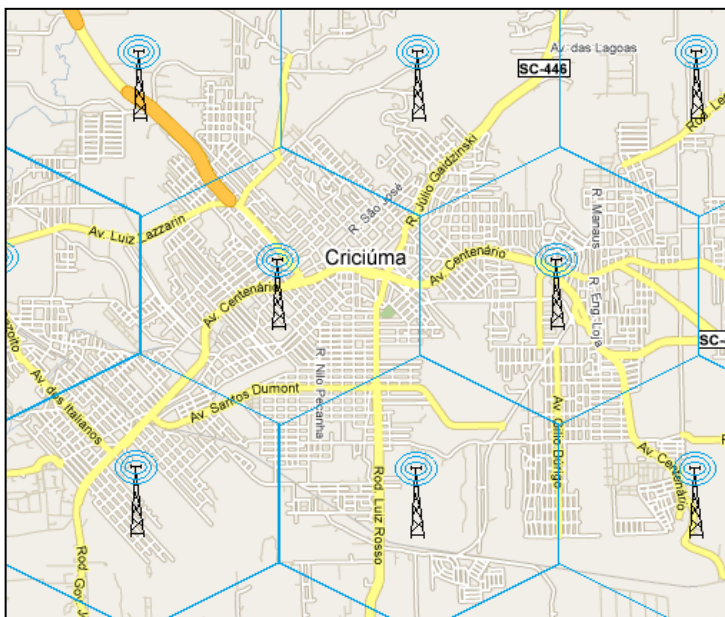


Figura 1. Modelo de células

3. Tecnologias de desenvolvimento

Para o desenvolvimento de aplicativos para dispositivos móveis existem as mais diversas ferramentas, IDE ou Ambiente Integrado de Desenvolvimento, e SDK, Kit de Desenvolvimento de Software. Muitas fornecidas ou recomendadas pelo próprio fabricante do aparelho e outras optadas por escolha individual. Cada sistema operacional dos smartphones tem as suas especificações e restrições quanto à linguagem que será utilizada para o desenvolvimento, dentro de um Windows Mobile, por exemplo, temos .NET ou Java, já para um Symbian, sistema operacional da Nokia, os aplicativos podem ser desenvolvidos na linguagem Symbian C++ ou em Java.



Figura 2. iPhone 4 com IOS e o novo

Nokia N8 com Symbian^3

3.1. Java

Desenvolvida na década de 90, pela Sun Microsystems, o principal objetivo de seus desenvolvedores foi a criação de uma linguagem simples e que permitisse ser escrita apenas uma vez e pudesse ser executada em qualquer plataforma disponível no mercado.

Segundo Damasceno e Rego (2007, p. 44) “a JavaSoft, empresa do grupo Sun Microsystems que [...] controla a compatibilidade e o desenvolvimento das sucessivas versões do Java, [...]”. Essas distribuições contêm desde ferramentas até conjuntos de API, também conhecido como interface de programação de aplicativos que servem para o desenvolvimento. J2ME, Java2 Micro Edition, é grupo que contém todo esse pacote para a criação de aplicativos para dispositivos de pouco poder computacional (MUCHOW, 2004).

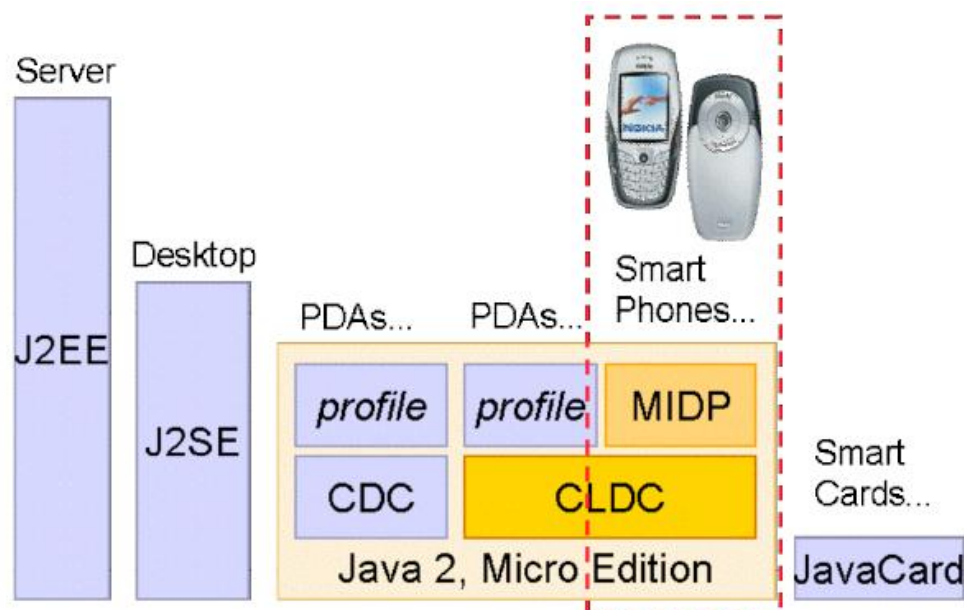


Figura 3. Arquitetura Java

3.2. Java2 Micro Edition

A versão micro do Java, nasceu da necessidade do desenvolvimento de aplicativos para dispositivos de pouco poder computacional, como pagers e até celulares. Caracterizado pela necessidade mínima de recursos para a sua execução. A máquina virtual do J2ME, a KVM, Máquina Virtual K, também exige pouco

poder de processamento, podendo, por exemplo, ser executada em processadores de 16 bits com apenas de 25MHz de frequência.

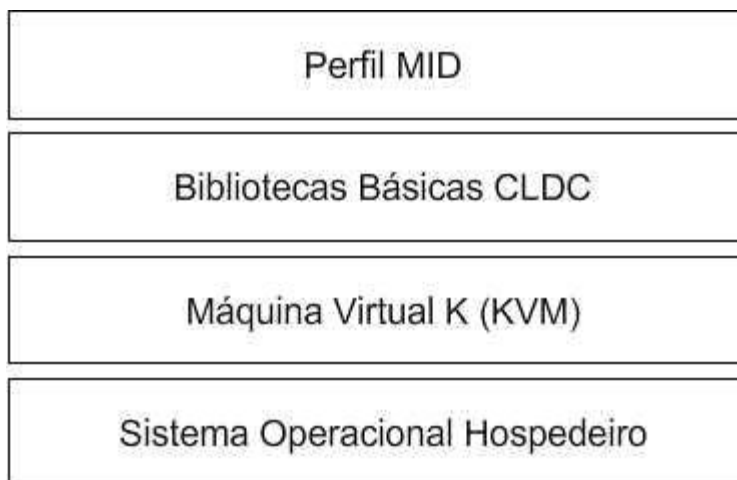


Figura 4. Arquitetura Java

4. Redes de comunicação

De acordo com Colcher, et al. (2005), pode-se dizer que uma rede, é dois ou mais dispositivos que trocam informações por meio de um sistema de comunicação, neste caso, existe a necessidade de ambos enviarem e receberem as informações de modo entendível. Estes dispositivos são chamados de Módulos Processadores (MPs), pois ambos têm a capacidade de trocar informações através de um sistema de comunicação, logo, um MPs pode ser, por exemplo, desde um computador até um telefone.

4.1. Voz sobre IP

Proposto inicialmente como um modo de comunicação entre computadores, mais tarde, na medida em que foi evoluindo, sua área de atuação também aumentou. Segundo Minoli (2006) em meados do ano 2000 é que começaram a surgir os primeiros sistemas comerciais VoIP em cima do modelo 1G, funcionamento de forma interligada ao serviço telefônico comercial. A partir de então com a expansão da Internet e aumento das taxas de transmissão, começaram a surgir empresas produzindo em larga escala equipamentos específicos para o VoIP, como roteadores, aparelhos telefônicos (telefones IP), adaptadores e outros. Como resultado desta evolução, ele também sofreu grandes melhorias em sua qualidade, assim com também começaram a surgir os primeiros padrões definidos pela ITU, International Telecommunications Union e IETF, Internet Engineering Task Force.

4.2. Redes sem fio - Wireless

As redes sem fio se tornaram popular, principalmente pela necessidade de colocar computadores em uma rede de dentro de um escritório e a inviabilidade da passagem de cabos. As redes sem fio, como o próprio nome sugere, dispensa a necessidade de se espalhar cabos pelo ambiente, é fácil de utilizar e seu custo acaba sendo muito mais acessível uma vez que não há necessidade da aquisição de cabos ou roteadores, mesmo quando se deseja aumentar o número de máquinas na rede, basta configurar o ponto de acesso e pronto. Segundo RISCHPATER (2001), além da facilidade de acesso, as redes sem fio permite que outros dispositivos além de computadores sejam conectados, como Smartphones.

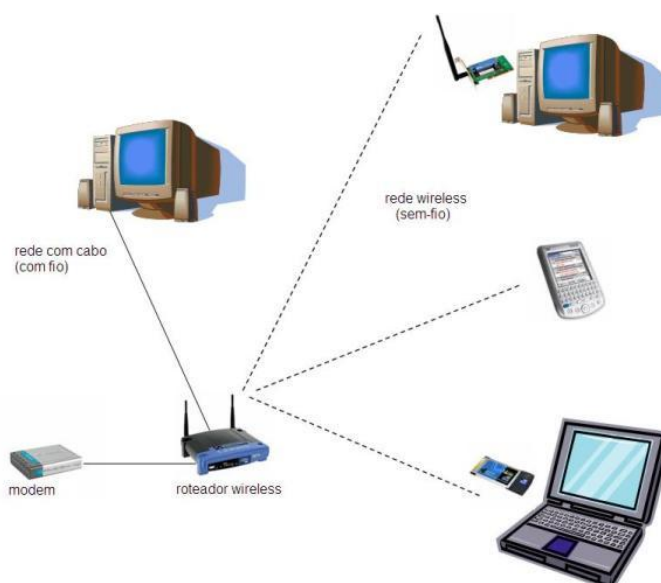


Figura 5. Rede Wireless

5. Resultados do aplicativo

O aplicativo desenvolvido demonstrou estabilidade de funcionamento, tanto na parte do servidor quanto na parte do cliente. Todos foram realizados dentro de uma rede local e contaram com a utilização de um access point para a distribuição do sinal Wireless. Os dispositivos móveis A e B foram adicionados à rede e iniciaram a conexão, o primeiro como servidor e o segundo como cliente passando o IP do servidor A como parâmetro.

A conexão e o envio dos dados tiveram bons resultados, pois o tempo de resposta do aplicativo é hábil. Mesmo a execução do áudio teve qualidade considerável. Foram detectados dois grandes problemas, o primeiro, foi em relação à gravação do áudio para posterior envio. Como a programação delimita

um tempo de gravação, para depois enviar e somente depois iniciar a nova gravação, esse desgaste causa perda de informações durante a fala enviando apenas metade do que deveria. A solução encontrada foi à customização do tempo de gravação na tela de configuração, outro fator importante é o tamanho do Datagram que será recebido, quando nas primeiras versões era utilizado um valor fixo, na versão mais atual é possível configurar no aplicativo o tamanho desejado, assim o pacote enviado sofrerá o mínimo possível de cortes. O segundo problema é em relação ao áudio recebido, que apesar de bom, existe um eco de causa não identificada, causando a repetição da fala por alguns momentos.



Figura 6. Aplicativo rodando em um Smartphone

A realização da conexão ponto-a-ponto entre os dispositivos não foi executada por dois motivos, primeiro que quando configurado o aparelho reiniciava sem causa aparente e segundo, a configuração ponto a ponto ficou por conta da configuração no próprio aparelho, dificultando a instalação e configuração de todo o aplicativo. Porém se utilizado dados de pacote para a realização da conexão, basta passar o IP atual do dispositivo servidor para os clientes realizarem a conexão.

Em relação aos programas semelhantes no mercado atualmente como Skype e Fring, aplicativo ainda é muito imaturo para ser comercializado ou utilizado o dia-a-dia. Algumas diferenças que podem ser notadas incluem o nível de complexibilidade dos aplicativos, onde a codificação reduz a quase zero a perda de informações durante o processo de gravação e envio, principal problema encontrado na finalização do desenvolvimento, a diferença de estrutura, onde os aplicativos citados possuem uma lista de contatos com status que é mantida atualizada automaticamente, um identificador único que torna desnecessário o cliente descobrir o IP do dispositivo destino e a disponibilização de um servidor de registro para ter controle das conexões e cadastros dos usuários, além de login e senha para acesso, isso permite, por exemplo, de um usuário do Skype acesse sua conta de outra máquina e possa se comunicar com seus contatos sem a

necessidade de cadastrá-los. Já no sistema desenvolvido, todo este cadastro fica armazenado localmente, inviabilizando o acesso em outros dispositivos, porém, também fornecendo mais privacidade dos dados armazenados.

6. Conclusão

Estudando os conceitos de VoIP e transmissão de dados, foi possível desenvolver um aplicativo em Java para dispositivos móveis que crie um canal de comunicação entre dois pontos e transmita voz permitindo assim a comunicação. A conexão realizada utilizando o Datagram, foi além do esperado, permitindo a utilização de outros recursos de conexão, como Internet compartilhada, no caso de dispositivos sem Wireless, ou dados de pacote. Apenas a realização de conexões ponto-a-ponto não foi efetuada em virtude de problemas encontrados, o primeiro referente ao dispositivo que reiniciava durante o estabelecimento da conexão e em segundo a dependência do usuário de configurar a conexão.

Durante todo o desenvolvimento foram encontrados diversos obstáculos, os primeiros relacionados ao conhecimento e domínio da linguagem trabalhada foram sanados com horas de estudo e consultas ao livro de Muchow (2004) e a própria documentação da linguagem. Depois vieram os problemas com o protocolo de comunicação, a dificuldade de escolha e principalmente como utilizar cada um e qual que se adaptaria melhor. A consulta em trabalhos relacionados e pesquisas em fóruns ajudaram na base de conhecimento pessoal para poder implementar e testar cada um dos protocolos disponíveis encontrados, assim foi possível escolher aquele que demonstrou os melhores resultados. E por fim veio o maior dos problemas encontrados e mais difíceis solução, o áudio, a gravação e execução não foi de difícil implementação pela existência de diversos exemplos disponíveis na Internet, porém a sua transmissão foi realmente complicado. Em diversos testes o áudio não era enviado ou quando era, escutavam-se apenas ruídos no lado receptor, depois vieram os erros de rede no dispositivo sem causa aparente. Em pesquisas não foram encontrados registros nem exemplos de como enviar um stream de um dispositivo a outro, em consultas a fóruns também não foi apresentado ajuda. Então após uma árdua leitura da documentação existente e diversos testes realizados com configuração diferentes, obteve-se êxito na transmissão do áudio. O problema do envio estava relacionado a duas questões, o tempo de gravação que não deve ser muito curto para gravar o suficiente e nem muito extenso, pois isso irá ultrapassar o tamanho do Datagram, enviando apenas metade ou menos das informações necessárias. A segunda questão é o tamanho do Datagram em si, que não deve extrapolar o máximo que o dispositivo comporta e nem menos que o tamanho do pacote que será enviado. Nos dispositivos de testes práticos a melhor configuração encontrada é um tempo de gravação de 2500 milissegundos com um Datagram de 40000 bytes, assim é possível um equilíbrio entre os dois parâmetros permitindo que haja uma boa transmissão dos dados com o mínimo de perda.

7. Referências

- Colcher, Sérgio et al. VoIP: Voz sobre IP. 3. ed. Rio de Janeiro: Elsevier, 2005.
- Damasceno, Nielsen C.; REGO, Joilson B. D. A. Controlador Híbrido Neural no Controle de Nível do Protótipo de uma Planta Industrial Existente na Universidade Potiguar UnP. Monografia Universidade Potiguar. Natal, p. 69. 2007. Disponível em: <<http://www.loopncd.hpg.com.br/Apostilas/Monografia1.pdf>>. Acesso em: 11 nov. 2010.
- Minoli, Daniel. Voice Over IPv6. [S.l.]: Elsevier, 2006. 366 p.
- Muchow, John W. Core J2ME: Tecnologia & MIDP. 1. ed. São Paulo: Pearson Makron Books, 2004.
- Perles, João B. Comunicação: conceitos, fundamentos e história. 17 f. Artigo - Faculdade de Selvíria. Selvíria. 2007. Disponível em: <<http://www.bocc.uff.br/pag/perles-joao-comunicacao-conceitos-fundamentos-historia.pdf>>. Acesso em: 11 nov. 2010.
- Rischpater, Ray. Desenvolvendo wireless para web: como enfrentar os desafios dos projetos para web sem fio. São Paulo: Makron Books, 2001. 345 p.

ANEXO A – ATIVIDADES REALIZADAS PELO TELEFONE CELULAR

J5 – ATIVIDADES REALIZADAS PELO TELEFONE CELULAR			
ACTIVITIES PERFORMED ON THE MOBILE PHONE			
Percentual sobre o total de pessoas que utilizam telefone celular ¹			
Percentage over the total number of people who use mobile phones ¹			
Percentual (%) Percentage (%)	Para acessar a Internet Accessing the Internet	Para enviar e/ou receber mensagens de texto (SMS/mensagens torpedos) Sending and/or receiving text messages (SMS)	Para enviar e/ou receber fotos e imagens Sending and/or receiving photos and images
TOTAL	5	58	24
Área urbana <i>Urban area</i>	6	59	25
Área rural <i>Rural area</i>	3	45	15
REGIÕES DO PAÍS REGION			
Sudeste <i>Southeast</i>	7	60	29
Nordeste <i>Northeast</i>	3	46	17
Sul <i>South</i>	4	67	23
Norte <i>North</i>	4	58	20
Centro-Oeste <i>Center-West</i>	5	58	25
SEXO GENDER			
Masculino <i>Male</i>	6	56	24
Feminino <i>Female</i>	4	59	24
GRAU DE INSTRUÇÃO SCHOOLING			
Analfabeto/ Educação infantil <i>Illiterate/ Kindergarten</i>	1	31	7
Fundamental <i>Primary</i>	4	56	21
Médio <i>Secondary</i>	7	70	31
Superior <i>Tertiary</i>	12	79	43
FAIXA ETÁRIA AGE			
10 - 15	5	58	26
16 - 24	9	78	39
25 - 34	7	71	31
35 - 44	3	51	18
45 - 59	2	37	9
60 +	1	24	4
RENDA FAMILIAR INCOME			
< R\$ 465,00	2	40	12
R\$ 466,00 - R\$ 930,00	3	53	18
R\$ 931,00 - R\$ 1.395,00	4	62	26
R\$ 1.396,00 - R\$ 2.325,00	6	66	33
R\$ 2.326,00 - R\$ 4.650,00	11	74	38
R\$ 4.651,00 +	22	76	44
CLASSE SOCIAL SOCIAL CLASS			
A	23	76	55
B	10	70	35
C	4	60	24
DE	2	40	12
SITUAÇÃO DE EMPREGO EMPLOYMENT STATUS			
Trabalhador <i>Worker</i>	6	61	26
Desempregado <i>Unemployed</i>	5	55	23
Não integra a população ativa <i>Is not part of the active population</i>	4	51	20

¹ Base: 15.007 entrevistados que utilizaram telefone celular nos últimos três meses. Respostas múltiplas e estimuladas.

¹ Base: 15.007 interviewees who used a mobile phone in the past three months. Multiple and stimulated answers.

Fonte / Source: NIC.br - set/nov (set/nov) 2009

ANEXO B – PROPORÇÃO DE INDIVÍDUOS QUE POSSUEM TELEFONE CELULAR COM ACESSO À INTERNET

J4 – PROPORÇÃO DE INDIVÍDUOS QUE POSSUEM TELEFONE CELULAR COM ACESSO À INTERNET			
<i>PROPORTION OF INDIVIDUALS WHO OWN MOBILE PHONES WITH INTERNET ACCESS</i>			
Percentual sobre o total de pessoas que possuem telefone celular ¹			
<i>Percentage over the total number of people who own mobile phones¹</i>			
Percentual (%) Percentage (%)	Sim Yes	Não No	Não sabe/ Não respondeu Does not know/ Did not answer
TOTAL	35	62	3
Área urbana <i>Urban area</i>	36	61	3
Área rural <i>Rural area</i>	27	70	3
REGIÕES DO PAÍS REGION			
Sudeste <i>Southeast</i>	41	55	4
Nordeste <i>Northeast</i>	23	74	3
Sul <i>South</i>	39	59	2
Norte <i>North</i>	32	67	2
Centro-Oeste <i>Center-West</i>	27	71	1
SEXO GENDER			
Masculino <i>Male</i>	38	62	3
Feminino <i>Female</i>	35	62	4
GRAU DE INSTRUÇÃO SCHOOLING			
Analfabeto/ Educação infantil <i>Illiterate/ Kindergarten</i>	15	78	7
Fundamental <i>Primary</i>	30	67	3
Médio <i>Secondary</i>	42	56	2
Superior <i>Tertiary</i>	54	45	1
FAIXA ETÁRIA AGE			
10 - 15	37	62	1
16 - 24	50	49	1
25 - 34	43	56	2
35 - 44	31	66	2
45 - 59	20	74	6
60 +	9	80	12
RENDA FAMILIAR INCOME			
< R\$ 465,00	19	74	7
R\$ 466,00 - R\$ 930,00	28	68	3
R\$ 931,00 - R\$ 1.395,00	35	63	2
R\$ 1.396,00 - R\$ 2.325,00	44	53	3
R\$ 2.326,00 - R\$ 4.650,00	49	48	3
R\$ 4.651,00 +	60	38	2
CLASSE SOCIAL SOCIAL CLASS			
A	60	38	2
B	48	50	2
C	34	63	2
DE	18	75	7
SITUAÇÃO DE EMPREGO EMPLOYMENT STATUS			
Trabalhador <i>Worker</i>	38	59	3
Desempregado <i>Unemployed</i>	32	64	4
Não integra a população ativa <i>Is not part of the active population</i>	28	67	4

Fonte (Source): NIC.br - set/nov (set/nov) 2009

¹ Base: 11.871 entrevistados que possuem telefone celular.

¹ Base: 11.871 interviewees who owned mobile phones.

Fonte: Comitê Gestor da Internet no Brasil (2010)

ANEXO C – CONFIGURANDO O NETBEANS PARA USAR FLOGGY

Configurando o Netbeans para usar Floggy

Categorias: [Códigos fonte](#), [Frameworks](#)

Para usar o **Floggy** é necessário fazer o seu download e depois configurá-lo de acordo com o ambiente que utilizará para o desenvolvimento da sua aplicação.

Para fazer o download do Floggy [clique aqui](#).

Agora mostraremos como configurar o Netbeans para usar este *framework*.

Para configurar outros ambientes veja o [site oficial do Floggy](#).

Para usar o **Floggy no Netbeans** são necessários 2 passos: Configurar o classpath e editar o build.xml.

Configurando o classPath

Vá nas propriedades do seu projeto (clcando com o botão direito sobre ele) e nas categorias selecione Bibliotecas & Recursos, em adicionar jar/zip procure na pasta que acabou de baixar do Floggy o arquivo **floggy-persistence-framework.jar**, que está na pasta lib.

Uma outra maneira de fazer isto é clicando com o botão direito sobre Recursos (dentro do seu projeto).

Editando o build.xml

Abra o **build.xml** do seu projeto e adicione as linhas abaixo, as que estão depois de "-->".

ANEXO C – CONTINUAÇÃO

```
view plain print ?
01. <!--
02.     IMPORTANT: You must change the value of this property pointing it to the di
03.     where you have installed Floggy.
04.
05.     Some examples:
06.     Linux: /opt/floggy
07.     Windows: c:\floggy
08. -->
09. <property name="floggy.path" value="c:\floggy"/>
10.
11. <path id="floggy.classpath">
12.     <fileset dir="{floggy.path}\lib" includes="*.jar"/>
13. </path>
14.
15. <target name="post-compile">
16.     <taskdef
17.         name="floggy-persistence-weaver"
18.         classname="net.sourceforge.floggy.persistence.WeaverTask"
19.         classpathref="floggy.classpath"/>
20.     <floggy-persistence-weaver
21.         bootclasspath="{platform.bootclasspath}"
22.         classpath="{libs.classpath}"
23.         input="{build.classes.dir}"
24.         output="{build.classes.dir}"/>
25. </target>
```

Em property mude o value para o diretório onde você colocou a pasta do Floggy.

No código, em dir já está indicado que os arquivos necessários estão na pasta lib. Então não é preciso colocá-la no caminho acima.

A única modificação que tem que fazer é esta, informando o caminho correto. O restante é só copiar e a sua aplicação estará pronta para ser executada.

No próximo post apresentaremos um **exemplo prático** da utilização de **Floggy**, uma agenda.

Para maiores informações veja o [site do projeto](#).