UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN

CENTRO DE CIÊNCIAS EXATAS E DA TERRA – CCET

DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA – DIMAP

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO

MOBILE INTERACT: Uma ferramenta de auxílio didático proporcionando interação entre professores e alunos através de celulares.

KAIO ALENCAR DE AZEVEDO DANTAS

Natal – RN

Abril / 2010

KAIO ALENCAR DE AZEVEDO DANTAS

MOBILE INTERACT: Uma ferramenta de auxílio didático proporcionando interação entre professores e alunos através de celulares.

Dissertação apresentada ao Programa de Pós-Graduação em Sistema e Computação da UFRN como requisito para obtenção do título de mestre.

Orientadores: Prof. Ivan Saraiva Silva (orientador)
Prof. Aquiles Medeiros Filgueira Burlamaqui (co-orientador)

Natal – RN
Abril / 2010

KAIO ALENCAR DE AZEVEDO DANTAS

MOBILE INTERACT: Uma ferramenta de auxílio didático proporcionando interação entre professores e alunos através de celulares.

Aprovada em __/__/__

BANCA EXAMINADORA

_____

Adilson Barboza Lopes

Examinador

_____

Tatiana Aires Tavares

Examinador

_____

Aquiles Medeiros Filgueira Burlamaqui

Co-Orientador

_____

Ivan Saraiva Silva

Orientador

Dedico este trabalho à minha família, principalmente aos meus pais José Alencar e Amália, e aos amigos que me apoiaram nas dificuldades e me incentivaram diante de todos os problemas que passei e que somente quem esteve mais próximo pôde ajudar a superá-los.

AGRADECIMENTOS

Em primeiro lugar aos meus pais e meus irmãos pelo apoio e incentivo em todos os sentidos e que, mesmo com a distância, sempre estiveram ao meu lado oferecendo todas as condições possíveis para que eu conseguisse alcançar meus objetivos.

Aos meus colegas da UFRN, tanto da minha turma da graduação quanto de outras turmas e cursos, principalmente os desenvolvedores da Natalnet, que compartilharam comigo algumas experiências e conhecimentos, ajudando no desenvolvimento e solução de alguns problemas da minha dissertação.

A todos os colegas, amigos e estudantes das turmas dos cursos de computação e de Ciências e Tecnologia por participarem dos testes feitos nesse trabalho, emprestando seus celulares e submetendo-se aos testes respondendo algumas questões.

A Ivan Saraiva e a Aquiles Burlamaqui por aceitarem ser meus orientadores, por terem sido pacientes comigo e por ajudarem na conclusão dessa dissertação.

Aos meus amigos que sabiam das minhas responsabilidades e problemas e me incentivaram no desenvolvimento dessa dissertação, principalmente nos momentos mais tensos, mostrando-me como incentivo a conclusão de mais uma etapa na minha vida.

A todas as pessoas que me ajudaram direta e indiretamente para a realização de meus objetivos.

Por fim, a Deus.

RESUMO

O ensino é uma das atividades mais antigas exercidas pelo homem, porém hoje ela ainda é realizada muitas vezes sem a criação de diálogos e debates entre todos os envolvidos, sendo os alunos muitas vezes agentes passivos, não havendo interatividade com os professores e o conteúdo abordado. Este trabalho apresenta uma ferramenta utilizada para prover interatividade em ambientes educacionais por meio de celulares, desta forma professores podem fazer uso da tecnologia para auxiliar no ensino e ter uma melhor avaliação dos alunos. A arquitetura da ferramenta desenvolvida é mostrada, trazendo características de tecnologias de comunicação sem fio utilizadas e como é feito o gerenciamento de conexões através de Bluetooth, que possui um número limitado de conexões simultâneas. Os detalhes das múltiplas conexões Bluetooth e como o sistema deve se comportar mediante inúmeros usuários são exibidos, mostrando um comparativo entre diferentes métodos de gerenciamento das conexões. Por último, são mostrados os resultados obtidos com o uso da ferramenta, seguido pela análise dos mesmos e uma conclusão sobre o trabalho desenvolvido.

Palavras-chave: interatividade; ensino; múltiplas conexões; multiusuário; celulares; Bluetooth.

ABSTRACT

Education is one of the oldest activities practiced by man, but today it is still performed often without creating dialogues and discussions among all those involved, and students are passives agents without interactivity with teachers and the content approached. This work presents a tool used for providing interactivity in educational environments using cell phones, in this way, teachers can use technology to assist in process of education and have a better evaluation of students. The tool developed architecture is shown, exposing features of wireless communication technologies used and how is the connection management using Bluetooth technology, which has a limited number of simultaneous connections. The details of multiple Bluetooth connections and how the system should behave by numerous users are displayed, showing a comparison between different methods of managing connections. Finally, the results obtained with the use of the tool are presented, followed by the analysis of them and a conclusion on the work.

Keywords: interactivity; education; multiple connections; mobiles; multiuser; Bluetooth.

SUMÁRIO

1 INTRODUÇÃO **11**

1.1 MOTIVAÇÃO 15

1.2 CONTRIBUIÇÃO 18

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO 19

2 TRABALHOS RELACIONADOS **21**

2.1 H-ITT 21

2.2 UBIQUITOUS PRESENTER 23

2.3 PLS TXT UR THOUGHTS 24

2.4 COMPARATIVO ENTRE O MOBILE INTERACT E OUTROS CRS 25

2.5 OUTROS PROJETOS 27

3 EMBASAMENTO TEÓRICO **29**

3.1 ACTIVE LEARNING E CRS 29

3.2 TECNOLOGIAS DE COMUNICAÇÃO SEM FIO 33

3.2.1 Infravermelho 34

3.2.2 IEEE 802.11 36

3.2.3 Bluetooth 38

3.2.4 Comparativo entre as tecnologias 45

4 MOBILE INTERACT **48**

4.1 USO DO SISTEMA 48

4.2 ARQUITETURA DO SISTEMA 50

4.2.1 Arquitetura baseada em filas 52

4.2.2 Arquitetura utilizando scatternets 53

4.2.3 Arquitetura de múltiplos servidores interligados 54

4.3 USO DO SISTEMA NOS CELULARES 56

5 IMPLEMENTAÇÃO DO SISTEMA **59**

5.1 AMBIENTE DE DESENVOLVIMENTO 59
5.2 ESPECIFICAÇÃO E IMPLEMENTAÇÃO 62
5.2.1 Módulo Bluetooth Marketing 62
5.2.2 Módulo Servidor 65
5.2.3 Módulo Celular 69
6 RESULTADOS 72
6.1 TESTES COM O BLUETOOTH MARKETING 73
6.2 TESTES COM O SERVIDOR WI-FI 74
6.3 TESTES COM O SERVIDOR BLUETOOTH 76
6.4 ANÁLISE GERAL 83
7 CONCLUSÃO 85
7.1 TRABALHOS FUTUROS 87
REFERÊNCIAS 89
ANEXO I – DIAGRAMA DE CLASSES DO MOBILE INTERACT 92

LISTA DE FIGURAS

Figura 1 - Receptor e Transmissor do H-ITT 22
Figura 2 - Caminho da resposta de um aluno do celular ao repositório de dados 25
Figura 3 - Clicker e receptor USB 32
Figura 4 - Ambiente com um servidor e apenas um usuário 42
Figura 5 - Ambiente com um servidor e até sete usuários 42
Figura 6 - Ambiente com um servidor e vários dispositivos 43
Figura 7 - Ambiente com vários dispositivos e apenas um servidor utilizando filas 45
Figura 8 - Diagrama de Casos de Uso 49
Figura 9 - Arquitetura baseada em filas 53
Figura 10 - Arquitetura utilizando scatternets 54
Figura 11 - Arquitetura de múltiplos servidores 55
Figura 12 - Arquitetura da biblioteca Bluecove 60
Figura 13 - Sistema Mobile Interact 62
Figura 14 - Diagrama de atividades do Bluetooth Marketing 63
Figura 15 - Diagrama de classes do módulo Bluetooth Marketing 64
Figura 16 - Diagrama de atividades do Servidor 66
Figura 17 - Diagrama de classes do módulo Servidor 67
Figura 18 - Diagrama de atividades do módulo Celular 69
Figura 19 - Diagrama de classes do módulo celular 70
Figura 20 - Ambiente de teste 81

GLOSSÁRIO

API - Application Programming Interface

ARS - Audience Response System

CDC - Connected Device Configuration

CLDC - Connected Limited Device Configuration

CRS - Classroom Response System

CSS - Cascading Style Sheets

HDTV – High Definition Television (Televisão de Alta Definição)

HTML - HyperText Markup Language

IrDA – InfraRed Data Association

JEE - Java Enterprise Edition

JME - Java Micro Edition

JSE - Java Standard Edition

JVM – Java Virtual Machine

MMS - Multimedia Messaging Service

SBTVD - Sistema Brasileiro de Televisão Digital

SMS - Short Message Service

STB - Set-Top-Box

TVD - Televisão Digital

TVDI – Televisão Digital Interativa

WAP - Wireless Application Protocol

WLAN - Wireless Local Area Network

WMAN – Wireless Metropolitan Area Network

WPAN - Wireless Personal Area Network

WTK - Wireless Toolkit

WWAN - Wireless Wide Area Network

XML - Extensible Markup Language

1 INTRODUÇÃO

Com a constante evolução tecnológica, a cada dia surgem novos meios de comunicação e interação entre as pessoas. Recursos multimídia trazem cada vez mais entretenimento e também ajudam o homem a absorver mais informações apresentadas no dia a dia de forma natural e interativa.

Por outro lado, o homem vem exercendo importantes tarefas desde os primórdios da civilização que não evoluíram tão rapidamente. Esse é o caso de atividades como o ensino, na qual professores devem adaptar seus métodos para educar os alunos, fato este em mudança graças à inserção da tecnologia nas salas de aula.

Com o uso da tecnologia em ambientes de ensino, a forma como o conteúdo é apresentado aos alunos em sala de aula é melhorado cada vez mais. O uso de projetores, apresentações multimídia, etc., facilita a forma como os professores passam novos assuntos para os alunos, melhorando a interatividade e a absorção do conteúdo pela turma.

Várias pesquisas em áreas pedagógicas foram feitas com o intuito de aperfeiçoar métodos de ensino (SILVA, 2000). Estudos com metodologias como o *Active Learning* mostram que os estudantes absorvem melhor o conteúdo apresentado quando participam ativamente do processo de ensino-aprendizado. O *Active Learning* é um processo no qual os estudantes são envolvidos ativamente através de perguntas que são feitas como estimulo a debates (CHEN, 2005).

Esse método é baseado na metodologia Sócrates de ensino, que prioriza a interatividade através de debates entre os alunos em uma sala de aula (BOYLE, 2002). Desta forma, a interatividade é considerada um importante componente no ensino e aprendizado, sendo uma das bases para o sucesso em classes tradicionais. Estudos mostram que os estudantes tentem a desenvolver melhor os pensamentos e conclusões sobre assuntos específicos quando tentam expô-los a outras pessoas, tentando convencê-las de suas ideias. Quando o debate é feito com

um professor, o aluno tende a absorver ainda mais suas ideias e pensamentos, levando a um rápido aprendizado (SIAU, et al., 2006).

Estudos também mostram que 33% de alunos em universidades brasileiras são tímidos, prejudicando o rendimento em salas de aulas, uma vez que alunos tímidos participam e interagem menos com os professores e colegas (SILVA, 2000). O problema é agravado em salas com um grande número de estudantes, pois quanto maior o número de pessoas presentes, mais intenso é o medo de falar em público e expor suas ideias na frente de todos.

Em turmas pequenas, os professores possuem maior controle sobre alunos e o conteúdo apresentado em sala de aula. Os alunos também costumam se sentir mais próximos do professor e os problemas de timidez não são tão evidentes. Desta forma, é mais fácil gerar uma maior interatividade entre os integrantes da turma, criando um debate mais ativo e estimulando os participantes a desenvolverem pensamentos sobre os assuntos abordados (SILVA, 2000).

É importante para o professor ter um *feedback* dos alunos, observando os principais problemas de uma turma diante um conteúdo específico. A maneira mais comum para obter esses resultados é através de provas aplicadas aos alunos, contudo, geralmente elas não trazem respostas tão instantâneas e o professor só vai notar a dificuldade da turma depois de já ter encerrado o assunto. Com resultados rápidos, os professores podem expressar melhor o conteúdo de maior dificuldade dos alunos, melhorando o desempenho da turma antes de iniciar um novo assunto (BAQUER, et al., 2009).

A avaliação dos alunos em salas de aula grandes também é um problema uma vez que a proximidade na relação entre professores e alunos é inversamente proporcional ao número de alunos, ou seja, em salas de aula grandes os professores estão mais afastados dos alunos e não conseguem identificar as necessidades de cada um. Em salas de aula pequenas é comum que os professores saibam mais sobre a personalidade de cada aluno e suas principais dificuldades, podendo direcionar assuntos e conteúdos a cada um deles, melhorando seus rendimentos (BAQUER, et al., 2009).

Ter salas de aulas pequenas não é um caso viável em todas as instituições de ensino. Muitas delas possuem salas com dezenas ou centenas de alunos, tornando o ensino cada vez mais impessoal. Em salas de aula grandes é difícil gerar uma interatividade entre todos os alunos: é praticamente impossível fazer com que todos os alunos falem e sejam ouvidos, principalmente quando entram questões como a timidez e a insegurança. Baseado nesta dificuldade, a tecnologia pode ser utilizada a fim de aprimorar o ensino e proporcionar uma maior interatividade em salas de aula (LOWRY, et al., 2006)

Um dos grandes avanços tecnológicos do século XX foi a criação e popularização de alguns meios de comunicação como o rádio, a televisão, internet e telefonia celular. Hoje é comum utilizarmos todo esse aparato tecnológico para obter informação sobre o mundo a nossa volta. A tecnologia nos proporciona cada vez mais mobilidade, permitindo-nos acessar informações em qualquer lugar. Os celulares vêm ganhando um grande destaque, pois nos auxilia em várias tarefas do nosso cotidiano, facilitando bastante nossos afazeres.

Voltando ao ambiente educacional, sistemas de avaliação em salas de aulas (CRS - *Classroom Response Systems*) foram desenvolvidos baseados em sistemas utilizados para medir notas em programas de audiência (ARS - *Audience Response Systems*), trazendo a possibilidade da realização de testes instantâneos a professores em suas classes (CHEN, 2005).

O uso desses sistemas CRS possibilita uma maior participação de todos os alunos de maneira individual, quebrando a barreira da timidez. O professor irá exibir uma pergunta de múltipla escolha e os alunos deverão respondê-la através de aparelhos individuais. Imediatamente o professor irá obter um *feedback*, podendo determinar o próximo assunto a ser mostrado aos alunos (CHEN, 2005).

A única desvantagem na utilização de um CRS é devida ao custo do sistema. Cada aluno deve portar um *clicker* (aparelho para interagir respondendo as questões), e as escolas teriam que arcar com todos os custos. Alguns sistemas desenvolvidos utilizam computadores para interação, mas desta forma o custo da

implantação do sistema se torna ainda mais caro. Neste trabalho é proposto um CRS que utiliza celulares, que são aparelhos pessoais e de alta disponibilidade. As escolas não precisarão adquirir equipamentos específicos, e cada aluno terá seu próprio dispositivo interativo, podendo portá-lo em qualquer sala de aula.

Certamente que o celular é o dispositivo eletrônico pessoal mais difundido na população (ANATEL, 2010), trazendo benefícios à medida que ele disponibiliza certo poder computacional, permitindo a execução de aplicativos, gerando uma ferramenta de interação e inclusão digital sempre ao alcance das pessoas. A capacidade de processamento e armazenamento dos aparelhos celulares está aumentando cada vez mais, podendo-se comparar a alguns modelos de computadores de 5 a 10 anos atrás. Desta forma o celular vem se tornando um aparelho pessoal indispensável na vida das pessoas, estando cada vez mais inseridos no nosso cotidiano. Devido ao crescente poder computacional dos celulares e a grande disponibilidade, estes foram escolhidos para criar o ambiente gerido por um CRS.

Para que a interatividade seja criada e os alunos possam utilizar seus celulares, será necessária a utilização de uma tecnologia de comunicação sem fio para a troca de informações entre os aparelhos e um servidor. Inicialmente, a principal tecnologia sem fio utilizada pelos celulares era o infravermelho, sendo substituído posteriormente pelo Bluetooth. Mais recentemente os aparelhos estão sendo equipados com Wi-Fi, melhorando ainda mais a comunicação entre eles.

O grande número de usuários com celulares fez com que as empresas fossem adotando as tecnologias de comunicação sem fio e, com o poder de comunicação e processamento dos celulares cada vez maiores, várias possibilidades foram abertas e novas formas de interação para os usuários foram criadas. Os usuários são capazes de trocar cartões de visitas, mensagens personalizadas, acessar estruturas de redes, ou executar jogos e aplicativos multiusuários, etc.

O infravermelho iniciou o processo das comunicações sem fio, porém ele possui alcance e taxa de transferência de dados bastante limitados. O Bluetooth, por utilizar ondas de rádio, desbancou algumas limitações do infravermelho, como a

necessidade dos dispositivos estarem na mesma linha de visão, e possui as mesmas vantagens, como o baixo consumo de energia e o baixo custo. O Wi-Fi é o padrão mais utilizado em terminais fixos, contudo esta começando a ser bastante utilizado por dispositivos portáteis. Cada uma dessas principais tecnologias tem suas vantagens e serão aqui abordadas.

Essa dissertação aborda este ambiente interativo criado pelo CRS, mostrando como os alunos podem interagir utilizando seus celulares, as principais vantagens do uso desse sistema, a forma como é feito o gerenciamento das conexões de todos os alunos, e detalhes da arquitetura desenvolvida. Serão abordados assuntos referentes ao funcionamento do sistema sendo exibido o método de como aplicativos serão disponibilizados e instalados nos celulares, como estes se conectarão a um servidor, a forma como múltiplas conexões podem ser criadas e gerenciadas, e por último os resultados obtidos com a conexão entre celulares e servidores junto a um comparativo entre as arquiteturas desenvolvidas para a criação do CRS.

1.1 MOTIVAÇÃO

Com o aumento no número de alunos em algumas turmas universitárias, ficou clara a necessidade do uso de uma nova metodologia de ensino que induzisse a uma maior participação de todos os alunos nas aulas. O tamanho da turma acaba alterando a qualidade do ensino: quanto maior a turma mais difícil é para o professor ter um controle sobre as necessidades de cada aluno.

Turmas pequenas induzem aos professores terem um contato mais próximo dos alunos, assim, eles podem ter uma melhor avaliação de cada aluno, vendo suas dúvidas e dificuldades. Com essa aproximação dos alunos, os professores podem direcionar conteúdos específicos visando melhorar o aprendizado de cada um deles. Pode-se observar que quanto menor o número de alunos, maior é a interação entre o professor e eles (SILVA, 2000).

Por outro lado, em turmas grandes o professor já não irá possuir um maior controle, tendo como base para um *feedback* da turma apenas os poucos alunos que costumam interagir mais, fazendo perguntar, criando debates, etc. Os próprios alunos se tornam mais fechados e individualistas, muitas vezes por medo de falar na frente dos colegas, medo de errar, etc. A timidez aumenta quando os alunos se tornam mais individualistas e pesquisas em universidades brasileiras mostram que um em cada três estudantes são tímidos ou extremamente tímidos. Assim também podemos notar que os alunos não participam ativamente das aulas comparado a salas pequenas onde os alunos não sofrem de tantos problemas de timidez (SILVA, 2000).

O processo de avaliação dos alunos também é um importante fator em atividades de ensino, direcionando o professor a solucionar dificuldades dos alunos em determinados assuntos. Quanto mais rápido um professor puder obter os resultados de sua metodologia de ensino, mais rápido ele pode decidir sobre novos conteúdos ou revisões a serem feitas. Em turmas pequenas os professores podem ter um *feedback* dos alunos mais rápido, uma vez que ele está mais próximo dos alunos, já em turmas grandes esse *feedback* real muitas vezes só pode ser obtido com a realização de provas e testes (BAQUER, et al., 2009).

Baseado nessas situações surgiu a ideia de utilizar um sistema que proporcionasse uma maior interatividade aos alunos em uma sala de aula. Este sistema deveria disponibilizar ferramentas para que um professor pudesse submeter uma pergunta e os alunos pudessem respondê-la de forma individual e instantânea, gerando um *feedback* rápido ao conteúdo apresentado pelo professor.

O sistema adotado é denominado *Classroom Response System*, que são sistemas desenvolvidos a partir de outro tipo de sistema, o *Audience Response System*, utilizado em votações rápidas através de um aparelho disponibilizado a cada participante. A desvantagem do uso de um CRS é o valor do sistema, aumentado devido à necessidade de aquisição dos módulos de interação individuais, gerando um valor proporcional ao número de alunos em sala de aula.

Para contornar esses problemas, decidiu-se desenvolver um CRS adaptado à realidade brasileira, sendo um CRS de baixo custo e fácil disponibilidade. As universidades e escolas geralmente não podem adquirir uma grande quantidade de equipamentos interativos, assim, o ideal é que cada aluno possua seu próprio equipamento.

Pesquisas foram feitas procurando adotar algum padrão que não gerasse altos custos aos usuários. A prioridade era a criação de um sistema que pudesse ser utilizado por dispositivos facilmente encontrados por todos os alunos, dispositivos estes que deveriam possuir capacidade para execução de aplicativos e alguma tecnologia de comunicação sem fio para troca de informações com um servidor.

Com o avanço tecnológico cada vez maior dos celulares e com a rápida disseminação destes aparelhos na população, em janeiro de 2010 havia uma média de 0,91 celular por pessoa no Brasil (ANATEL, 2010), além de que a maioria dos celulares possuem tecnologias de comunicação sem fio como Bluetooth e Wi-Fi, decidiu-se adotá-los como dispositivos individuais de interação.

Utilizando a linguagem de programação Java foi possível criar aplicativos que podem ser instalados em celulares ou outros dispositivos portáteis que suportem a tecnologia, como é o caso de alguns reprodutores multimídia, PDA’s, etc. Desta forma, um número ainda mais diversificado de dispositivos pode ser utilizado para interação por cada aluno.

A principal motivação para este trabalho é a criação de um sistema multiusuário de baixo custo e fácil acesso aos estudantes, capaz de gerar interatividade através de celulares com conexões Bluetooth ou Wi-Fi, proporcionando ao professor agilidade na avaliação dos alunos em sala de aula. Ainda mais, o sistema criado poderá ser adaptado para uso em ambiente de TVDI (Televisão Digital Interativa) no padrão brasileiro (SBTVD) através do *middleware* Ginga, proporcionando a interatividade multiusuário através de celulares ou outros dispositivos portáteis com um STB (Set-Top-Box).

A pesquisa aqui desenvolvida está relacionada ao projeto “MultIN - Lidando com múltiplos dispositivos de interação em Programas educativos para TV Digital Interativa”, que tem como objetivo permitir que usuários com vários dispositivos portáteis interajam com aplicações de TVDI utilizando um único televisor. Um possível exemplo da utilização do Mobile Interact junto ao projeto MultIN é a criação de programas de ensino, como o Telecurso da Rede Globo, que poderiam disponibilizar aplicativos com os quais os telespectadores poderiam interagir, e a partir de um aparelho de TVDI posicionado numa sala de aula, os alunos poderiam utilizar seus celulares para responder às perguntas e obter resultados individuais. Com um controle remoto essa interação não seria possível por ele não possui tela, gerando uma limitação na qual um aluno de cada vez teria que utilizar a tela da TV para ler e responder às questões.

1.2 CONTRIBUIÇÃO

A principal contribuição deste trabalho é a criação de um CRS (*Classroom Response System*) de baixo custo que permita que alunos utilizem seus próprios celulares para interação respondendo questões elaboradas por um professor em sala de aula. O sistema deve possuir ferramentas que permitam a interação de múltiplos usuários com um servidor de conteúdo interativo através de conexões Bluetooth ou Wi-Fi dos celulares.

Para que seja possível a utilização do sistema por vários usuários simultaneamente, deve ser feito o gerenciamento de múltiplas conexões entre os celulares utilizando Bluetooth e Wi-Fi. O maior problema neste gerenciamento é o limite máximo de conexões Bluetooth ativas, que é de no máximo sete conexões, necessitando de técnicas especiais que possibilitem um número maior de conexões estarem ativas simultaneamente.

No caso do Wi-Fi, com o uso de *sockets*, as conexões podem ser criadas normalmente sem a necessidade de um gerenciador mais elaborado, uma vez que nesses dispositivos as conexões Wi-Fi não diferem das conexões normais entre computadores, podendo assim ser utilizada a mesma arquitetura de rede. Porém o

Bluetooth possui uma arquitetura e camadas de rede diferentes, necessitando de uma ferramenta que gerencie as conexões entre os celulares.

Três arquiteturas para o gerenciamento das conexões Bluetooth foram desenvolvidas, criando diferentes topologias de redes entre os dispositivos Bluetooth. Estas arquiteturas de gerenciamento trás a possibilidade de seu uso não só no Mobile Interact, mas também em outros sistemas que necessitem permitir mais de sete conexões Bluetooth ativas.

O sistema possui alguns módulos que agregam as funcionalidades necessárias ao uso real do mesmo; são elas:

- Um módulo interativo para o servidor que permite que o professor disponibilize o conteúdo interativo, além de gerenciar as conexões ativas;
- Um aplicativo JME para ser instalado no celular com a qual o usuário pode interagir com o conteúdo disponibilizado pelo servidor;
- Um módulo de Bluetooth Marketing que distribui os aplicativos interativos para os celulares, enviando seus instaladores aos dispositivos por Bluetooth.

O servidor foi desenvolvido em Java, com o objetivo de também ser compatível com os padrões de programação do *middleware* Ginga, uma vez que futuramente ele poderá ser facilmente adaptado e disponibilizado em canais interativos televisivos. A aplicação para interação do usuário através dos celulares foi desenvolvida em JME por ser suportada por uma vasta gama de aparelhos. O envio dos aplicativos para os aparelhos é feito utilizando técnicas de Bluetooth Marketing na qual um software controlador busca por dispositivos Bluetooth visíveis na sua área de alcance e envia os arquivos (os instaladores) aos mesmos.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Este documento está organizado por capítulos, começando por uma introdução no capítulo 1. Em seguida, nos capítulos 2 e 3, é feito um embasamento teórico mostrando as características da pratica do Active Learning (explicado nos capítulos seguintes) junto aos sistemas CRS, e as tecnologias e padrões de

comunicação sem fios abordados no trabalho, dando uma ênfase na arquitetura de redes Bluetooth e suas peculiaridades. No capítulo 4 detalhes do projeto e das arquiteturas desenvolvidas são abordados, seguidos dos detalhes de implementação das mesmas no capítulo 5. O capítulo 6 mostra os resultados obtidos e traz uma análise da viabilidade das arquiteturas. No capítulo 7 é feita uma conclusão sobre o tema abordado, seguido pelos projetos futuros.

2 TRABALHOS RELACIONADOS

Outros sistemas comerciais para auxilio em salas de aulas já existem, sendo denominados CRS e derivados dos ARS (*Audience Response System*), contudo nenhum dos sistemas existentes utiliza celulares com comunicação Bluetooth para interação. Todos eles utilizam hardware específico e comunicação através de infravermelho, internet, mensagens de texto, tecnologias próprias, etc.

Alguns desses sistemas são os “Hyper-Interactive Teaching Technologies” (H-ITT), “Ubiquitous Presenter” e o “PLS TXT UR Thoughts”. O primeiro utiliza hardware próprio com comunicação através de infravermelho. Os outros dois utilizam celulares, enviando as respostas por SMS para o servidor que irá computar as informações recebidas.

Além desses sistemas, este trabalho está relacionado a outros dois projetos desenvolvidos na UFRN referentes ao uso de celulares como dispositivos de interação em aplicações de TVDI, mostrados nos subcapítulos a seguir.

2.1 H-ITT

O H-ITT é um sistema que consiste de um controlador (dispositivo de interação) com transmissor infravermelho e seis botões, um conjunto de receptores infravermelho para um computador e um componente de software para processamento das respostas dos alunos.

Os controladores permitem aos alunos responderem a questões de até cinco alternativas, sendo necessário apontar o controlador para um receptor e clicar no botão equivalente a alternativa escolhida. Eles utilizam raios infravermelhos com 10ms de duração quando algum botão é pressionado, diminuindo a probabilidade de colisão entre os sinais de diferentes dispositivos, uma vez que os raios infravermelhos podem sofrer interferência um dos outros. Desta forma é possível determinar a quantidade máxima de dispositivos presentes em uma sala: o recomendado é que exista um receptor para cada 50 controladores. O problema com

o uso de infravermelho é a distância máxima que um controlador deve estar de um receptor, não sendo aconselhado estar a mais de 3m (PETR, 2005).

Cada controlador possui um identificador que pode ser associado a um aluno em particular. Este registro pode ser feito manualmente ou automaticamente pelo sistema. Os receptores coletam os dados transmitidos pelos controladores dos estudantes e repassam-nos para o computador com o software que irá gerenciar as respostas (PETR, 2005).

Ao responder uma questão, os alunos serão alertados através de um LED no controlador, informando que a resposta foi computada. Como os controladores não possuem tela para exibir os resultados, o software receptor poderá enviar as respostas e estatísticas detalhadas com os rendimentos de cada aluno para seus emails se forem definidos antecipadamente. A Figura 1 mostra um receptor e um controlador utilizados no H-ITT (PETR, 2005).

Figura 1 - Receptor e Transmissor do H-ITT[1]

[1] Fonte: Experience with a multiple-choice audience response system in an engineering classroom (PETR, 2005)

2.2 UBIQUITOUS PRESENTER

O Ubiquitous Presenter é um sistema baseado no envio de mensagens através de celulares para um email. O sistema suporta tanto informações digitadas quanto informações escritas pelos alunos, tendo o servidor o papel de distingui-las. Segundo Lindquist (LINDQUIST, et al., 2007), o envio de informações escritas permite que os alunos submetam resoluções ou dúvidas de questões anonimamente e de forma natural para o professor, podendo escrever a mão, não havendo limitações com caracteres especiais.

Através dos celulares, os estudantes podem enviar SMS (mensagem de texto) digitando as informações relevantes no campo texto ou então escrever a informação em um papel, tirar uma foto dela e enviá-la por MMS (mensagem multimídia). Os alunos podem utilizar também laptops ou PDAs caso estejam disponíveis, enviando as mensagens escritas para o email da mesma forma que os celulares (LINDQUIST, et al., 2007),

As mensagens devem ser enviadas para um email fornecido pelo professor o qual é monitorado por um serviço que atua como uma ponte entre o servidor de emails e um repositório de dados. Quando as informações são recebidas no email, o serviço irá organizá-las verificando se as mesmas são imagens ou texto editável. Caso seja texto, o sistema irá processar as informações, organizando as respostas de acordo com as questões específicas. Caso seja uma imagem, esta será exibida ao professor, permitindo que ele responda ou debata sobre o conteúdo da imagem (LINDQUIST, et al., 2007), Um problema com essa arquitetura é que não é possível contar estatísticas através das imagens, apenas do texto enviado, tornando o sistema apenas uma forma para os alunos enviarem suas dúvidas anonimamente.

Quando uma imagem é submetida, o sistema faz uma edição nela a fim de melhorar sua visualização. Muitas imagens obtidas com câmeras de celulares possuem baixa resolução, cores e contrastes ruins, etc., desta forma, o sistema tenta aplicar filtros para melhorar a exibição dessas imagens (LINDQUIST, et al., 2007).

A desvantagem dos celulares para o envio de mensagens SMS ou MMS é devido ao custo das mesmas. Se um professor exibe muitas questões e exige que os alunos participem ativamente de todos os questionários e debates através do envio de mensagens, esta atividade pode gerar altos custos aos alunos. O problema pode ser solucionado com o uso de computadores ou dispositivos com acesso à internet gratuito, para que os alunos possam submeter suas respostas para o email sem que gastos adicionais sejam gerados.

2.3 PLS TXT UR THOUGHTS

Como o Ubiquitous Presenter, o PLS TXT UR Thoughts também é um sistema que utiliza celulares para interação através do envio de mensagens para um servidor. A diferença entre eles é que o segundo aceita apenas mensagens SMS, interpretando apenas o conteúdo válido digitado na mensagem (MARKETT, et al., 2006).

Para esse sistema não é necessária a instalação de uma infraestrutura de rede nas salas de aula. É necessário ter apenas um computador com um Phone Card capaz de receber e interpretar mensagens SMS. O Phone Card habilita o computador com um número telefônico e interpreta as mensagens recebidas, direcionando-as a um serviço específico para o processamento do texto (PRINCE, 2007).

O professor tem liberdade para fazer as perguntas da maneira mais conveniente, e os alunos podem respondê-las através de seus celulares. As perguntas devem ser de múltipla escolha, levando os alunos a responderem digitando apenas uma alternativa no texto da mensagem, uma vez que os SMS são limitados a 160 caracteres (PRINCE, 2007).

Ao receber uma mensagem de texto, o Phone Card localizado no servidor irá repassar o texto recebido junto do número telefônico que enviou (servirá como identificador) para um serviço desenvolvido em Java e Python que irá processar as informações e armazenar os dados em uma tabela. O professor pode gerar os

resultados e estatísticas a partir desta tabela, verificando o desempenho da turma. A Figura 2 mostra o caminho de uma mensagem enviada de um celular para o sistema (MARKETT, et al., 2006).

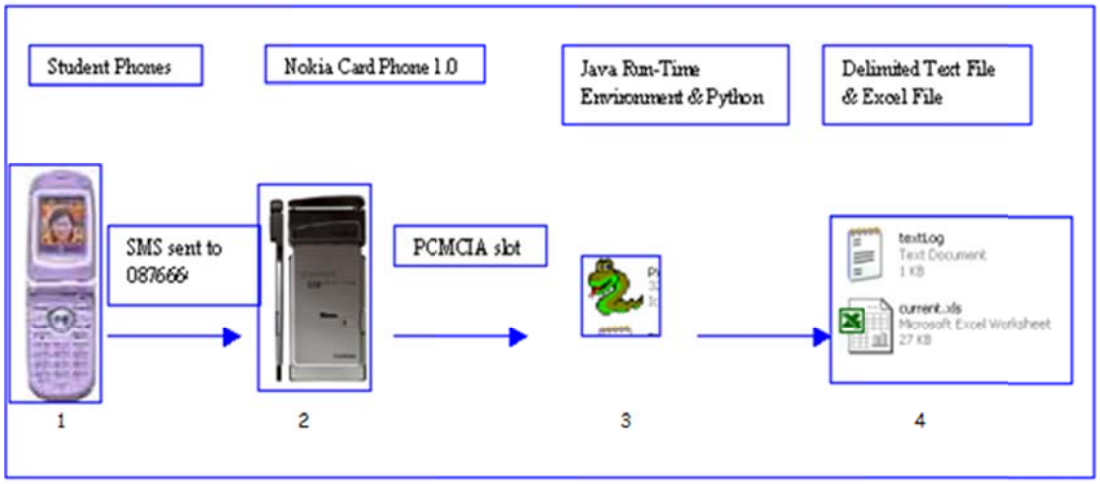


Figura 2 - Caminho da resposta de um aluno do celular ao repositório de dados[2]

Como no sistema Ubiquitous Presenter, as mensagens enviadas pelos celulares geram custos aos alunos, sendo uma desvantagem do uso do sistema.

2.4 COMPARATIVO ENTRE O MOBILE INTERACT E OUTROS CRS

Cada um dos sistemas mostrados anteriormente tem características específicas, com pontos positivos e negativos. A Tabela 1 mostra um comparativo entre esses CRS e o Mobile Interact.

Em relação ao H-ITT, o Mobile Interact não utiliza hardware próprio, sendo necessários apenas computadores com Bluetooth e/ou Wi-Fi para o servidor, e celulares como dispositivos de interação. Por possuir hardware próprio, o H-ITT proporciona aos alunos um uso mais intuitivo, com uma rápida resposta no cálculo dos resultados. Apesar do Mobile Interact possuir uma interface intuitiva, há a necessidade da instalação de aplicativos nos celulares, o que gera algumas dúvidas aos usuários.

[2] Fonte: Using short message service to encourage interactivity in the classroom (MARKETT, et al., 2006).

	Forma de interação	Vantagens	Desvantagens
H-ITT	-Hardware próprio que utiliza infravermelho	-Facilidade de uso -Velocidade nos resultados	-Hardware caro -Necessidade de vários receptores (curto alcance) -Não possui tela para exibição das questões
Ubiquitous Presenter	-Celular (imagem da câmera ou mensagem de texto) -Computadores e PDAs (email)	-Liberdade nas respostas (imagens escritas à mão) -Vários dispositivos compatíveis -Usuários não precisam adquirir hardware específico	-Custo extra para envio de mensagens por celulares -Análise das respostas não é automática se imagens forem utilizadas
PLS TXT UR Thoughts	-Celular (mensagem de texto)	-Usuários não precisam adquirir hardware específico	-Custo extra para envio de mensagens por celulares
Mobile Interact	-Celular (Bluetooth ou Wi-Fi)	-Usuários não precisam adquirir hardware específico -Sem custos para uso do sistema	-Limitação do número de conexões Bluetooth

Tabela 1 - Comparativo entre os CRS

As desvantagens do H-ITT estão relacionadas ao hardware, pois é necessária a aquisição de vários dispositivos próprios para interação, além de alguns receptores infravermelho. Como os receptores possuem um pequeno alcance, eles devem ser espalhados pela sala de aula, cobrindo toda a área da mesma. O hardware gera um custo maior na implantação do sistema, ao contrário do Mobile Interact que não necessita a aquisição de hardware próprio, tendo cada aluno seu próprio dispositivo de interação (o celular). Uma outra desvantagem do H-ITT é que os dispositivos de interação não possuem tela própria, assim os alunos não podem ler as questões em seus dispositivos, nem ver seus resultados individualmente.

Assim como o Mobile Interact, o PLS TXT UR Thoughts e o PLS TXT UR Thoughts utilizam celulares como dispositivos de interação, não necessitando de aquisição de hardware específico de interação para os alunos. No PLS TXT UR Thoughts o aluno utiliza seu celular apenas para enviar uma resposta às questões por mensagem de texto para o servidor. O servidor analisa as mensagens recebidas e computa os resultados, informando ao professor. A desvantagem é que os alunos podem digitar texto incompreensível pelo servidor, anulando suas respostas.

No PLS TXT UR Thoughts os alunos são capazes de enviar suas questões por mensagens de texto ou tirar uma foto da solução feita à mão e enviá-la por mensagem multimídia. Os alunos podem ainda utilizar computadores e PDAs, enviando suas respostas por email para o servidor. Mais uma vez o problema está no cálculo dos resultados, já que o servidor não é capaz de interpretar todo conteúdo enviado pelos alunos. Se imagens forem enviadas ao servidor, ele irá melhorar algumas propriedades da imagem e a exibirá ao professor para que ele interprete-a. O sistema não gera o rápido feedback ao professor, necessitando que ele interprete alguns resultados enviados pelos alunos, não gerando estatísticas automáticas.

O uso do PLS TXT UR Thoughts e do PLS TXT UR Thoughts implica em gastos extras aos alunos com o envio de mensagens para o servidor através dos celulares. O Mobile Interact propõe o uso de conexões locais através de Bluetooth ou Wi-Fi entre os celulares e o servidor, não gerando gastos para os usuários do sistema. A única desvantagem está relacionada ao uso do Bluetooth que possui a limitação de sete usuários conectados simultaneamente. Este é o problema solucionado nesta dissertação através do uso de diferentes arquiteturas para o gerenciamento das conexões.

2.5 OUTROS PROJETOS

Uma parte deste trabalho está relacionada ao trabalho de conclusão de curso, “Utilizando Dispositivos Móveis e Bluetooth para Aplicação de Avaliações em Meio Digital” (FERREIRA, 2009). No trabalho, Ferreira desenvolveu uma aplicação para

ser distribuída por programas televisivos, utilizando o middleware Ginga, na qual permite que um Set-Top-Box (STB) possa se comunicar com um celular através de Bluetooth.

No trabalho, a ferramenta desenvolvida permite que um STB possa ter até sete celulares conectados por Bluetooth, sendo sete o número máximo de conexões possíveis para um único dispositivo. O trabalho não gerencia a forma como as conexões são realizadas nem possibilita a conexão de um número grande de usuários (FERREIRA, 2009).

Um exemplo de aplicação utilizado no trabalho foi a criação de questionários para serem exibidos pelo programa televisivo interativo e respondidos pelos telespectadores utilizando seus celulares. As questões respondidas eram armazenadas nos celulares e eram corrigidas quando submetidas ao servidor (FERREIRA, 2009).

O trabalho desenvolvido está ligado ao projeto de pesquisa MultIn, desenvolvido na UFRN, que tem como objetivo possibilitar novas formas de interação com a TVDI, permitindo que alunos possam ter aulas à distancia por meio de programas televisivos.

Esta dissertação se relaciona com o projeto MultIn criando um CRS que utiliza celulares como dispositivos de interação. O trabalho desenvolvido por Ferreira foi utilizado como base para criar as conexões Bluetooth entre os celulares e servidores, sendo empregado o exemplo dos questionários desenvolvido por ele. A arquitetura para gerenciamento das múltiplas conexões Bluetooth é desenvolvida nesta dissertação, podendo ser utilizada também no projeto MultIn para possibilitar a interação de vários usuários utilizando seus celulares no âmbito da TVDI.

3 EMBASAMENTO TEÓRICO

Este capítulo tem como objetivo dar um embasamento teórico na metodologia de ensino Active Learning, mostrando suas características e benefícios; os sistemas utilizados em sala de aula para prover interatividade, os CRS; e as principais tecnologias de comunicação sem fio utilizadas por dispositivos portáteis, como o infravermelho, Bluetooth e Wi-Fi.

3.1 ACTIVE LEARNING E CRS

A maior parte da tecnologia utilizada em sala de aula é utilizada para oferecer suporte à metodologia pedagógica chamada *Active Learning*. O objetivo desta metodologia é submergir os alunos no processo de ensino, utilizando muitas ideias de como interagir diretamente com os estudantes.

A metodologia *Active Learning* pode ser comparada às metodologias utilizadas pelos filósofos gregos na antiguidade. Eles acreditavam que a melhor forma de ensinar era interagindo diretamente com os estudantes, fazendo-os perguntas individuais e levando-os a pensar sobre um assunto específico, induzindo-os a chegar a suas próprias conclusões. Esta metodologia utilizada pelos gregos pode ser considerada a primeira forma do *Active Learning* (ABRAHAMSON, 1999).

A metodologia não trata apenas da utilização de material adequado junto aos alunos e sim à interação direta com os alunos, ajudando-os a desenvolver seus próprios pensamentos e argumentos. Esta teoria de ensino é chamada de Construtivismo. Uma teoria alternativa, o Behaviorismo, acredita que os estudantes aprendem apenas estando diante de informações novas, como se absorvessem o conteúdo incondicionalmente. Exemplos desta teoria seriam palestras nas quais os participantes são todos ouvintes passivos (PRINCE, 2007)

Como o próprio nome diz, *Active Learning* é um processo de aprendizado ativo no qual o aluno deve assimilar o conteúdo dado por um professor, montando

novas ideias e modelos mentais a partir de assuntos conhecidos previamente adicionados de novas ideias trazidas pelo professor. Esses novos pensamentos não podem ser criados por conta própria devendo ser direcionados pelos professores, fazendo com que os alunos construam suas ideias seguindo um pensamento correto. Para que isto seja possível, o aluno deve participar ativamente das atividades criadas pelo educador, até mesmo para que este possa analisar e direcionar o aluno ao pensamento correto (PRINCE, 2007).

Em algumas pesquisas realizadas com a metodologia *Active Learning*, foi observado que ela era bastante eficaz, aumentando consideravelmente os resultados obtidos com avaliações aplicadas aos alunos (ABRAHAMSON, 1999). Uma outra pesquisa realizada com mais de 6000 alunos mostra que em classes nas quais o professor utilizava a metodologia *Active Learning*, os alunos obtiveram resultados melhores, havendo um ganho percentual de duas vezes o desvio padrão encontrado em relação aos alunos de turmas que tiveram aulas baseadas em palestras (HAKE, 1998).

Um exemplo prático de utilização da metodologia seria definido em três fases: primeiro o professor passa o conteúdo aos alunos; segundo, exercícios individuais são apresentados; terceiro, os estudantes resolvem os exercícios e tentam realizar discussões sobre o assunto com outros estudantes ou com o professor. Os exercícios individuais são fundamentais para a construção de ideias dos alunos; eles constituem o primeiro passo para a criação de novos conceitos, ajudando aos alunos elaborarem seus próprios pontos de vista. Estes exercícios são a base para a utilização de um *Classroom Response System*, foco de discussão nesta dissertação.

Um *Classroom Response System* é um sistema que permite aos educadores apresentar uma questão aos alunos e estes responderem através de algum dispositivo. Assim, rapidamente o educador obterá as respostas dos alunos. As salas de aula devem ser equipadas com dispositivos para interação em cada cadeira, e no momento que o professor fizer uma pergunta, os alunos poderão respondê-la através dos dispositivos. O professor saberá quantos alunos responderam (quantos dispositivos foram pressionados) e a contagem dos acertos e erros (BAQUER, et al., 2009).

Um CRS é um sistema derivado de um ARS (*Audience Response System*), que é um sistema utilizado em votações rápidas nas quais todos os participantes portam um dispositivo de votação que possui teclas referentes às opções de voto. Este sistema pode ser facilmente encontrado em programas de auditório. A vantagem do sistema é a rapidez na votação, trazendo o resultado logo após o último foto ser finalizado. Um Classroom Response System possui uma estrutura base semelhante, porém voltada para ensino em salas de aula.

Um CRS consiste basicamente de alguns componentes: dispositivos individuais para resposta às perguntas elaboradas por um professor (*clickers*), um dispositivo de coleta dos dados emitidos pelos *clickers* (servidor), um software para que o professor possa redigir e disponibilizar as questões e obter os resultados das respostas dos alunos (BAQUER, et al., 2009).

Como servidor, dependendo dos requisitos do sistema utilizado, sejam eles de hardware ou software, o professor pode utilizar um computador próprio, como um notebook, ou um computador previamente instalado em cada sala de aula. Os servidores devem possuir um receptor que possa fornecer comunicação com os *clickers,* sendo exemplos dispositivos USB com protocolos de comunicação própria, pontos de acesso de redes sem fio, dispositivo infravermelho, e Bluetooth, que é o caso deste trabalho (KALETA, et al., 2007).

Cada um desses computadores deve portar o software responsável pelo envio das perguntas e recebimento das respostas, fornecendo o *feedback* exigido pelo professor. O principal objetivo do software do servidor é coletar os dados dos alunos e gerar estatísticas através de suas respostas. Os dados coletados são então organizados e exibidos ao professor. Além disso, caso seja possível exibir os resultados através dos *clickers*, o servidor pode enviar as respostas corrigidas aos alunos, que também poderão analisar seus resultados (BAQUER, et al., 2009).

As salas de aula devem ser equipadas com os dispositivos de interação mais conhecidos como *clickers*, estando disponíveis para todos os alunos. Estes dispositivos variam muito de acordo com o CRS utilizado. Alguns usam hardware

próprio e outros utilizam equipamentos como laptops, PDAs, celulares, etc. Os *clickers* possuem um teclado de entrada para que os alunos possam responder às questões através da escolha de alguma alternativa. Muitos *clickers* possuem teclado com opções de 0 a 9 ou letras de A a E para indicar as alternativas de resposta (KALETA, et al., 2007).

A maioria dos dispositivos, com exceção dos computadores, PDAs, etc., não possui tela própria, assim os professores devem fazer as questões de maneira clara, utilizando projetores ou outros métodos convenientes, de forma que todos os alunos possam acompanhá-las (KALETA, et al., 2007). A Figura 3 ilustra um dispositivo de interação (*clicker*) com um receptor USB.

Figura 3 - Clicker e receptor USB[3]

Resultados obtidos com o uso de um CRS em uma sala de aula mostraram que o sistema envolve os estudantes com mais conteúdo e conceitos durante as aulas, melhorando os seus desempenhos. O uso de um CRS modifica a dinâmica das aulas, fazendo uso auxiliar da tecnologia, resultando em alunos mais atentos e ativos (KALETA, et al., 2007).

[3] Fonte: Student Response System: A University of Wisconsin System Study of Clickers (KALETA, et al., 2007)

Registros indicam que o primeiro CRS desenvolvido e comercializado foi o “Classtalk”, com vendas iniciadas em 1992. O sistema foi desenvolvido por um engenheiro da NASA com a colaboração de outras entidades de ensino e pesquisa, utilizando dispositivos semelhantes a calculadoras gráficas como os dispositivos de interação. O servidor utilizado pelo professor era um computador Macintosh que possuía um console apropriado e uma tecnologia de rede desenvolvida especialmente para o sistema. O sistema não obteve muito sucesso por ser um sistema caro: geralmente um dispositivo de interação era compartilhado por quatro estudantes, além de necessitar uma instalação especial da rede em cada sala de aula (BEATTY, 2004).

No final da década de 90, o “Classtalk” saiu do mercado sendo substituído por outros sistemas mais simples e baratos, como o “EduCue PRS” e o “eInstruction CPS”. Eles usavam dispositivos denominados *clickers* para interação, utilizando infravermelho como meio de comunicação com o servidor, sendo comparados a grandes controles remotos. (BEATTY, 2004).

Mais recentemente novos CRS que utilizam outras tecnologias e dispositivos começaram a ser lançados comercialmente. Eles utilizam laptops, tablet PCs e PDAs para prover a interação, devendo cada estudante ter o seu próprio dispositivo. Muitos utilizam redes Ethernet, Wi-Fi e tecnologias próprias para comunicação com o servidor (BEATTY, 2004). Outros utilizam celulares como dispositivos de entrada, porém todos utilizavam o envio de mensagens para o servidor para que pudessem submeter suas respostas. Nenhum deles utilizava uma tecnologia de rede sem fio gratuita para o usuário.

3.2 TECNOLOGIAS DE COMUNICAÇÃO SEM FIO

Uma rede sem fio é uma rede de computadores sem a necessidade do uso de cabos, feita por meio de equipamentos que usam, por exemplo, radiofrequência (comunicação via ondas de rádio) ou comunicação via Infravermelho. Seu uso já é bastante comum em redes de computadores, servindo como meio de acesso à

Internet através de locais remotos como um escritório, um restaurante, um parque, etc (INFOWESTER, 2009).

Tecnologias como o Bluetooth foram desenvolvidas de maneira que possibilitasse a conexão e a troca de informações entre dispositivos portáteis como celulares, computadores, impressoras, câmeras digitais, etc., através de uma frequência de rádio de curto alcance não licenciada e segura, abrindo inúmeras possibilidades de comunicação (INFOWESTER, 2009).

A classificação das redes é baseada na área de abrangência, sendo elas redes pessoais ou de curta distância (WPAN), redes locais (WLAN), redes metropolitanas (WMAN) ou redes geograficamente distribuídas ou de longa distância (WWAN). Cada tecnologia possui qualidades específicas que determinam qual o melhor ambiente para seu uso, como o Bluetooth e o Infravermelho que possuem um baixo consumo de energia, sendo ideal para uso em dispositivos portáteis com restrições de energia (SIG, 2009).

Neste trabalho apenas as redes locais e de curta distância serão abordadas, sendo mostradas características do Bluetooth, do Wi-Fi e do infravermelho, que são as principais tecnologias usadas, principalmente por dispositivos portáteis. Um comparativo entre elas será mostrado junto da justificativa pela escolha de cada tecnologia a ser utilizada no projeto.

3.2.1 Infravermelho

O infravermelho (IrDA - Infrared Data Association) é uma tecnologia já utilizada em vários dispositivos, como controles remotos, há anos. Em 1993, grandes empresas da área de comunicações se juntaram para elaborar uma especificação que possibilitaria a troca de dados por comunicação ponto-a-ponto ou ponto-a-multiponto entre dispositivos com infravermelho, visando a comunicação entre dispositivos móveis e portáteis, como notebooks, PDAs, celulares, etc (PAPYRUS, 2009).

A comunicação é feita através da emissão de luz por LEDs infravermelhos em uma frequência específica, sendo captadas por outro dispositivo infravermelho. Normalmente todos os dispositivos possuem um transmissor e um receptor. A distância para comunicação entre os dispositivos deve ser de 10 cm a 1 metro com um ângulo de 30 graus entre eles, embora este intervalo possa ser aumentado dependendo da potência do hardware do dispositivo. As taxas de transmissão variam de acordo com a especificação do dispositivo num intervalo de 9,6 kbps a 4 Mbps. Alguns dispositivos com velocidades de 1 Gbps estão sendo desenvolvidos (INFRARED DATA ASSOCIATION, 2009).

As especificações IrDA incluem normas tanto para os dispositivos físicos quanto para os protocolos de comunicação. Existem dois grupos de protocolos: o *IrDA Data Standard*, que é composto por protocolos padrões e alguns opcionais, e o *IrDA Control Standards*, que é utilizado por periféricos como teclados, mouses, joysticks, entre outros dispositivos que provêem interatividade. Os protocolos incluídos nesses dois grupos são: IrPHY, IrLAP, IrLMP, IrCOMM, Tiny TP, IrOBEX, IrLAN e IrSimple; porém eles não serão abordados em detalhes neste trabalho (WIKIPEDIA, 2009).

As normas do IrDA não especificam padrões de segurança uma vez que os dois dispositivos precisam estar dentro da mesma linha de visão para que seja estabelecida uma conexão. Desenvolvedores devem implementar os próprios recursos de segurança, como autenticação, encriptação, etc. (EVERS, et al., 2003)

A tecnologia foi muito popular entre os notebooks e outros dispositivos portáteis ate o início dos anos 2000, que foi quando novas tecnologias concorrentes popularizaram-se, como o Bluetooth e Wi-Fi. Estas novas tecnologias não necessitam estar dentro de uma linha de visão para criar uma conexão, possuindo uma maior área de cobertura, além de maiores taxas de transmissão de dados. Apesar de não ter protocolos de segurança padronizados, a tecnologia é de certa forma segura e não possui os mesmos problemas de segurança que as tecnologias baseadas em transmissão por ondas de rádio (PAPYRUS, 2009).

3.2.2 IEEE 802.11

Mais conhecido como Wi-Fi, o padrão IEEE 802.11 descreve uma tecnologia para comunicação criando redes sem fios e garantindo a interoperabilidade entre diferentes dispositivos. O padrão opera em faixas de frequências que não necessitam de licença para instalação e/ou operação, 5 GHz e 2,4 GHz (WI-FI ALLIANCE, 2010)

As redes Wi-Fi estão cada vez mais comuns e em muitos lugares é possível utilizar serviços de internet que podem ser acessados por meio de um ponto Wi-Fi. Utilizando um notebook, PDAs, ou mesmo alguns celulares compatíveis, é possível acessar um Hot spot (ponto de acesso Wi-Fi) e acessar uma rede interna, internet ou outros recursos (CUNHA, et al., 2005).

Atualmente, praticamente todos os computadores portáteis vêm de fábrica com dispositivos de rede sem fio no padrão Wi-Fi (802.11 a, b, g ou n). O que antes era acessório está se tornando item obrigatório, principalmente devido ao fato da redução do custo de fabricação. Muitas operadoras de telefonia, provedores de internet e empresas de comunicação estão investindo bastante em redes Wi-Fi trazendo novas soluções para clientes empresariais e residenciais (CUNHA, et al., 2005).

Roteadores Wi-Fi também podem criar pontes entre redes sem fio e redes com fio, facilitando a conexão entre duas redes cabeadas distintas as quais não seria possível ligá-las através de cabos, como duas redes internas em duas casas, por exemplo. Uma rede Wi-Fi também provê acesso à rede em lugares pouco usuais além de permitir mobilidade por toda sua área de cobertura.

Algumas empresas e instituições, como a UFRN, oferecem acesso à rede interna e à internet através de uma rede Wi-Fi dentro de suas dependências. Neste caso, alguns extensores de alcance ou repetidores são utilizados para aumentar a área de alcance da rede. Também já há algumas cidades que oferecem internet gratuita para seus habitantes, porém uma tecnologia mais recente é mais adaptada a esta tarefa, a WIMAX. Antenas difusoras e antenas receptoras direcionadas

também podem ser utilizadas para propagar os sinais transmitidos por centenas de metros (CUNHA, et al., 2005).

A área de abrangência de um ponto Wi-Fi geralmente está entre 30m e 100m em média, dependendo dos hardwares utilizados. Alguns fornecedores aumentam essa área de abrangência de modo que, com a utilização de hardwares do mesmo fabricante, é possível obter um melhor desempenho. As antenas também podem aumentar essa área em quilômetros, porém precisam ser direcionadas, perdendo a mobilidade (MURILO, 2006).

São definidos vários padrões na família IEEE 802.11, sendo os principais os padrões 802.11 A, B, G e N. O IEEE 802.11a opera na frequência de 5 GHz com capacidade teórica de transmissão de 54 Mbps. O IEEE 802.11b opera na frequência de 2,4 GHz com capacidade teórica de transmissão de 11 Mbps, este padrão utiliza *Direct Sequency Spread Spectrum* (Sequência Direta de Espalhamento de Espectro) para diminuição de interferência. O IEEE 802.11g opera na frequência de 2,4 GHz com capacidade de 54 Mbps. O IEEE 802.11n pode trabalhar com as faixas de 2,4 GHz e 5 GHz, o que o torna compatível com os padrões anteriores, e possui capacidade teórica de transmissão de 300 Mbps podendo atingir até 600 Mpbs (WI-FI ALLIANCE, 2010).

O padrão 802.11g ainda é o mais utilizado para redes Wi-Fi, mas já é possível encontrar vários equipamentos à venda que trabalham também com o padrão 802.11n, cujo desenvolvimento se iniciou em 2004 e ainda não foi completamente terminado. Em alguns casos a área de cobertura do padrão 802.11n pode passar dos 400 metros (WI-FI ALLIANCE, 2010).

O 802.11 provê os protocolos WPA, WPA2 e WEP como padrões de segurança utilizadas em sua especificação. O WEP possui falhas de segurança, podendo ser quebrado após pouco tempo de escuta dos dados transmitidos, sendo substituído pelos mais recentes WPA e WPA2 (MURILO, 2006).

3.2.3 Bluetooth

Bluetooth é uma especificação de protocolos para comunicação sem fio projetado para comunicação e transferência de uma baixa taxa de dados entre diferentes dispositivos próximos, provendo um baixo consumo de energia e dispositivos de baixo custo (JOHNSON, 2004).

A tecnologia utiliza uma frequência de rádio de curto alcance globalmente não licenciada e segura, possuindo uma área de alcance pequena, variando entre 1 a 10 metros, e posteriormente aumentada para uma área de até 100 metros, dependendo da classe do dispositivo e hardware utilizado. Alguns dispositivos mais antigos podem ter serviços limitados, como apenas uma conexão por vez, menor alcance, etc. A taxa de transferência de dados é de até 1 Mbps na versão 1.2 da especificação, 3 Mbps na versão 2.0 e de até 24 Mbps na versão 3.0 (SIG, 2009).

A tecnologia Bluetooth está bastante difundida atualmente e continua evoluindo. A versão mais atual da especificação, a 3.0, foi lançada no inicio de 2009, conseguindo atingir velocidades de transferência de aproximadamente 24 Mbps. A versão emprega praticamente o mesmo consumo de energia da versão 2.0, e também é compatível com as versões anteriores. Alguns novos protocolos e características foram adicionados na nova especificação, provendo mais serviços e possibilidades aos usuários e desenvolvedores. Contudo, os celulares e dispositivos portáteis mais comuns utilizam mais as especificações 1.0 e 2.0 do Bluetooth; deste modo, eles ainda não possuem uma grande taxa de transferência e uma grande área de alcance. (FERREIRA, 2009)

Por ter a característica principal de baixo consumo de energia, a tecnologia é ideal para o uso em dispositivos portáteis que sofrem com restrições em relação a fonte de alimentação e uso de baterias, como é o caso de celulares e PDA’s, além de computadores portáteis, impressoras, eletrodomésticos inteligentes como geladeiras, etc.

O Bluetooth surgiu a partir de estudos feitos pela Ericsson que tinha como objetivo criar uma interface de rádio que tivesse baixo consumo de energia e baixo

custo para substituir cabos em seus aparelhos de telecomunicações. Em 1998 grandes empresas formaram o *Bluetooth Special Interest Group* (SIG) para desenvolver a tecnologia sem fios, onde criaram a especificação Bluetooth 1.0 através de padrões abertos. Em 2000 foram lançados os primeiros equipamentos com Bluetooth integrado, como o primeiro celular, o primeiro adaptador para computador, etc., e também foi aprovada a especificação 802.15.1 pelo IEEE como a tecnologia wireless Bluetooth, baseado na versão 1.1 de sua especificação. Atualmente o SIG conta com aproximadamente 8000 membros e mais de um bilhão de dispositivos vendidos pelo mundo (SIG, 2009).

O protocolo possui uma arquitetura Cliente-Servidor, possibilitando a criação de redes pessoais sem fio (*Personal Area Network - PAN*). A tecnologia Bluetooth é formada de hardware de transmissão por ondas de radio na frequência 2,45 GHz, uma pilha de protocolos e perfis de interoperabilidade (INFOWESTER, 2009).

A pilha de protocolos Bluetooth provê alguns protocolos de alto nível e API's para o serviço de busca por dispositivos e entradas e saídas seriais, além de prover também protocolos de baixo nível para segmentação de pacotes, multiplexação e qualidade de serviço. (ORTIZ, 2004)

Os perfis de interoperabilidade definem os requisitos para consistência e interoperabilidade entre plataformas, incluindo perfis com diferentes objetivos. Entre os vários perfis, alguns se destacam por estarem mais relacionados a este trabalho (ORTIZ, 2004), são eles:

- GAP (*Generic Access Profile*), que define as funcionalidades de gerenciamento do dispositivo e é base para todos os outros perfis
- SDP (*Service Discovery Application Profile*), para descobrir serviços em um dispositivo remoto
- SPP (*Serial Port Profile*), que define os requisitos de interoperabilidade e capacidades para a emulação de cabo serial
- GOEP (*Generic Object Exchange Profile*), que provê a base para todos os perfis de troca de dados, sendo baseado no OBEX

- FTP (*File Transfer Profile*), que provê acesso ao sistema de arquivos do outro dispositivo

Cada dispositivo possui um identificador único de 48bits, sendo denominado *Bluetooth Address.* Os endereços geralmente são mascarados por um nome atribuído ao dispositivo que pode ser configurado pelo usuário. Com esse identificador, uma lista de dispositivos “pareados” pode ser criada em cada dispositivo, e, estando um dispositivo nesta lista, ele estará apto a se comunicar com o outro dispositivo que foi previamente pareado. Parear dispositivos significa estabelecer uma conexão e autenticá-la, permitindo a troca de informações segura entre os aparelhos conectados (JOHNSON, 2004).

Os dispositivos podem realizar uma varredura em busca de outros dispositivos disponíveis, porém cada um deles pode ser configurado para estar visível ou não. Mesmo que um dispositivo esteja invisível, se o que estiver buscando já estiver previamente autenticado, sabendo o endereço do outro dispositivo, é possível estabelecer uma conexão entre eles (YANG, et al., 2005).

Uma aplicação que use Bluetooth pode ser servidora, cliente ou pode criar uma rede P2P, fazendo o papel de cliente e servidor. Na conexão entre dispositivos, o primeiro passo é a inicialização da pilha de protocolos; em seguida o cliente verifica os dispositivos que estão próximos a ele, fazendo uma verificação dos serviços oferecidos por estes dispositivos. Um servidor deve informar os serviços disponíveis para os clientes, para isso ele registra os serviços no SDDB (*Service Discovery Database*) e fica aguardando conexões. Ao finalizar um serviço ele é removido do SDDB (ORTIZ, 2004).

Estruturas de redes Bluetooth

Quando um dispositivo Bluetooth conecta-se a outro, eles formam uma *piconet*, que é uma rede *ad-hoc* constituída de um dispositivo mestre e até sete dispositivos escravos. Uma lista com até 255 dispositivos inativos pode ser criada, e cada um desses dispositivos pode ser posto de volta no estado ativo quando for

necessário. Também é possível criar conexões entre *piconets*, formando uma *scatternet*. Isso ocorre quando um dispositivo mestre em uma *piconet* cria uma conexão com um dispositivo de outra *piconet*, gerando uma ponte entre elas. Essa nova rede formada pela junção de duas ou mais piconets é denominada *scatternet* (ORTIZ, 2004).

No contexto interativo de um CRS aqui abordado, se mais de um aluno necessitar obter conteúdo de um servidor, para que eles interajam simultaneamente, cada um deles deve portar algum dispositivo de interação. O esperado é que esses dispositivos possuam uma tela para que os alunos possam ver as informações transmitidas, sendo um celular o dispositivo adotado neste trabalho. Desta forma, cada aluno se conecta diretamente ao servidor usando seu aparelho, levando os dados de cada um. Assim, o celular se comporta como um dispositivo de entrada e saída de dados, mostrando informações através de sua tela e capturando as entradas do usuário.

Para o gerenciamento das conexões, o servidor estabelece a forma como elas são organizadas e pode requisitar que os dispositivos também façam parte da rede de gerenciamento, deixando de ser unicamente escravos, passando a ser mestres na arquitetura de redes Bluetooth.

Os ambientes interativos podem ser definidos de três formas: um ambiente que possui apenas o servidor e um usuário, um ambiente com um servidor e até sete usuários utilizando um dispositivo de interação, e um ambiente com um servidor e vários usuários interagindo com seus dispositivos. Como um dispositivo Bluetooth possui a limitação de sete conexões, esses ambientes são delimitados de acordo com o limite de conexões de cada servidor.

O primeiro ambiente seria o caso no qual há apenas um usuário interagindo diretamente com o servidor através de um celular, como é mostrado na Figura 4. É o caso mais básico de conexão.

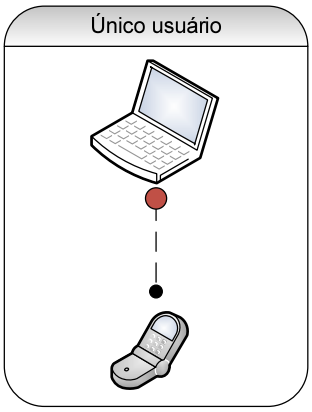


Figura 4 - Ambiente com um servidor e apenas um usuário

No segundo ambiente, da mesma forma que no primeiro, como um dispositivo Bluetooth suporta até sete conexões ativas, é possível a interação de até sete usuários sem necessitar o desenvolvimento de uma forma de gerenciamento mais complexa das conexões. O servidor poderia conectar-se diretamente a cada um dos dispositivos e trocar informações com eles (enviar questões, obter respostas, mostrar resultados dos questionários, etc.). Assim, uma rede *piconet* é criada entre eles, como é ilustrado na Figura 5.

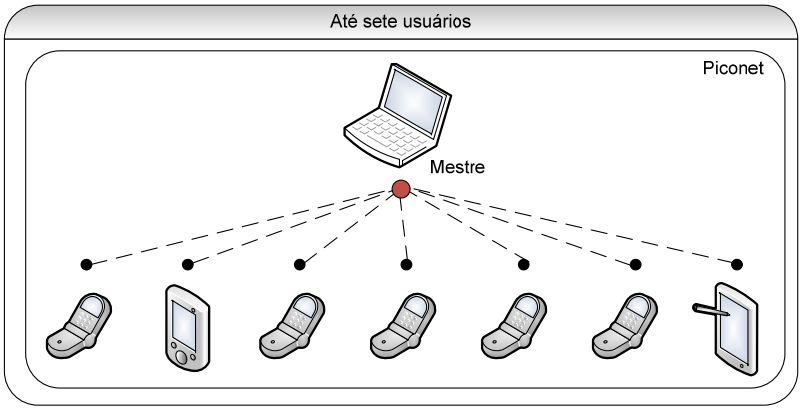


Figura 5 - Ambiente com um servidor e até sete usuários

O terceiro ambiente é o caso mais complexo. Para a troca de informações entre vários dispositivos, redes *piconets* e *scatternets* devem ser criadas para o gerenciamento do trafego de dados. Neste modelo, o servidor (mestre) tentará se conectar a um maior número de dispositivos (escravos) criando uma *piconet*, e estes escravos, por sua vez, irão se conectar a outros dispositivos que ainda não estão conectados, criando outras *piconets.* O dispositivo que era escravo passa a ser um

mestre na *piconet* criada por ele, e desta forma é criada uma *scatternet* (interligação das *piconets*).

Alguns celulares possuem limitações sobre o número de conexões Bluetooth simultâneas, como apenas uma ou duas conexões devido a restrições no consumo de energia (YANG, et al., 2005). Se existirem outros dispositivos que suportem um maior número de conexões como PDA's, computadores, etc. no ambiente, estes serão escalonados como mestres em *piconets*. A princípio são verificados os recursos Bluetooth disponíveis em cada dispositivo, seguida de uma seleção dos mestres de acordo com os mais aptos. Se um dispositivo possuir mais recursos, ele será definido como mestre. A Figura 6 esboça a arquitetura para vários dispositivos comunicando-se com um servidor dentro de uma grande *scatternet*.

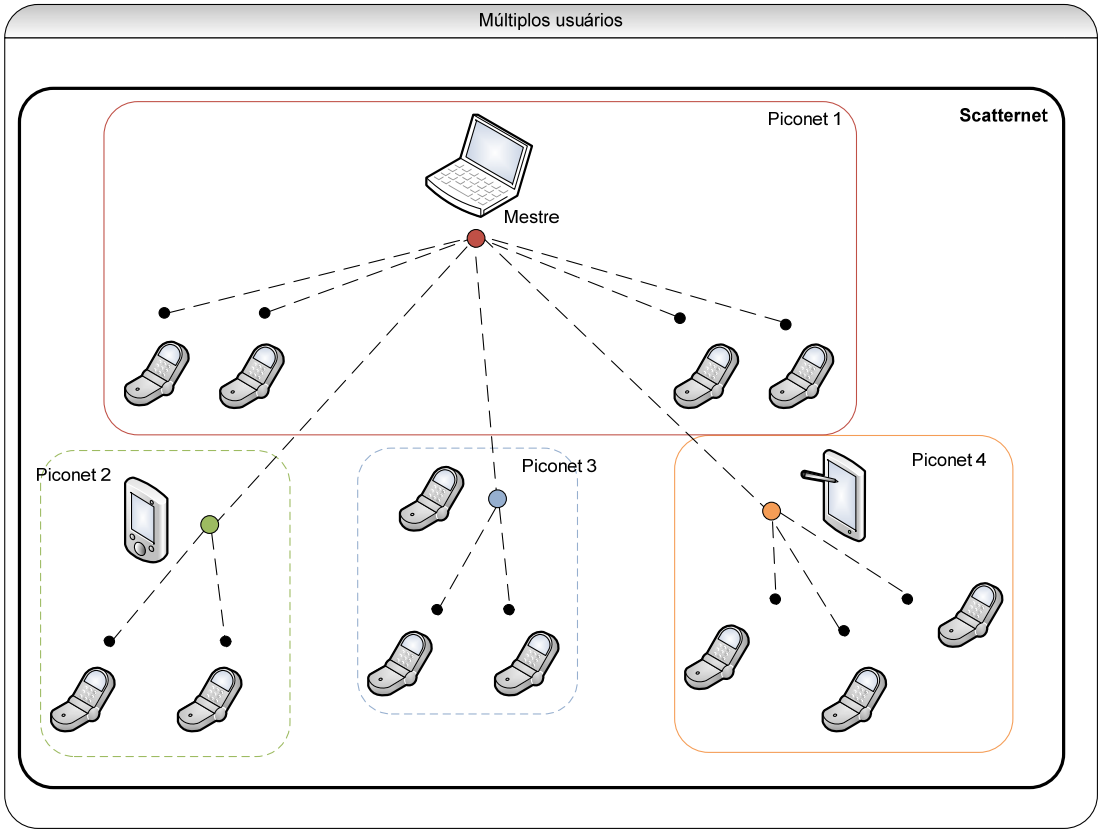


Figura 6 - Ambiente com um servidor e vários dispositivos

O gerenciamento das conexões feito para o terceiro ambiente abrange os casos dos dois primeiros, uma vez que neles necessita-se de um gerenciamento detalhado das conexões realizadas entre os celulares e o servidor. Neste trabalho, a solução implementada pelo Mobile Interact foi desenvolvida baseada nos problemas

observados no terceiro ambiente, gerindo todos os casos independentes do número de conexões nos ambientes mostrados.

Os aplicativos JME a serem instalados nos celulares possuem um algoritmo para organização das redes Bluetooth, pois caso seja necessário criar *piconets* entre os celulares, o aplicativo já estará apto a fazer esse gerenciamento em sintonia com o servidor. Desta forma os celulares poderão ter status de escravos e mestres em diferentes *piconets* (um dispositivo não pode ser mestre em mais de uma *piconet*), gerando uma *scatternet* (YANG, et al., 2005).

Uma opção para possibilitar que todos os celulares se conectem ao um único servidor seria a adoção de filas no processo de comunicação entre os celulares e o servidor. Desta forma não seria necessário um gerenciamento tão complexo das conexões e os celulares poderiam se conectar diretamente ao servidor sem a necessidade da criação de uma *scatternet*.

À medida que os celulares tentam conectar-se ao servidor, os primeiros vão ocupando os slots de conexões disponíveis e os outros vão entrando em espera até que alguma conexão esteja disponível, sendo ordenados pelo *Bluetooth Stack*. Esta arquitetura é mais simples de ser implementada, dispensando a necessidade de gerenciamento mais complexo em relação à entrada e saída de dispositivos numa rede *scatternet,* sendo cada dispositivo independente (SUNKAVALLI, et al., 2004).

Teoricamente, a desvantagem que torna essa opção inviável é o tempo de espera para conexão de cada celular, sendo proporcional ao número de dispositivos tentando estabelecer uma conexão. Exemplificando, no caso do CRS, quanto maior o número de alunos em uma sala de aula, mais demorado é para que todos possam interagir com um servidor.

Apesar desta limitação, foi possível observar que essa opção apresentava bons resultados na prática. Como não é necessário que os celulares transmitam dados continuamente para o servidor, sendo a comunicação feita apenas quando se deseja receber ou enviar novo conteúdo interativo, o tempo necessário para cada troca de informações entre o servidor e um celular é pequeno, fazendo com que eles

ocupem pouco tempo na fila de espera. A Figura 7 ilustra um ambiente utilizando filas de espera.

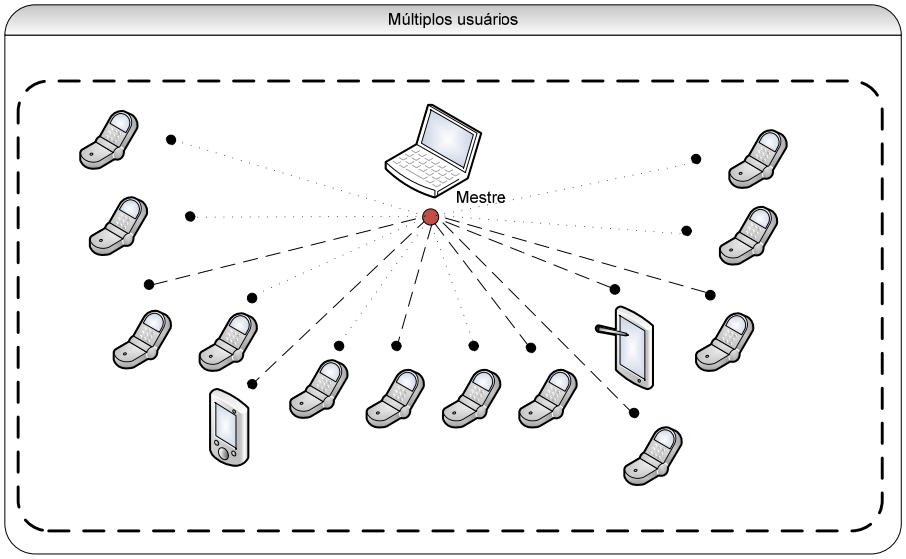


Figura 7 - Ambiente com vários dispositivos e apenas um servidor utilizando filas

3.2.4 Comparativo entre as tecnologias

O Infravermelho teve um importante papel no desenvolvimento dos primeiros CRS, sendo o precursor do Bluetooth. Todavia, suas taxas de transmissão são muito baixas, além de necessitar que os dispositivos estejam numa mesma linha de visão já que utiliza raios infravermelhos para transferência de dados. Outro ponto negativo é que o infravermelho possui uma pequena distância de cobertura: em torno de 1m. Seu custo e seu consumo de energia são baixos, sendo suas principais vantagens (EVERS, et al., 2003). A tecnologia foi um meio de comunicação bastante utilizado para trocas de arquivos por celulares e dispositivos móveis antes da popularização do Bluetooth. Hoje ainda é comum utilizar infravermelho em controles remotos (PAPYRUS, 2009).

Em comparação a Wi-Fi, o Bluetooth possui uma menor área de cobertura e uma menor taxa de transferência de dados, porém é mais barato e possui uma maior economia de energia. Sua área de alcance pode ser um problema em ambientes muito grandes, contudo, na maioria dos casos ele supre bem as necessidades (YANG, et al., 2005).

O Wi-Fi possui um hardware mais robusto, tendo maior consumo de energia. Tanto o Bluetooth quanto o Wi-Fi utilizam a mesma frequência de transmissão, porém empregam esquemas diferentes de multiplexação. Enquanto o Bluetooth é um substituto para o cabo em uma variedade de aplicações, o Wi-Fi é um substituto do cabo direcionado ao uso para acesso à rede local, possuindo objetivos diferentes (MURILO, 2006). Alguns CRS que usam notebooks e PDAs como dispositivos de interação utilizam o Wi-Fi como forma de conexão entre eles por ser a tecnologia mais comum presente nesses dispositivos. Por outro lado, não foi encontrado nenhum CRS que utilize celulares com conexão Wi-Fi, uma vez que não é tão comum encontrar aparelhos com essa tecnologia (PRINCE, 2007).

O Bluetooth é pratico e não exige tanto em termos de configurações para se criar uma comunicação entre dois dispositivos. Já o Wi-Fi exige a configuração e ingresso em uma rede local (há também a possibilidade de criação de uma rede Ad Hoc) para que os dispositivos possam se comunicar (MURILO, 2006). A Tabela 2 mostra um comparativo com as características de cada tecnologia.

	Infravermelho	Wi-Fi	Bluetooth
Alcance (por padrão)	~ 1m	~ 1 a 100m	~ 1 a 10m
Transferência de dados	Baixa	Alta	Média Baixa
Custo	Baixo	Alto	Baixo
Consumo de energia	Baixo	Alto	Baixo

Tabela 2 – Comparativo entre as características das tecnologias

Por ser voltado para comunicações rápidas e entre dispositivos portáteis, e, principalmente por ser a tecnologia mais difundida entre os celulares e outros dispositivos portáteis, o Bluetooth foi adotado neste trabalho como a principal forma de conexão entre os dispositivos de interação e o servidor do CRS. Entretanto, as conexões Bluetooth necessitam de um gerenciamento mais detalhado, pois a tecnologia possui o limite das sete conexões simultâneas.

Como alternativa ao Bluetooth, também foi adotado o Wi-Fi para comunicação entre os dispositivos já que é uma tecnologia que vem se popularizando bastante

entre os dispositivos portáteis. As diferenças no comportamento de cada uma dessas tecnologias serão mostradas com os testes realizados com um exemplo prático no capítulo de Resultados.

4 MOBILE INTERACT

O Mobile Interact é um *Classromm Response System* que utiliza celulares como dispositivos de interação. O sistema torna ambientes de ensino mais interativos, possibilitando que os alunos possam se conectar a um servidor por meio de Bluetooth ou Wi-Fi para obter questionários disponibilizados pelo professor. Para que seja possível realizar múltiplas conexões com o servidor, o sistema cuida do gerenciamento das conexões existentes, controlando a forma como os celulares se conectam ao servidor.

O sistema é composto por três módulos distintos: o servidor, o aplicativo cliente para os celulares, e o módulo de distribuição dos aplicativos clientes baseado em estratégia de Bluetooth Marketing, definidos nos próximos tópicos deste capítulo.

4.1 USO DO SISTEMA

O Mobile Interact é um sistema que possui uma fácil usabilidade, proporcionando um ambiente interativo com vários usuários. O usuário principal do sistema é o professor, que irá controlar o sistema, disponibilizar o conteúdo e analisar as respostas dos alunos.

Primeiramente o professor deve configurar o sistema na sala de aula, iniciando o servidor e instalando os aplicativos interativos nos celulares. Para isso, o professor deve iniciar o módulo de Bluetooth Marketing que envia os aplicativos por Bluetooth para todos os dispositivos encontrados (REIS, 2010). Ao ser iniciado, o módulo envia os arquivos instaladores dos aplicativos para os celulares de forma automática, e ao receber os instaladores, os alunos devem instalá-los nos seus aparelhos.

Os instaladores só precisam ser enviados e instalados nos celulares uma vez, assim o professor não precisará enviar os aplicativos todas as vezes que for apresentar uma aula interativa usando o Mobile Interact. Somente se uma nova

versão dos aplicativos JME for disponibilizada é que será necessário o envio dos novos instaladores aos celulares; do contrário, os aplicativos podem ser executados inúmeras vezes, inclusive sendo utilizados para interação com conteúdos de diferentes disciplinas e professores.

Com os aplicativos instalados em seus celulares, os alunos estão aptos a utilizar os aparelhos para se comunicar com o servidor e receber o conteúdo disponibilizado pelo professor. O professor deve acessar o servidor e informar as questões a serem enviadas aos celulares, iniciando o processo interativo. Em seguida, os alunos devem iniciar o processo de procura por um servidor através do aplicativo instalado e estabelecer a conexão com o mesmo, requisitando as questões disponibilizadas.

Com os vários alunos tentando estabelecer uma conexão com o servidor, este deve gerenciar as várias conexões, organizando a maneira como elas são realizadas. Com a conexão estabelecida, o servidor envia as perguntas aos celulares e os alunos poderão respondê-las individualmente. Em seguida, cada aluno envia sua resposta ao servidor que, por sua vez, irá gerar estatísticas para o professor. O diagrama da Figura 8 ilustra os casos de uso do Mobile Interact.

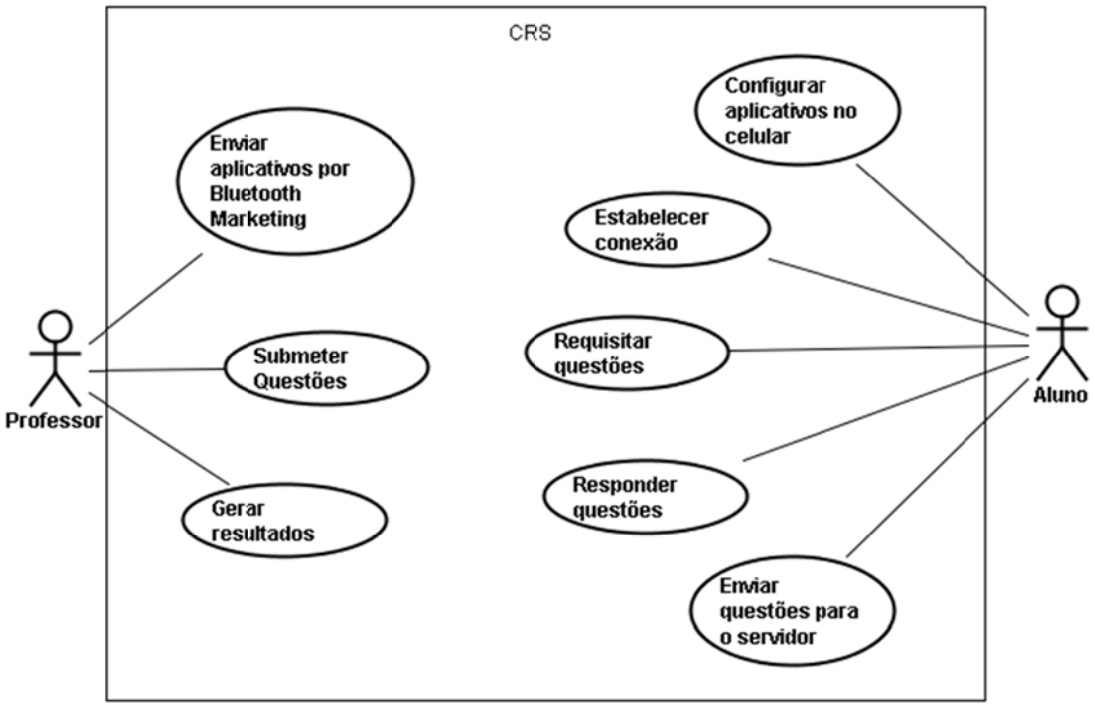


Figura 8 - Diagrama de Casos de Uso

O principal problema do sistema está no gerenciamento das conexões Bluetooth, sendo necessário o desenvolvimento de uma alternativa que possibilite a conexão de vários celulares, contornando o limite de sete conexões simultâneas. Três arquiteturas foram desenvolvidas para o gerenciamento das conexões, sendo detalhadas no próximo tópico.

4.2 ARQUITETURA DO SISTEMA

Como ponto de partida foi definida uma arquitetura base entre os servidores e os celulares a fim de tornar possível a comunicação entre eles e possibilitar que os alunos pudessem interagir utilizando um CRS. Para isto foi feito um levantamento e uma definição dos recursos do sistema, sendo necessários: o desenvolvimento de um aplicativo a ser instalado nos celulares para que os alunos pudessem ler e responder os questionários; uma ferramenta para o envio dos aplicativos JME do servidor aos celulares; uma ferramenta para gerenciamento das múltiplas conexões; e, por último, uma ferramenta para geração de estatísticas e exibição dos resultados respondidos pelos usuários, sendo estes cálculos feitos pelo servidor.

A primeira etapa foi o desenvolvimento do módulo que envia os instaladores dos aplicativos JME para os celulares por meio de Bluetooth. Seguindo a estratégia de Bluetooth Marketing (REIS, 2010), os aplicativos são enviados continuamente para todos os dispositivos Bluetooth que estão dentro do alcance do servidor. Esta estratégia consiste na busca contínua por dispositivos Bluetooth que estejam no alcance do servidor, com o envio de arquivos a estes dispositivos logo quando são descobertos.

Quando o módulo utilizado para o Bluetooth Marketing é iniciado, logo em seguida o arquivo do instalador JME deve ser selecionado. A partir daí, o módulo pode iniciar o envio automático do instalador para todos os dispositivos no alcance. Opcionalmente, o usuário do sistema (o professor) pode enviar o instalador, ou até mesmo outro arquivo, manualmente para um dispositivo específico. O módulo oferece a opção de envio automático ou manual.

No envio manual o sistema faz uma varredura inicial dos dispositivos Bluetooth e mostra os que foram encontrados. O professor deve escolher a qual dispositivos o instalador deve ser direcionado e o sistema solicitará autorização para o serviço de envio de objetos OBEX (*OBject EXchange*) ao celular que irá receber o instalador (ORTIZ, 2004). Quando o usuário autoriza a conexão, o servidor inicia o envio do arquivo e o celular avisará o recebimento do mesmo quando concluído.

No caso do envio automático, depois que um arquivo é selecionado e o sistema inicializado, uma varredura contínua é feita a fim de encontrar dispositivos Bluetooth ativos dentro da área de cobertura do servidor. Enquanto ele estiver sendo executado, a qualquer momento que um dispositivo for detectado, o servidor irá solicitar uma conexão para o envio de objetos OBEX, enviando o arquivo selecionado para o dispositivo encontrado. O sistema realiza a busca por novos dispositivos até que o usuário pare o serviço.

O dispositivo que aceitar a conexão é colocado em uma lista para que não sejam mais realizadas conexões para envio se ele for encontrado novamente. Neste caso um dispositivo poderia permanecer ativo o tempo todo e sempre o sistema iria encontrá-lo e tentaria enviar arquivos para ele. Se um dispositivo foi detectado pelo servidor mas houve algum erro ao estabelecer-se uma conexão com ele, uma nova tentativa de conexão será feita, e, caso o erro volte a ocorrer, ele também será inserido na lista para que novas tentativas de conexão não sejam feitas. Caso o usuário (aluno) não aceite a conexão no celular, ele também será colocado na lista.

O módulo do Bluetooth Marketing é útil em um ambiente com muitos dispositivos, automatizando o envio de arquivos para os celulares. Ele opera no lado do servidor junto ao principal módulo do sistema, o módulo que gerencia as conexões com os celulares e disponibiliza o conteúdo interativo. Este módulo ministra dois serviços distintos: **o servidor Bluetooth** e o **servidor de rede baseado em *sockets***, utilizado para conexão por Wi-Fi.

O servidor para conexão por Wi-Fi cria um serviço baseado em *sockets* em uma porta específica e aguarda que conexões de entrada sejam estabelecidas com o serviço. Os aplicativos JME que forem enviados para os celulares devem possuir

previamente o endereço (o IP e a porta) do servidor na rede, assim ele tentará se conectar ao computador que está executando o serviço.

Para suportar múltiplas conexões o servidor inicia um processo de comunicação em novas *Threads* para cada celular, permitindo que elas sejam executadas independentemente. Em cada uma das *Threads* o servidor gerencia a troca de informações, enviando questões e recebendo as respostas dos celulares. As respostas recebidas são enviadas a outro processo que faz o processamento dos dados e exibe os resultados obtidos, independente da conexão ser feita por *sockets* ou Bluetooth.

Paralelamente ao servidor Wi-Fi, o servidor Bluetooth entra em execução para manipular as conexões feitas através de Bluetooth. A diferença entre os dois serviços está na forma como as conexões são gerenciadas; as questões a serem enviadas, o processamento e a exibição dos resultados são realizados pelas mesmas classes no servidor, não sendo necessária a criação de funções diferentes.

O processo de comunicação em redes Bluetooth difere em alguns pontos das demais redes TCP/IP, necessitando de uma atenção especial. Para o sistema Mobile Interact foram implementadas três arquiteturas para o gerenciamento das conexões Bluetooth: **a arquitetura baseada em filas de espera**, na qual os usuários se conectam diretamente a um servidor; **a arquitetura utilizando *scatternets***, na qual qualquer nó pode ser cliente e servidor em diferentes *piconets*, formando uma espécie de rede Mesh; e **a arquitetura de múltiplos servidores interligados** que utilizam a fila de espera, sendo esta arquitetura uma mescla das duas primeiras.

4.2.1 Arquitetura baseada em filas

Na arquitetura baseada em filas o servidor fica ativo aguardando conexões e, quando um celular tenta estabelecer uma comunicação, o servidor a inicia imediatamente se estiver disponível ou coloca o celular em uma fila de espera até que uma conexão possa ser estabelecida. Desta forma, o único gerenciamento a ser feito é em relação à fila de espera no servidor, analisando quando um celular deve

entrar ou sair dela; contudo, o próprio *Bluetooth Stack* do sistema se encarrega desta tarefa. O *Bluetooth Stack* acessa diretamente o hardware Bluetooth, agindo como intermediário entre os aplicativos que utilizam Bluetooth e o *driver* do dispositivo (ORTIZ, 2004). A Figura 9 ilustra esta arquitetura.

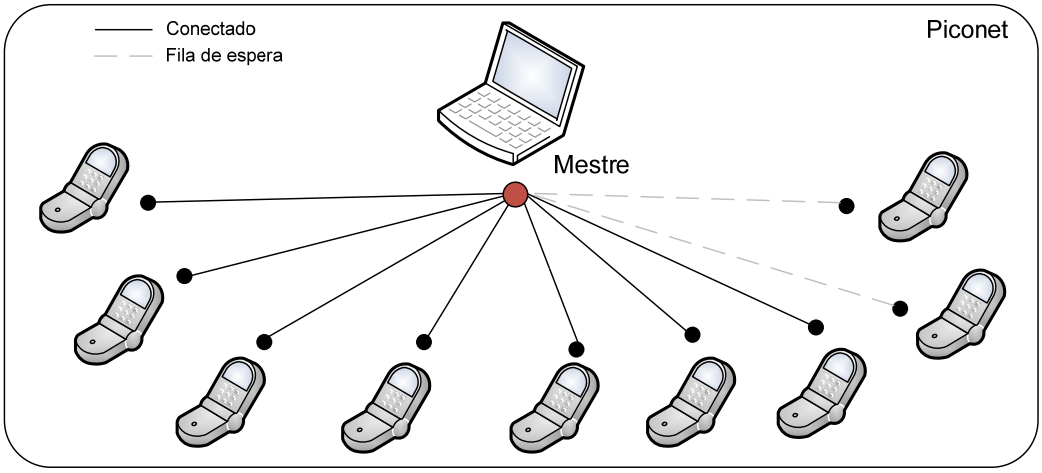


Figura 9 - Arquitetura baseada em filas

4.2.2 Arquitetura utilizando scatternets

Na segunda arquitetura existe uma maior complexidade no processo de comunicação, sendo os celulares pontos indispensáveis na organização da rede. Como o objetivo é fazer com que todos os dispositivos permaneçam conectados, várias *piconets* são criadas e interligadas, criando uma *scatternet,* a partir do momento que novos dispositivos tentam ingressar na rede, reorganizando a distribuição das conexões conforme necessário.

Por exemplo, se o servidor só suporta sete conexões ativas, para que dez celulares estejam conectados, algum dos sete já conectados deve estabelecer uma conexão com os outros três restantes, formando uma *piconet*. Assim, para a informação chegar do décimo celular ao servidor de conteúdo, ela deve ser passada para o mestre da sua *piconet* que, por sua vez, irá repassar a informação ao mestre da primeira *piconet,* chegando ao servidor.

O servidor tem o papel de reorganizar a disposição dos nós (celulares) na rede, indicando qual nó deverá receber conexões de novos dispositivos, criar novas *piconets*, etc. Entradas e saídas constantes de nós geram uma perda no desempenho da rede, em alguns casos aumentando consideravelmente o tempo entre estabelecer uma conexão com o servidor e estar apto a receber uma mensagem do mesmo. Este e outros problemas com essa arquitetura serão abordados em maiores detalhes no próximo capítulo e no capítulo de resultados. A Figura 10 ilustra a arquitetura que utiliza *scatternets*

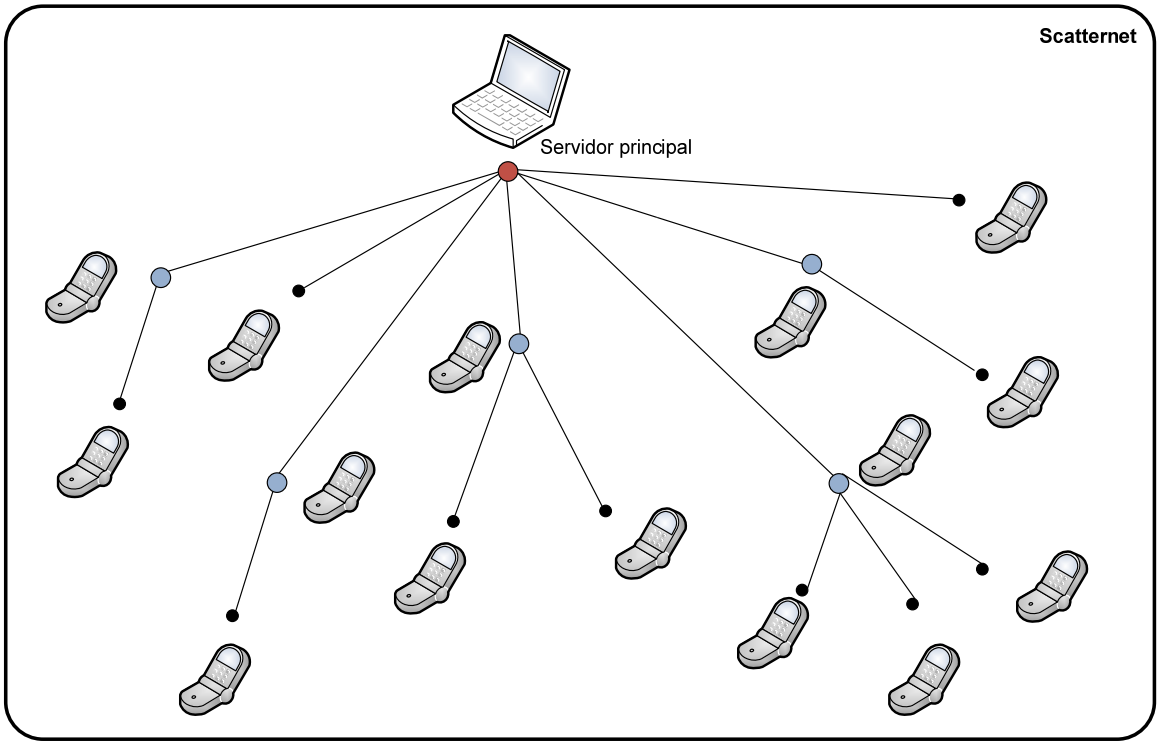


Figura 10 - Arquitetura utilizando scatternets

4.2.3 Arquitetura de múltiplos servidores interligados

Os problemas encontrados com a criação da topologia utilizando *scatternets* e os resultados (detalhados no capítulo Resultados) encontrados com o uso da fila de espera fizeram com que uma nova arquitetura fosse desenvolvida e testada mesclando as duas arquiteturas iniciais. Nela, mais de um dispositivo irá atuar como servidor e os celulares poderão conectar-se diretamente a eles, trazendo como

vantagens a rápida conexão aos servidores e a disponibilidade de várias conexões simultâneas, número este que será relativo à quantidade de servidores disponíveis.

Nesta terceira arquitetura, um servidor atua como o servidor principal e os outros servidores fazem uma busca inicial à procura do mesmo, conectando-se ao servidor principal. Por sua vez, o usuário (professor) interage apenas com o servidor escolhido como principal, sendo que os outros replicam as informações passadas pelo principal. O número de servidores secundários disponíveis deve ser menor que sete, que é o número de conexões Bluetooth simultâneas suportadas pelo servidor principal, caso contrário não será possível realizar a conexão simultânea entre o principal e todos os servidores secundários.

Ao tentar estabelecer uma conexão, os celulares buscam os servidores disponíveis e tentam conectar-se ao primeiro que for encontrado. Caso o servidor coloque-o em uma fila de espera e caso o celular tenha encontrado outro servidor, ele tenta se conectar a este último, não precisando aguardar até que haja uma conexão livre. Se nenhum outro servidor livre estiver disponível, o celular aguarda por uma conexão na lista de espera do último servidor encontrado.

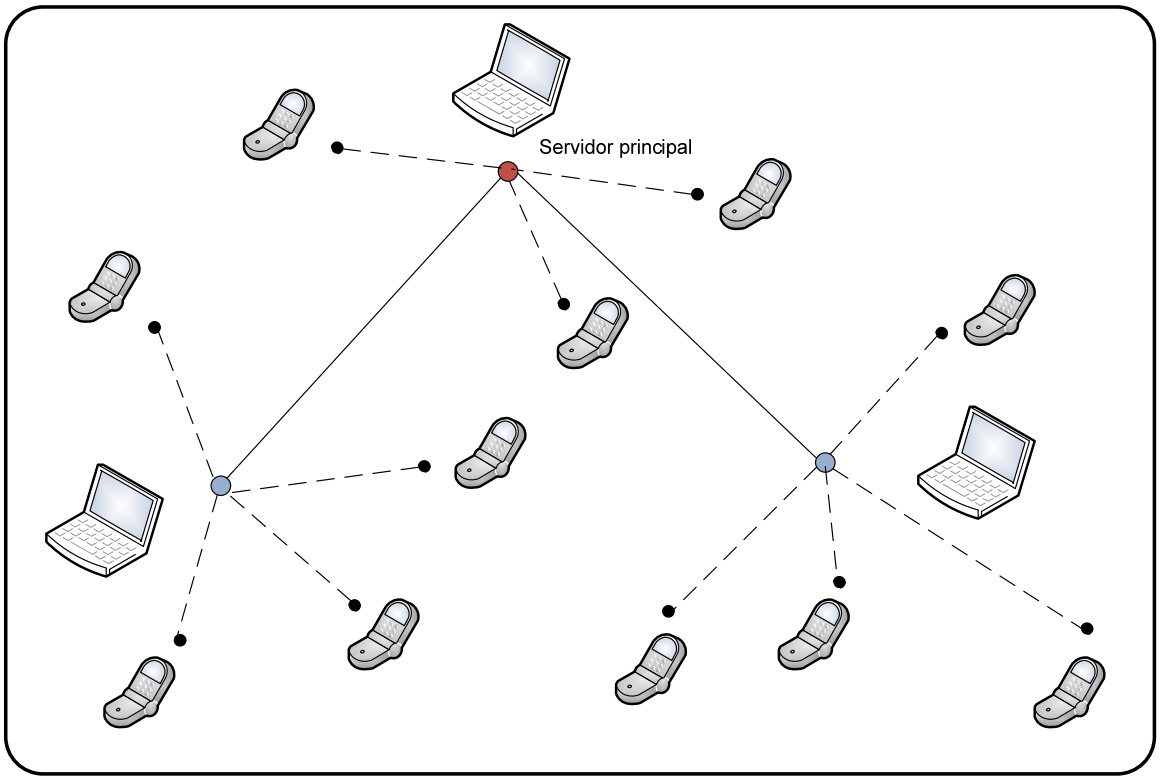


Figura 11 - Arquitetura de múltiplos servidores

O módulo do servidor possui uma interface gráfica que permite ao professor cadastrar questões a serem enviadas e também uma tela para exibição dos resultados respondidos pelos alunos nos celulares. Uma interface mais simples foi desenvolvida para chegar-se aos resultados conclusivos deste trabalho, sendo abstraída a implementação de uma interface gráfica muito detalhada.

4.3 USO DO SISTEMA NOS CELULARES

Independente da forma de gerenciamento das conexões utilizada e como os servidores se comportam, o celular deve executar um aplicativo para poder conectar-se a um servidor e obter o conteúdo interativo. Este aplicativo é enviado do servidor pelo módulo de Bluetooth Marketing, e, a partir do momento que o usuário aceita e instala o aplicativo em seu celular, ele já esta apto a buscar os servidores e iniciar a interação.

De acordo com o método de conexão utilizado pelo usuário, sendo por Bluetooth ou Wi-Fi, o aplicativo busca os servidores e informa ao usuário a situação da busca. Só é possível realizar as conexões por Bluetooth ou Wi-Fi se o aparelho incluir essas tecnologias. Algumas informações do aparelho, como o número máximo de conexões possíveis, versão do Bluetooth, etc., também podem ser visualizadas através do menu principal do aplicativo.

No caso da utilização do Wi-Fi como método de conexão, o celular deve inicialmente estar configurado para acessar a rede local na qual o servidor se encontra. Cabe ao usuário configurar seu celular adequadamente, estabelecendo uma conexão com o roteador Wi-Fi. No momento que o usuário inicializa a conexão e ingressa na rede, o aplicativo busca o servidor pelo IP específico, juntamente com a porta na qual está localizado o serviço. Quando o servidor é encontrado, uma conexão por meio de *sockets* é estabelecida e o celular pode exibir o conteúdo disponibilizado pelo servidor.

Utilizando o Bluetooth para conexão, há uma diferença na forma como os celulares se comportam de acordo com as três arquiteturas implantadas no servidor.

Para a arquitetura que utiliza apenas um servidor e uma fila de espera, os celulares buscam pelo servidor e conectam-se a ele se houver conexões disponíveis, senão ele entra na fila de espera e realiza a conexão assim que uma conexão ativa seja encerrada.

Sendo a arquitetura das *piconets* e *scatternet* a mais complexa, cada celular pode ser cliente e também servidor para outros celulares. Os celulares devem primeiramente fazer uma busca por um servidor disponível (que também pode ser outro celular) e conectar-se a ele. Dependendo da quantidade de celulares e da forma como a rede será organizada, o celular pode ser requisitado a criar uma nova *piconet* e se tornar servidor (mestre) desta piconet, esperando outros celulares requisitarem uma conexão.

A ordem de como os celulares devem se interligar vem do servidor principal: mensagens são enviadas a cada celular contendo o endereço Bluetooth (*Bluetooth Address*) do dispositivo que será o seu mestre (servidor ou outros celulares). O celular estabelece a ligação e informa ao servidor, que responde com uma segunda mensagem dizendo se ele deve formar alguma *piconet*, sendo mestre nela. Todas as mensagens trocadas pelos dispositivos contêm um campo com o identificador dos remetentes e destinatários uma vez que as mensagens podem ser repassadas por dispositivos intermediários.

Na arquitetura que usa uma lista de espera em servidores interligados, o gerenciamento das conexões no celular se assemelha ao da primeira arquitetura: os celulares apenas buscam os servidores e tentam conectar-se a eles. A diferença é que nesta arquitetura o celular procura outros servidores caso o primeiro encontrado não esteja disponível. Se todos os servidores encontrados estiverem com os seus limites de conexões ativas estabelecidas, o celular entrará na fila de espera do último servidor encontrado e irá aguardar pelo estabelecimento da conexão quando outra conexão for encerrada.

Para as três arquiteturas, caso nenhum servidor seja encontrado, o celular refaz a busca até que seja encontrado algum servidor ou até que o usuário cancele o processo de busca. Se a conexão for perdida ou ocorrer algum erro, os celulares

tentam conectar-se ao mesmo servidor. Caso seja o servidor que não esteja mais disponível, por exemplo, se ocorreu algum erro no servidor ou se ele foi desligado, uma nova busca por servidores é feita, reiniciando o processo de conexão.

Independente da forma de conexão utilizada, quando o celular conecta-se ao servidor, este envia o conteúdo interativo ao celular, que por sua vez irá exibir em uma interface apropriada, de forma que os usuários possam interagir adequadamente. Quando os usuários respondem às perguntas, o celular envia as respostas ao servidor para a correção e exibição dos resultados.

5 IMPLEMENTAÇÃO DO SISTEMA

Neste capítulo o sistema desenvolvido é abordado em detalhes, sendo mostradas características da implementação, os ambientes de desenvolvimento e ferramentas utilizadas. Também são expostos diagramas de classes e diagramas de atividades no tópico da especificação do sistema.

5.1 AMBIENTE DE DESENVOLVIMENTO

Observando os celulares como os principais dispositivos portáteis com capacidade de processamento, os programas interativos foram desenvolvidos em uma linguagem de programação que é bastante difundida entre os celulares: o Java JME. Ela foi escolhida por estar presente na maioria dos celulares, além de ser utilizada também por outros dispositivos portáteis com poder de processamento. Sendo o Bluetooth uma das formas de conexão sem fio mais utilizadas pelos celulares, ele é adotado como a principal tecnologia de conexão; como segunda opção o Wi-Fi também é utilizado no Mobile Interact, uma vez que vem se tornando bastante popular entre os celulares e outros dispositivos portáteis.

Para a criação dos módulos do sistema foram utilizadas as seguintes ferramentas de desenvolvimento: um IDE para desenvolvimento de aplicativos Java, uma API que desse suporte a programação utilizando Bluetooth em Java, e uma plataforma para a criação dos aplicativos JME juntamente com um emulador para executá-los.

O IDE adotado foi o NetBeans 6.8, devidamente configurado, dando suporte em todos os módulos de programação Java. O IDE possui ferramentas para o desenvolvimento de interface gráfica, além de ter o módulo do WTK 3.0 (Wireless Toolkit) integrado para o desenvolvimento de aplicativos JME. O NetBeans é um IDE gratuito e seu download pode ser feito pelo site da Sun[4].

[4] http://java.sun.com/

A biblioteca que fornece suporte à programação utilizando a API Bluetooth em Java utilizada foi a Bluecove. Ela é uma biblioteca livre, contudo possui limitações: nem todos os perfis Bluetooth estão implementados e ela só oferece suporte aos dispositivos Bluetooth que fazem uso das *stacks* Mac OS X, WIDCOMM, BlueSoleil ou Microsoft Bluetooth stack (BLUECOVE, 2009). A biblioteca foi adquirida pelo site do projeto Bluecove[5], e em seguida foi adicionada aos projetos do NetBeans que necessitam da comunicação Bluetooth, como no módulo do Bluetooth Marketing e no módulo do servidor Bluetooth, sendo dispensada no módulo do aplicativo para os celulares uma vez que o WTK já possui uma API Bluetooth integrada e funcional para uso em celulares.

Como a biblioteca não controla o dispositivo Bluetooth diretamente, ela utiliza a *Bluetooth stack* disponível no sistema para ter acesso ao dispositivo, desta forma a *stack* trabalha como uma intermediadora na comunicação entre a Java Virtual Machine e o dispositivo Bluetooth (BLUECOVE, 2009). No Mobile Interact os dispositivos Bluetooth utilizados fazem uso da Microsoft Bluetooth Stack, disponível no Windows7, que foi o sistema operacional utilizado no desenvolvimento do projeto. A Figura 12 ilustra a arquitetura da biblioteca Bluecove.

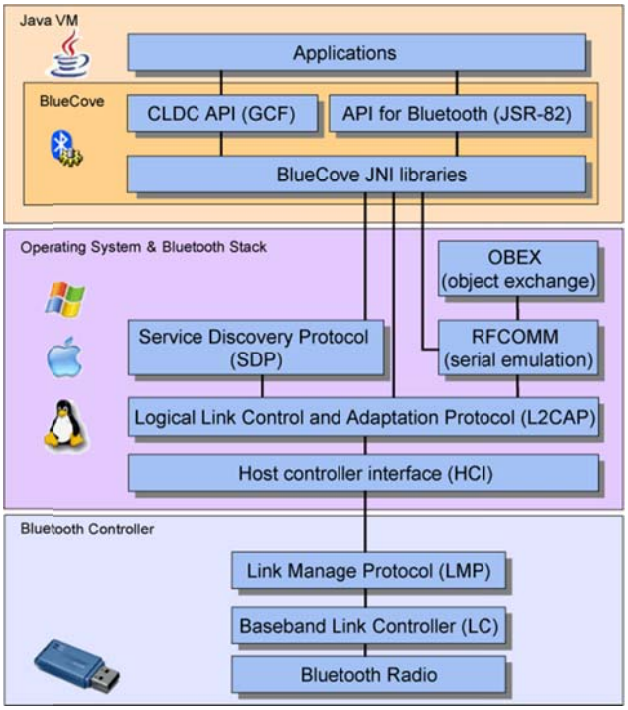


Figura 12 - Arquitetura da biblioteca Bluecove[5]

[5] Fonte: http://bluecove.org/

Outras bibliotecas também estão disponíveis, porém são bibliotecas proprietárias e todas se assemelham à Bluecove, possuindo alguma limitação. Os perfis necessários para o envio de arquivos e para comunicação serial através de Bluetooth estão disponíveis na Bluecove, não havendo problemas na adoção desta biblioteca.

Para o desenvolvimento de aplicativos JME, diferentemente de aplicativos JSE, são necessárias algumas bibliotecas complementares, referentes às Configurações (*Configurations*) e Perfis (*Profiles*) dos dispositivos portáteis. No caso de desenvolvimento de aplicações para celulares, o WTK (*Wireless Toolkit*) da Sun fornece todos os requisitos necessários para a criação de MIDlets (aplicativos JME), como um compilador e emuladores de aparelhos celulares.

A versão dos Perfis do JME escolhido foi o CLDC 1.1 e o MIDP 2.0 por serem as versões mais difundidas para celulares atualmente, além de ser possível a implementação do sistema utilizando o módulo Bluetooth sem ter que utilizar nenhuma API adicional, diferente dos compiladores para aplicações JSE que não possuem a biblioteca Bluetooth integrada.

Como o NetBeans 6.8 já possui o WTK 3.0 integrado, não foi necessária nenhuma instalação e configuração adicional para integrá-lo ao IDE. Um projeto JME foi criado no NetBeans que, por sua vez, cuida da compilação e simulação do aplicativo, fazendo as adaptações necessárias ao ambiente dos celulares. As classes do projeto JME são compiladas pelo NetBeans, sendo gerados os arquivos de instalação dos aplicativos JME, contendo as propriedades de distribuição e atributos de cada um deles. Em seguida eles poderão ser enviados e instalados nos celulares suportados.

Os módulos do Bluetooth Marketing e do servidor também foram desenvolvidos no NetBeans, sendo cada um definido em um projeto próprio. Eles foram compilados individualmente, podendo ser executados de forma independente.

5.2 ESPECIFICAÇÃO E IMPLEMENTAÇÃO

O sistema Mobile Interact foi dividido em três módulos para implementação: o módulo do Bluetooth Marketing, o módulo do servidor e o módulo do aplicativo celular que é o cliente. Para cada um dos módulos foi criado um projeto no NetBeans, sendo os dois primeiros projetos JSE e o último um projeto JME. A Figura 13 ilustra a divisão do sistema.

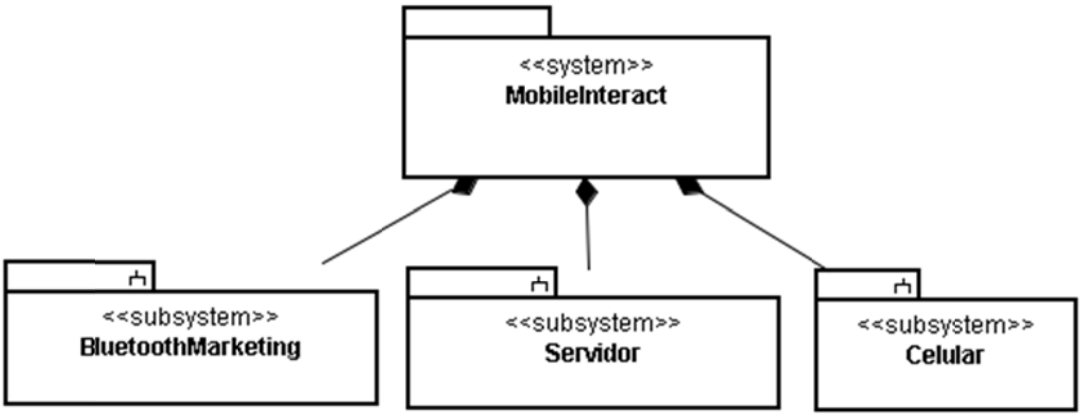


Figura 13 - Sistema Mobile Interact

5.2.1 Módulo Bluetooth Marketing

O primeiro módulo desenvolvido foi o do Bluetooth Marketing, sendo de fundamental auxílio na posterior distribuição dos aplicativos para os celulares. Os módulos do servidor e do aplicativo para celular foram desenvolvidos em paralelo uma vez que eles estão interligados, possuindo uma dependência um do outro.

O módulo do Bluetooth Marketing foi dividido em duas partes: o envio automático de um arquivo para todos os dispositivos, e o envio manual para um dispositivo selecionado. Uma interface gráfica também foi desenvolvida para o módulo utilizando o editor visual do NetBeans.

No envio automático a busca por dispositivos Bluetooth é feita de forma automática pelo controlador Bluetooth, sendo que quando algum dispositivo é encontrado, o servidor irá requisitar uma conexão com o mesmo. Uma busca pelos serviços aceitos pelo dispositivo é iniciada a fim de verificar-se a possibilidade dele

receber um arquivo através de uma conexão OBEX, e se for possível, a conexão é estabelecida e o arquivo é enviado.

Quando o servidor envia o arquivo ao dispositivo, o *Bluetooth Address* do dispositivo é armazenado para que novas tentativas de conexões não sejam realizadas, enviando o mesmo arquivo repetidas vezes. Este ciclo de busca e envio por dispositivos é realizado até que o usuário cancele o processo do Bluetooth Marketing.

No envio manual para um dispositivo específico, o processo de estabelecimento de conexão com o dispositivo é feito da mesma forma que no envio automático. A diferença entre os dois processos é que no envio manual é feita uma varredura dos dispositivos Bluetooth, exibindo os dispositivos encontrados. Com a lista de dispositivos disponíveis, o usuário deve selecionar o dispositivo alvo e iniciar o envio do arquivo escolhido. Logo após a seleção, o estabelecimento da conexão e o início do envio do arquivo são feitos da mesma maneira que no processo automático. A Figura 14 exibe o diagrama de atividades do módulo Bluetooth Marketing e logo após, a Figura 15 exibe o diagrama de classes do módulo. No Anexo I as classes desses diagramas são mostradas em detalhes.

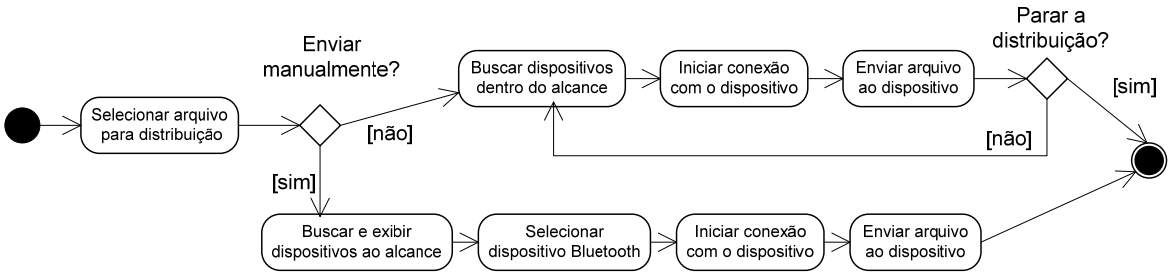


Figura 14 - Diagrama de atividades do Bluetooth Marketing

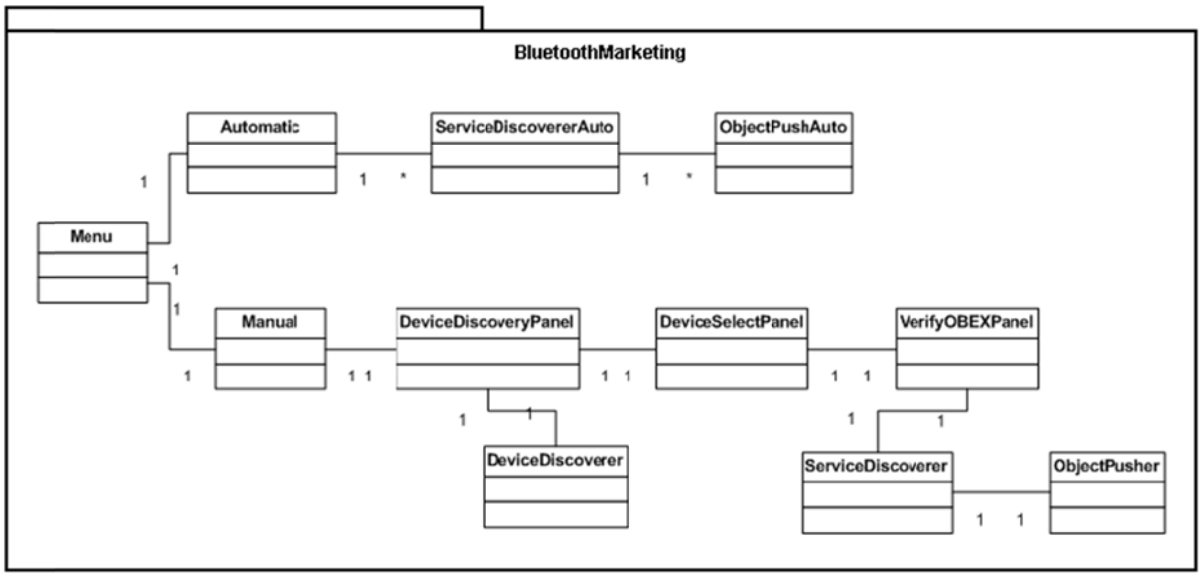


Figura 15 - Diagrama de classes do módulo Bluetooth Marketing

A classe principal do módulo é a *Menu.java*, que exibe as opções de envio manual e automático. No envio automático, a classe *Automatic.java* é instanciada, dando inicio ao processo de envio do arquivo escolhido. Esta classe inicia a busca por dispositivos Bluetooth ao alcance através da criação de um objeto *DiscoveryListener(),* que implementa o método de busca *deviceDiscovered()*.

Quando um dispositivo Bluetooth é encontrado, é feita uma chamada ao método *deviceDiscovered(),* informando o *Bluetooth Address* do dispositivo identificado. O método inicia a busca pelos serviços aceitos pelo dispositivo através da instância de um objeto da classe *ServiceDiscovererAuto.java.*

A classe *ServiceDiscovererAuto.java* também implementa a *DiscoveryListener*, porém somente os métodos *servicesDiscovered()* e *serviceSearchCompleted()* são sobrescritos, métodos esses que são referentes à busca e conclusão da busca por serviços suportados por um dispositivo. Quando é confirmado que o serviço é suportado, uma tentativa de conexão para o envio do arquivo ao dispositivo será realizada, e uma *thread* com um objeto da classe *ObjectPusherAuto.java* será criado para esta finalidade.

A classe *ObjectPusherAuto.java* serializa o arquivo a ser enviado e depois envia-o ao dispositivo através de um objeto *OutputStream*. Quando o envio é

finalizado, a conexão é encerrada, uma mensagem é exibida ao usuário informando o status da transação e o endereço Bluetooth do dispositivo é gravado para que outras tentativas de conexões não sejam realizadas.

No envio manual, o processo foi implementado utilizando classes adicionais para facilitar a criação de uma interface gráfica que permitisse ao usuário selecionar os dispositivos encontrados. Desta forma, a cada etapa no processo de busca e envio, o usuário é informado com os dispositivos encontrados e do status de cada um. Os métodos utilizados pelas classes são praticamente iguais; a diferença está nos métodos de busca por dispositivo e no da escolha individual de um dispositivo para envio, sendo aguardados comandos do usuário para o início de cada um desses processos.

5.2.2 Módulo Servidor

O módulo do servidor implementa dois processos distintos: o servidor Wi-Fi e o servidor Bluetooth. Ao ser executado, o servidor inicia paralelamente os dois processos em *threads* separadas, aguardando por pedidos de conexão. No caso do servidor Bluetooth, o sistema deve gerenciar as conexões de acordo com uma das três arquiteturas utilizada, possuindo diferenças nesta parte de execução.

Após estabelecer as conexões, independente de ser por Bluetooth ou Wi-Fi, o sistema aguarda o professor disponibilizar uma questão e logo disponibiliza-a para o envio aos celulares. Depois que os alunos respondem a questão e envia as respostas ao servidor, este corrige as respostas e gera as estatísticas com o total de acertos, total de erros, etc. O resultado corrigido das questões é enviado aos celulares e as estatísticas são exibidas ao professor. A Figura 16 mostra o diagrama de atividades do módulo Servidor.

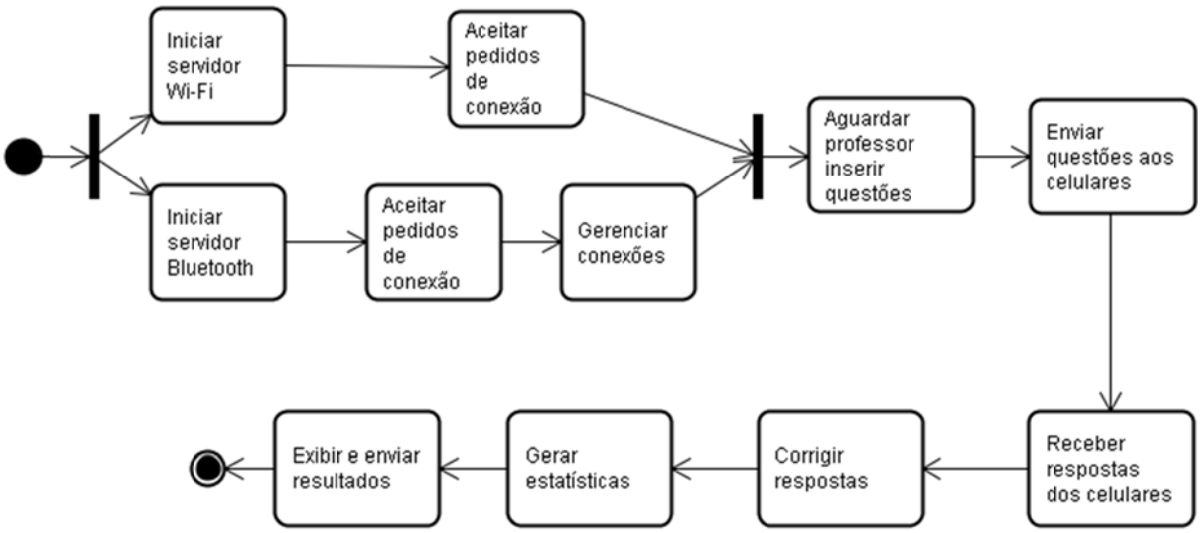


Figura 16 - Diagrama de atividades do Servidor

No servidor Wi-Fi, o gerenciamento das conexões é feito iniciando-se um serviço baseado em *sockets* em uma porta especificada. O serviço inicia uma escuta na porta definida e quando algum cliente requisita acesso ao mesmo, uma nova *thread* é criada, aceitando e manipulando a conexão com o cliente. Essa comunicação cliente-servidor é aberta em *threads* diferentes para que seja possível oferecer suporte a múltiplas conexões sem que elas interfiram uma nas outras. A Figura 17 mostra o diagrama de classes do servidor, sendo instanciada uma classe para o servidor Bluetooth, uma para o servidor Wi-Fi e outra para interface gráfica logo quando o módulo servidor é inicializado.

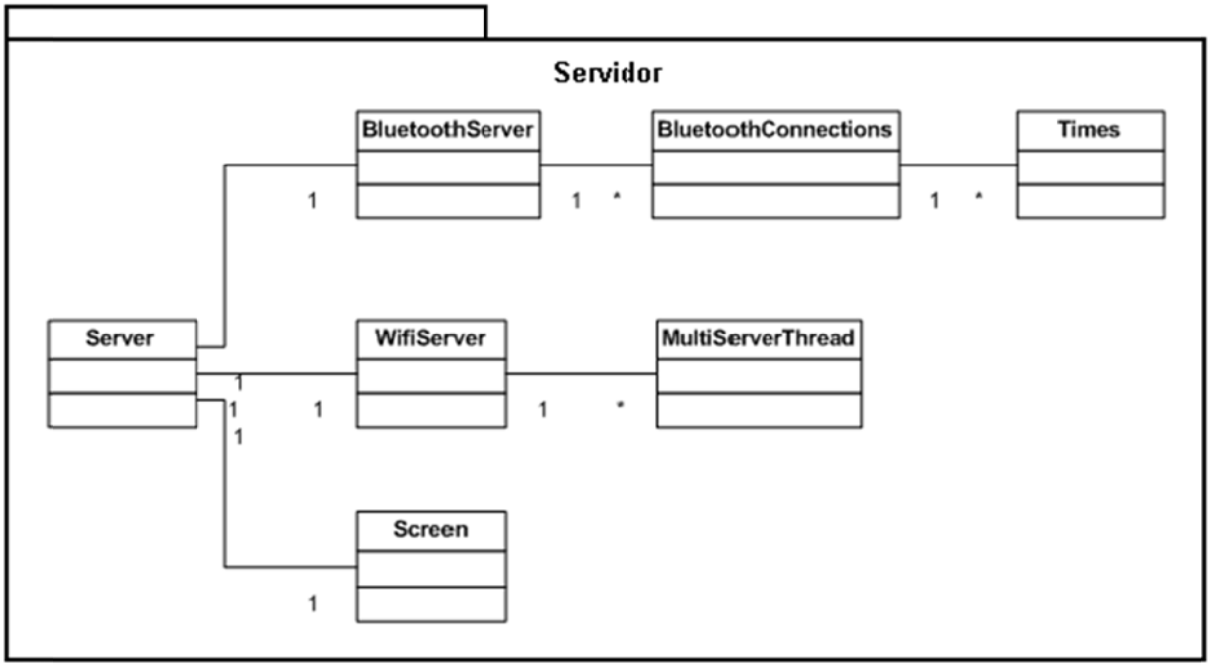


Figura 17 - Diagrama de classes do módulo Servidor

Um objeto *ServerSocket* é criado na classe WifiServer logo quando o servidor Wi-Fi é iniciado, criando o serviço na porta específica e aguardando por conexões. A porta utilizada pelo serviço é a “2222”. Quando a comunicação com o servidor é estabelecida, um objeto da classe *MultiServerThread* é inicializado como uma nova *thread*. Nesta classe, um objeto *DataInputStream* e um *DataOutputStream* são criados e o servidor fica aguardando que o cliente solicite alguma informação. Ao término da comunicação entre o servidor e o cliente, a conexão e o *socket* são finalizados, e a *thread* parada.

Para o servidor Bluetooth inicialmente foi desenvolvida uma interface e uma base comum às três arquiteturas analisadas. Cada arquitetura foi desenvolvida e testada separadamente utilizando projetos diferentes. A base das três arquiteturas foi implementada de maneira semelhante ao servidor Wi-Fi: o serviço é inicializado e fica á espera de pedidos de conexão. Quando chega um pedido de conexão, o servidor dá inicio ao processo de gerenciamento de conexão em uma nova *thread;* assim, o gerenciamento é feito de maneira separada, utilizando uma das três arquiteturas de gerenciamento criadas.

Com a utilização da primeira arquitetura, o servidor apenas aguarda por conexões, estabelecendo-as assim que um celular requisitá-la. A *Bluetooth Stack* irá

gerenciar a fila de espera dos dispositivos Bluetooth que requisitam conexão quando o limite de conexões é atingido.

Na segunda arquitetura o servidor estabelece conexões com o máximo de dispositivos, sendo os novos dispositivos que requisitam conexão redirecionados para conectarem-se a algum dos dispositivos já conectados ao servidor, criando a *scatternet* proposta nesta arquitetura. Sempre que um dispositivo estabelece uma conexão, o servidor informa aos outros dispositivos o ocorrido e, caso seja necessário, envia mensagens para reorganização das ligações entre eles, informando quais dispositivos devem ser ligados ao servidor principal, e quais devem ligar-se com outros dispositivos.

A terceira arquitetura tem uma implementação similar à primeira, contudo, como há mais de um servidor, os servidores secundários devem se conectar com o principal, replicando as informações dele. Na inicialização do módulo, os servidores secundários fazem uma busca pelo servidor principal e iniciam uma ligação Bluetooth entre eles. Essa busca é feita da mesma forma que a busca por um servidor através de um celular: o endereço do servidor principal é previamente conhecido e os outros servidores irão buscar pelo endereço do servidor principal. O servidor principal aceita as conexões e repassa o conteúdo disponibilizado pelo professor aos outros servidores.

Com o conteúdo disponibilizado em cada servidor, a distribuição é iniciada e cada servidor aceitará as conexões requisitadas pelos celulares. As conexões são gerenciadas pela *Bluetooth Stack*, que organiza a fila de espera quando não há mais conexões disponíveis para novos dispositivos.

As conexões Bluetooth são gerenciadas pela classe BluetoothConnections. Nela são criados objetos *DataInputStream* e *DataOutputStream* para a troca de mensagens com o cliente. A classe irá gerenciar as mensagens, processando as entradas e respondendo apropriadamente de acordo com os dados requisitados, agindo da mesma forma que no servidor Wi-Fi.

5.2.3 Módulo Celular

O módulo do celular também implementa dois processos diferentes: o cliente Wi-Fi e o cliente Bluetooth. Ao ser executado, o usuário (aluno) deve escolher a forma de conexão que deseja realizar: seja por Bluetooth ou Wi-Fi, caso estejam disponíveis no celular. No caso do cliente Bluetooth, o aplicativo tentará se conectar diretamente aos servidores na primeira e terceira arquitetura, ou irá se conectar a outro dispositivo podendo também se tornar um servidor para outros celulares na segunda arquitetura. A Figura 18 mostra o diagrama de atividades do módulo do aplicativo para celular.

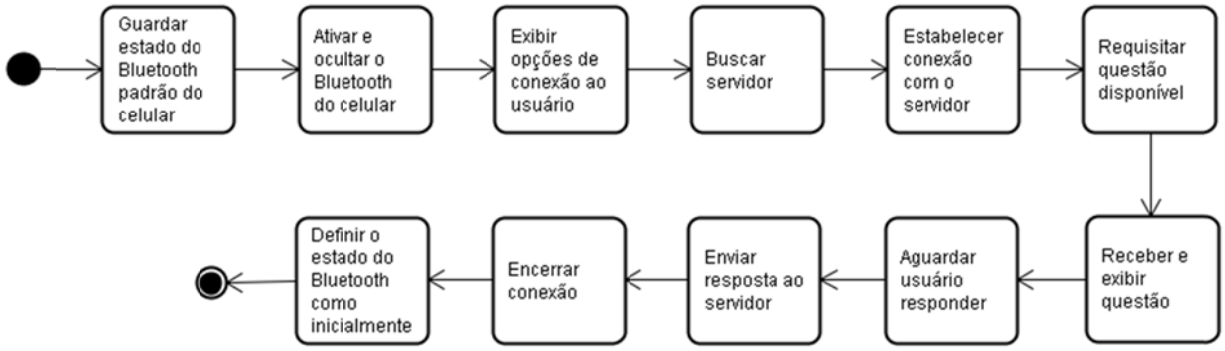


Figura 18 - Diagrama de atividades do módulo Celular

Quando inicializado no celular, o aplicativo irá obter o estado do Bluetooth do aparelho, armazenando em uma variável para posterior modificação. O aplicativo ativa o Bluetooth do aparelho e define o seu estado como oculto, assim outros celulares não o encontrarão e evitarão estabelecer conexões com ele. No caso da segunda arquitetura na qual um celular pode vir a ser um servidor, o estado do Bluetooth é definido como visível para que outros celulares possam encontrá-lo.

O usuário deve escolher o método de conexão utilizada (Bluetooth ou Wi-Fi) e o celular irá iniciar uma busca pelo respectivo servidor. Quando o servidor é encontrado e a conexão é estabelecida, o celular irá requisitar a questão disponibilizada pelo professor no servidor, preparando-se para o recebimento e exibição dela. Quando o aluno responde a questão pelo celular, o aplicativo envia a resposta ao servidor, que irá corrigir e retornar uma resposta. Finalmente, quando a

conexão é encerrada, o estado do Bluetooth do aparelho é definido como era na inicialização do aplicativo.

No lado do cliente (celular), ao tentar estabelecer uma comunicação com o servidor através de Wi-Fi, um objeto do tipo *SocketConnection* na classe *DispositivoMovel.java* será criado e tentará abrir uma conexão por *sockets* com o endereço específico do servidor (este endereço já deve ser conhecido pelo aplicativo). Ao ser estabelecida a conexão, um objeto *DataInputStream* e um *DataOutputStream* também serão criados para o envio e recebimento de mensagens para/do servidor. A Figura 19 mostra o diagrama de classes do módulo celular.

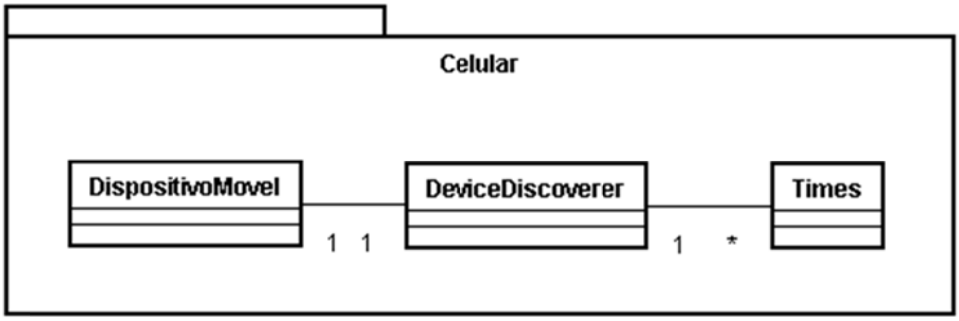


Figura 19 - Diagrama de classes do módulo celular

De acordo com os objetivos do usuário, seja receber questões ou enviar respostas, diferentes mensagens são enviadas como espécie de comando para o servidor, que retornará mensagens apropriadas como resposta. Um método no servidor e outro no cliente irão manipular essa troca de informações entre eles, sendo que cada tarefa possui uma mensagem específica como comando. Por exemplo, se o aplicativo requisita que o servidor envie as questões, ele envia uma mensagem com o código “1” ao servidor. O servidor interpreta o comando e responde com a questão requisitada. Para enviar a resposta do usuário, o aplicativo envia uma mensagem com o código “2” e o servidor interpretará o comando, aguardando o recebimento da mensagem com a resposta.

A base para o lado do cliente Bluetooth também é semelhante ao cliente Wi-Fi, sendo implementado apenas um método de gerenciamento mais complexo das conexões na segunda arquitetura. Quando o usuário inicia a busca por servidores Bluetooth no aplicativo, é criada uma instancia da classe *DeviceDiscoverer* que

implementa a classe *DiscoveryListener*. Esta por sua vez irá gerenciar a forma como o aplicativo irá buscar e se conectar a algum servidor.

Na classe *DeviceDiscoverer* os métodos de busca por dispositivos Bluetooth e por serviços nesses dispositivos são implementados de formas diferentes. Para a primeira arquitetura simplesmente é feita uma busca pelos dispositivos Bluetooth ativos, e o *Bluetooth Address* de cada dispositivo é verificado para saber se aquele é o servidor. Se for o servidor, o serviço que foi definido para a interação entre eles é buscado e se ocorrer tudo conforme planejado, uma conexão será estabelecida entre eles. Caso o número máximo de conexões simultâneas tenha sido atingido, o dispositivo ficará na fila de espera.

Na segunda arquitetura, a busca pelo servidor é feita de maneira geral, não havendo um *Bluetooth Address* específico previamente definido. O celular busca pelo servidor Bluetooth em todos os dispositivos, e quando o serviço é encontrado em algum dispositivo, o celular requisita uma conexão. O celular envia mensagens ao servidor principal para obter dados sobre como ele mesmo deve se comportar na rede: a quem ele deve se conectar diretamente, se deve ser um servidor ou não, etc. Quando o celular souber a quem ele deve se conectar diretamente, uma nova busca com um *Bluetooth Address* específico será realizada e uma conexão será estabelecida com o dispositivo especificado. O servidor informa os outros dispositivos sobre a entrada de um novo dispositivo na *scatternet*, alertando-os para uma possível alteração nas ligações entre eles.

A terceira arquitetura é implementada de maneira semelhante à primeira, sendo o módulo do celular exatamente o mesmo. A diferença está no módulo servidor que inicialmente tem um método para buscar e estabelecer uma conexão com outros servidores no mesmo ambiente, criando assim uma *piconet* entre eles. Os celulares buscam pelos servidores, e, se encontrarem mais de um e se não conseguir conectar-se ao primeiro, o celular tenta conectar-se ao segundo, e assim por diante até que ele consiga conectar-se a algum servidor. Se for o caso de todos os servidores estarem indisponíveis, o celular aguarda na fila de espera do último servidor que foi tentada a conexão.

6 RESULTADOS

Testes com a metodologia Active Learning já realizados em outros trabalhos mostram que ela é uma metodologia eficaz, melhorando o rendimento dos alunos em sala de aula (BOYLE, 2002) (HAKE, 1998). Também já é conhecido que o uso de sistemas CRS provêem a interatividade requisitada pelo Active Learning; desta forma, os testes realizados com o Mobile Interact se detiveram em obter dados que pudessem aprovar o uso do sistema em ambientes reais.

Os testes realizados com o Mobile Interact tiveram como objetivo validar o uso do sistema por vários alunos em sala de aula. Para isso foram observados os tempos necessários para que os alunos conseguissem acessar as informações dos servidores através de seus celulares de forma que os professores e alunos não perdessem muito tempo de aula com o uso do sistema, e também, o número máximo de alunos conectados aos servidores.

Com os testes realizados com o sistema foi possível verificar o comportamento do mesmo na maioria das situações e ambientes físicos. O principal fator que levou a diferentes resultados nos teste foi a variação no número de celulares que fazem parte do sistema, mas outros fatores como área do alcance Bluetooth e os tipos de dispositivos utilizados também provocaram algumas diferenças detalhadas neste capítulo.

Os testes foram feitos em três ambientes distintos: primeiramente foram utilizados emuladores, depois poucos celulares (três a nove celulares) foram utilizados em pequenas salas com apenas um servidor, e por último, os testes foram realizados em sala de aula com um maior número de usuários. Os testes com emuladores e em ambientes com poucos celulares foram feitos para a procura e correção de erros.

Com o desenvolvimento do projeto, os emuladores ofereceram um auxílio na verificação de possíveis erros, porém eles são bastante limitados, principalmente os emuladores de celulares do WTK. O problema destes emuladores do WTK é que

eles não utilizam um dispositivo Bluetooth real do sistema para comunicação com outros dispositivos; eles apenas simulam um dispositivo Bluetooth integrado à plataforma de simulação. Desta forma, os emuladores de celulares não podem se conectar a um servidor, mesmo que ele esteja localizado na mesma máquina; os emuladores só são capazes de estabelecer comunicação com outros emuladores da mesma plataforma WTK. Uma solução para este problema foi a criação de um novo projeto JME para atuar como servidor dentro dos emulados WTK, assim, o código do servidor Bluetooth foi adaptado para ser executado em JME e os testes puderam ser realizados com os emuladores de clientes e servidor.

Os problemas dos emuladores foram referentes à simulações com os módulos Bluetooth; já com o módulo Wi-Fi não houve a necessidade da criação de um projeto JME para atuar como servidor, uma vez que os aplicativos clientes utilizando os emuladores do WTK puderam acessar endereços de rede normalmente. Assim, eles se conectavam da maneira mais real possível ao servidor Wi-Fi utilizando *sockets*.

6.1 TESTES COM O BLUETOOTH MARKETING

Como o módulo do Bluetooth Marketing foi o primeiro a ser desenvolvido, ele foi também o primeiro a ser testado e todas as vezes que era utilizado também era feita uma verificação do seu desempenho, ocasionando em uma gradativa implementação de melhorias.

Os testes com o módulo mediam o tempo de envio de arquivos (os instaladores JME) aos celulares, assim como a eficácia do método, ou seja, se ele conseguia enviar o arquivo para todos os celulares ou não. Nas primeiras versões do módulo, a busca por outros dispositivos estava sendo feita em um longo período de tempo e os envios só iniciavam depois da conclusão da busca, o que gastava muito tempo para o envio. Também ocorriam alguns erros quando um dispositivo não era mais encontrado pelo Bluetooth (saiu da área de cobertura, foi desligado, etc.): o sistema entrava em espera até que o dispositivo fosse encontrado ou encerrava sua execução.

Medições de tempo foram feitas com a realização de cinco testes utilizando a primeira versão do módulo, sendo calculada uma média de 63 segundos no envio para cinco celulares e uma média de 40 segundos para um único celular, sendo que algumas vezes era apresentado erro no envio nos testes com cinco celulares. Algumas alterações foram realizadas otimizando o desempenho do módulo: o tempo de busca de outros dispositivos foi diminuído, e ao termino da busca, outra nova busca era iniciada, entrando em um ciclo até que o usuário parasse a execução do módulo. Com as devidas alterações, logo que um dispositivo for encontrado, uma conexão tentará ser estabelecida antes do término da busca.

Outro ponto que foi ajustado a partir de resultados dos testes foi a definição de uma conexão Bluetooth sem a necessidade de autorização e encriptação dos dados; assim, celulares não precisavam sempre autenticar-se junto ao servidor, poupando o tempo dessa configuração prévia de senhas e autorizações Bluetooth.

Novos testes foram realizados com a versão final do módulo, sendo repetidos dez vezes o envio de um mesmo arquivo para 1, 5 e 13 celulares. Os novos tempos obtidos foram uma média de 30 segundos para um celular, 49 segundos para cinco celulares, e 56 segundos para os treze celulares, sendo os desvios padrão de 2,12; 4,25 e 4,98 respectivamente. Os tempos eram calculados a partir do momento que o módulo de Bluetooth Marketing iniciava os envios, com finalização quando o último celular recebia o arquivo. Todos os celulares estavam dentro de um raio de 10 metros de distância do servidor. Somente no teste com treze celulares que ocorreu um erro no envio do arquivo para um mesmo aparelho em dois testes diferentes, que foi solucionado com o envio manual do arquivo. Apesar de serem tempos longos, o envio dos aplicativos deve ser feito apenas uma vez, assim o professor pode fazê-lo antes de iniciar sua aula.

6.2 TESTES COM O SERVIDOR WI-FI

O módulo do servidor Wi-Fi foi o que se comportou de maneira mais estável, sendo executado de forma idêntica e sem surpresas durante todos os testes

realizados. Inicialmente o módulo foi desenvolvido e simulado em um mesmo computador, não havendo diferenças entre as simulações e os posteriores testes reais.

Como há o limite de sete instâncias em execução dos emuladores, foi criado também um projeto Java SE que não dependesse dos emuladores do WTK para ser executado. Desta forma foi possível colocar em execução vinte instâncias de clientes e uma instância do servidor em um mesmo computador. Foi verificado que o servidor não perdeu desempenho e suportou todos os vinte clientes em execução, apresentando os mesmos tempos para o estabelecimento de conexão.

Um segundo teste mais elaborado foi realizado dividindo as instâncias dos clientes em dois computadores na mesma rede. Foram utilizadas 15 instâncias em cada um dos computadores, todas tentando conectar-se em um mesmo horário especificado. A medição do tempo foi feita com o início da contagem no momento que o cliente começa a procura pelo servidor, e com o final quando a última mensagem contendo uma questão a ser exibida pelo cliente é enviada do servidor ao cliente. O teste foi repetido dez vezes e foi observado o mesmo desempenho, com os mesmos tempos, não havendo diferença significativa entre os testes. Os tempos encontrados foram uma média de 5 segundos nos testes, com um desvio padrão de 0,59.

Com o correto funcionamento do servidor e dos clientes Wi-Fi, um teste real com celulares que possuíam conexão Wi-Fi foi feito, porém em menor escala devido à falta de disponibilidade de aparelhos com essa tecnologia. Quatro celulares foram utilizados simultaneamente e o desempenho foi similar ao desempenho dos computadores executando várias instâncias. O teste consistia em conectar-se ao servidor Wi-Fi e obter o conteúdo interativo, que era uma pergunta com resposta dada em quatro alternativas. O tempo também foi medido a partir do momento que o usuário tenta estabelecer a conexão com o servidor e o momento em que a pergunta com suas alternativas é enviada ao celular.

Para a realização deste teste foi utilizado um roteador D-Link DI524 que possui uma capacidade de 24 conexões wireless ativas (D-LINK). Os celulares

utilizados foram previamente configurados para acessar a rede criada pelo roteador e quando o aplicativo de interação foi executado, os celulares se conectaram ao servidor obtendo o conteúdo disponível. O teste foi repetido 10 vezes e em todos eles, os celulares tiverem tempos semelhantes aos tempos dos emuladores. A média de tempo para conexão também foi de 5 segundos com os celulares, porém o desvio padrão foi bem maior: 3,68.

O tempo final nesses mesmos testes, que incluiu o tempo que o aluno levou para responder à pergunta e enviar a resposta ao servidor, foi uma média de 11 segundos, com desvio padrão de 7,42. Esse tempo final variou muito devido ao tempo que cada usuário levou para responder à pergunta. A pergunta feita foi “Você entendeu a explicação?”, com alternativas “Sim”, “Não” e “Mais ou menos”, sendo apenas para fins de exemplo.

6.3 TESTES COM O SERVIDOR BLUETOOTH

Sempre que havia alguma alteração na implementação dos módulos para os celulares, os programas eram novamente compilados e os instaladores precisavam ser enviados aos celulares para que fossem atualizados. O módulo do Bluetooth Marketing foi de fundamental auxílio nos testes reais, uma vez que poupava tempo, enviando os instaladores a vários celulares rapidamente.

Assim como na arquitetura do servidor que utiliza Wi-Fi, os testes consistiam na medição do tempo para a conexão dos celulares com o servidor. Para comparação dos testes foi medido o tempo para se estabelecer uma comunicação com o servidor, o tempo que o celular levou até receber a questão, e o tempo final do envio da questão respondida ao servidor (este tempo também depende do tempo que o aluno gasta para responder á questão).

Nos primeiros testes realizados utilizando-se os emuladores de celulares do WTK, a primeira arquitetura parecia ser bastante confiável e seu comportamento parecia constante em todos os testes realizados. Esse cenário mudou logo quando a arquitetura foi testada em alguns celulares reais. Os tempos de busca pelo servidor

iam aumentando à medida que novos celulares tentavam estabelecer uma comunicação; além disso, vários erros de conexão eram observados quando o servidor estava próximo de seu limite de conexões, ocasionando uma falha na conexão do aplicativo celular.

Diante dos testes realizados com os celulares, foi possível observar que na prática os dispositivos Bluetooth não se comportavam de forma constante, nem de maneira idêntica aos emuladores. Nos testes eles não suportavam o limite máximo das sete conexões, se tornavam indisponíveis em alguns momentos, entre outros problemas. Os verdadeiros motivos dos problemas podem ter sido ocasionados pelo hardware ou pela API Bluecove para Java. Testes com diferentes equipamentos foram feitos, porém todos se comportaram da mesma forma. Posteriormente, um teste com um aplicativo desenvolvido na linguagem C e que acessa o dispositivo Bluetooth diretamente (sem ser intermediado por uma biblioteca de terceiros) foi feito junto aos dois hardwares utilizados nos testes anteriores, mas todos os resultados foram inconclusivos, uma vez que também ocorriam os mesmos problemas relativos ao Bluetooth observados em alguns testes.

Outros problemas encontrados foram a espera infinita dos celulares caso eles requisitassem uma conexão com o servidor e este não enviasse nenhuma resposta, e o aumento no tempo de busca por dispositivos e servidores caso muitos celulares estivessem com o status do seu Bluetooth visível para todos.

Apesar dos problemas com o Bluetooth, algumas alterações foram feitas para contornar os possíveis erros se eles ocorressem. A primeira alteração feita foi a tentativa de reconexão, caso um celular tentasse se conectar a um servidor e não obtivesse resposta dentro de um período de tempo. Desta maneira os celulares não entravam mais em uma espera infinita pelo servidor e, a partir do momento que ele percebia que um erro tinha ocorrido e o servidor ignorou o pedido de conexão, ele reiniciava o processo e tentava estabelecer uma nova conexão.

A segunda alteração foi a definição do estado do Bluetooth de cada celular como oculto para os demais; assim, quando um celular fazia uma busca por um

servidor, não ficava encontrando outros celulares ativos, melhorando o tempo da busca por dispositivos servidores.

Tudo ocorreu mais uma vez como esperado na simulação, contudo, na prática, nem todos os celulares testados mudaram seus estados para oculto, sendo características de alguns modelos. Isto é devido a cada fabricante de celular desenvolver sua própria maquina virtual Java, ocasionando diferenças na execução de algumas funcionalidades em diferentes aparelhos. Uma outra alternativa a este problema foi continuar utilizando a definição do estado como oculto, caso o celular pudesse alterar seu estado, e ainda atribuir previamente o *Bluetooth Address* do servidor nas aplicações para que os celulares só tentassem estabelecer uma conexão com o dispositivo que possui o endereço específico.

Foi verificado também que, apesar da especificação Bluetooth informar que um dispositivo pode se conectar a até sete outros dispositivos, o hardware Bluetooth utilizado no servidor não conseguia atingir o limite de sete conexões; o máximo observado foi de apenas cinco conexões simultâneas. Todos os outros dispositivos que tentavam se conectar entravam na fila de espera do servidor. Foram testados cinco dispositivos Bluetooth, sendo que quatro deles possuíam hardware diferentes. Todos os dispositivos testados possuíam um máximo de sete conexões simultâneas conforme as especificações dos fabricantes, porém nenhum deles atingiu esse máximo.

O modelo Toshiba Bluetooth Blue582 e os dois dispositivos USB PenBlue 2.0 conseguiram conectar-se a cinco outros dispositivos simultaneamente; já um Bluetooth interno do notebook HP TouchSmart tx2 só suportou três dispositivos. Não foi possível identificar o modelo do quinto dispositivo, sendo um dispositivo genérico. Este último só suportou uma conexão ativa. Apesar de todas as pesquisas realizadas e os testes feitos com o Mobile Interact e outros sistemas, não foi possível identificar o que ocasiona essa limitação no número de conexões.

Mesmo sem a identificação do que ocasionam esses problemas, um teste final com a primeira arquitetura foi realizado para se obter os tempos de conexão com quantidades diferentes de celulares. O teste foi dividido em três etapas: uma

utilizando três celulares clientes, a segunda utilizando cinco celulares e a última utilizando nove celulares, sendo cada uma das etapas realizadas cinco vezes. Todos eles mediram os tempos de conexão entre o celular e o servidor, como anteriormente, e a quantidade de erro apresentada pelos celulares no momento da conexão. Os celulares foram disponibilizados dentro de um raio de 10 metros do servidor.

O teste realizado com três celulares teve tempos semelhantes aos tempos observados utilizando os emuladores do WTK, sendo uma média de 16 segundos e desvio padrão de 2,68. No teste com cinco celulares a média do tempo foi um pouco maior: 20 segundos foram necessários para que os celulares se conectassem, com um desvio padrão de 4,1. Já no terceiro teste, como havia mais celulares do que o número suportado de conexões, a média aumentou significativamente: média de 41 segundos necessários para a conexão com o servidor com um desvio padrão de 11,63.

Feito os testes com a primeira arquitetura, a segunda arquitetura foi desenvolvida e colocada em testes, e, a partir dos testes iniciais, foi observado que ela não era tão eficiente quanto se imaginava. O que parecia ser uma vantagem, interligar os celulares numa grande *scatternet,* estava aumentando muito o tempo que os celulares gastavam até que estivessem aptos a obter e interagir com o conteúdo disponibilizado por um servidor.

Esse aumento do tempo foi devido principalmente à procura de um servidor dentre tantos dispositivos Bluetooth ativos (nesta arquitetura os celulares não ficam com o estado do Bluetooth oculto). Os celulares requisitam algumas informações a todos os dispositivos encontrados, aumentando o tempo de busca. Outro fator para o aumento do tempo é que quando um novo celular ingressa na *scatternet*, esta deve reestruturar-se, criando novas ligações entres os celulares.

Na prática, os tempos dos celulares que se conectavam diretamente ao servidor eram pouco maiores que os tempos encontrados com a primeira arquitetura, porém quando era preciso se conectar a outro celular, o tempo de conexão mais que dobrava. Os resultados encontrados com o teste realizado com seis celulares foram

uma média de 23 segundos com desvio padrão de 3,6, para celulares que se conectavam diretamente ao servidor, e uma média de 63 segundos e desvio padrão 12,4 para um celular que se conectava a outro celular. Para o mesmo teste realizado com nove celulares, as médias foram de 26 segundos e desvio padrão de 4,1 para os celulares que se conectavam diretamente ao servidor e 71 segundos com desvio padrão de 15,8 para os celulares que se conectavam a outros celulares. No teste com nove celulares foi observado a ocorrência de erro na conexão com três celulares diferentes, sendo dois no mesmo teste e o terceiro em um outro teste.

Também foi observado nos testes que o limite de conexões da maioria dos celulares é baixo, geralmente uma ou duas conexões, não sendo viável para a criação piconets sendo os celulares mestres delas, uma vez que eles poderão ter um número pequeno de conexões com outros celulares (escravos).

A partir dos resultados obtidos, algumas tentativas de melhorias foram feitas, mas sem resultados significativos; com isso, a arquitetura foi revisada e foi desenvolvida a terceira arquitetura baseada na junção das características da primeira com a segunda arquitetura.

Na terceira arquitetura não houve como realizar os testes utilizando os emuladores WTK pois não foi possível emular servidores em diferentes computadores. Todos os testes foram feitos utilizando celulares reais e dois computadores como servidor.

Os resultados alcançados foram bons, com tempos semelhantes aos da primeira arquitetura, mas com um maior número de celulares conectados simultaneamente. Os testes realizados com esta arquitetura foram divididos em duas etapas, uma utilizando três celulares e a outra utilizando o máximo de celulares disponíveis, que foram treze celulares. A segunda etapa foi realizada numa sala de aula com vinte alunos, sendo um teste mais elaborado e real. É importante frisar que nem todos os alunos na sala de aula tinham celulares compatíveis, e apenas treze dos vinte alunos puderam interagir com o Mobile Interact através de seus celulares.

Os resultados obtidos com a primeira etapa dos testes foram em média 16 segundos com um desvio padrão de 1,5 para que os três celulares se conectassem e pudessem interagir com o servidor. O teste foi repetido cinco vezes e os celulares estavam no raio de 10 metros de distância do servidor. Não houve erro de conexão em nenhum dos testes.

Na segunda etapa dos testes foi possível observar que nem todos os celulares ficaram definidos com o estado Bluetooth oculto, o que aumentou alguns segundos no tempo de conexão, uma vez que mais dispositivos eram encontrados a cada busca. Estes foram os testes mais elaborados, feitos em uma sala de aula real com área de 10 por 18 metros. Inicialmente apenas um servidor foi posicionado próximo ao quadro, na parte da frente da sala, assim, os alunos com celulares no final da sala não conseguiam conectar-se ao servidor. O segundo servidor foi então posicionado no centro da sala e todos os alunos puderam conectar-se ao sistema. A Figura 20 demonstra a disposição dos servidores e dos alunos com celulares na sala de aula.

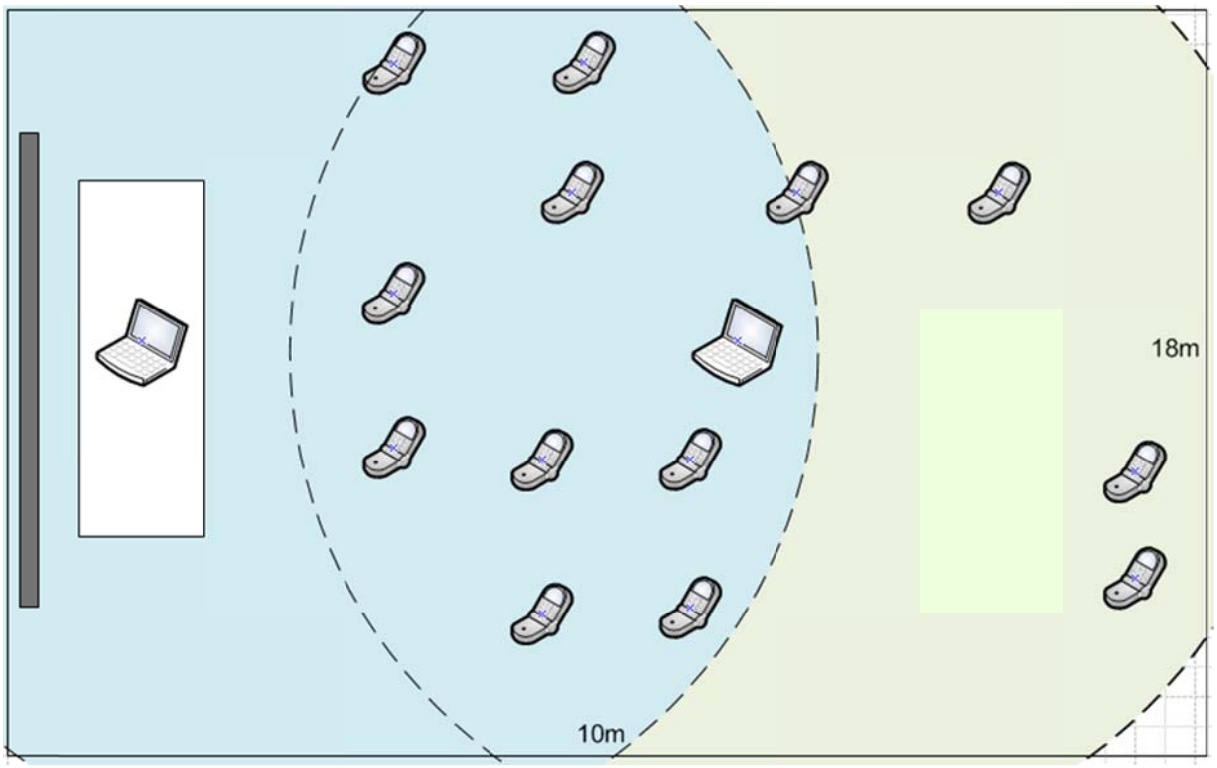


Figura 20 - Ambiente de teste

No teste foram utilizados treze celulares, sendo repetido 10 vezes utilizando a mesma pergunta para o aluno: “Você entendeu o assunto?” com as opções de

resposta “Sim”, “Não” e “Mais ou menos”. Mais uma vez a medição dos tempos foi feita desde o inicio das conexões até quando os alunos respondem e enviam as respostas ao servidor. O tempo é obtido de forma automática pelo sistema e salvo em um arquivo para geração destes resultados. O *Bluetooth Address* de todos os dispositivos são salvos para diferenciar os celulares em cada teste.

Neste teste foram coletados quatro tempos: o tempo para busca de dispositivos servidores, o tempo que o celular tenta iniciar a conexão, requisitando a questão, o tempo que a conexão é estabelecida e o celular recebe a questão, e o tempo que o aluno envia a resposta da questão para o servidor. O tempo médio observado para busca pelos servidores foi de 11 segundos, com desvio padrão de 8,7. O segundo tempo médio, o tempo para o celular se conectar e requisitar a questão foi em média de 13 segundos, com desvio padrão 7,6. O terceiro tempo teve uma média de 24 segundos, com desvio padrão 9.2. O último tempo, que é calculado baseado na interação do usuário, variou muito, sendo uma média de 34 segundos e desvio padrão de 16,7. Foi observado que durante a execução de 2 testes, dois celulares diferentes não conseguiram estabelecer a conexão com o servidor, sendo reiniciado o processo.

Outro fato importante observado nos teste foi que com o uso de dispositivos diferentes nos servidores, um deles possuindo um maior alcance, os celulares localizavam este dispositivo primeiro e tentavam conectar-se a ele. As posições dos servidores foram invertidas e foi observado o mesmo efeito: o servidor de maior alcance ainda foi mais acessado que o outro, mas desta vez, dois celulares que estavam mais próximos ao outro servidor se conectaram a ele.

Já com o uso de dois dispositivos Bluetooth idênticos nos servidores não havia um padrão para conexão entre os celulares e servidores. Algumas vezes o mesmo celular se conectava ao servidor mais próximo, outras vezes ao servidor mais distante. O teste foi refeito com o uso de apenas um aparelho de cada vez para que não houvesse interferência com a limitação no número de conexões e os resultados continuaram os mesmos.

6.4 ANÁLISE GERAL

A Tabela 3 a seguir mostra os tempos médios e desvio padrão para conexão em cada módulo do sistema, comparando os resultados. Alguns testes foram feitos com quantidades diferentes de celulares, assim a tabela está com alguns campos em branco.

	1 Cel.	3 e 4 Cels.	5 e 6 Cels.	9 Cels.	13 Cels.
Bluetooth Marketing	30s / 2,12		49s / 4,25		56s / 4,98
Wi-Fi		5s / 3,68			
1ª Arquitetura		16s / 2,68	20s / 4,1	41s / 11,63	
2ª Arquitetura			63s / 3,6	71s / 15,8	
3ª Arquitetura		16s / 1,5			24s / 9,2

Tabela 23 - Comparativo entre os tempos dos testes

Os testes feitos com a primeira arquitetura utilizando as filas de espera Bluetooth mostraram que a arquitetura apresentava um desempenho aceitável no ambiente de um CRS interativo, principalmente quando poucos usuários tentavam se conectar ao servidor. O maior problema desta arquitetura é o suporte a um pequeno número de usuários, e, à medida que cresce o número de usuários com celulares no ambiente, o tempo para se estabelecer uma conexão vai crescendo proporcionalmente.

Para a segunda arquitetura, os testes mostraram que ela é uma arquitetura bastante complexa e não trazia diminuições significativas nos tempos de conexão, e, pelo contrário, os tempos até aumentavam em alguns casos. A vantagem dessa arquitetura seria sua utilização em ambientes que necessitam de uma conexão contínua, como no caso de transmissões de áudio, porém não é exatamente o caso desse trabalho, no qual os celulares se conectam aos servidores apenas para receber e enviar as questões disponibilizadas.

Mesmo com a utilização da arquitetura em transmissões contínuas, a taxa de transferência de dados do Bluetooth não é alta (1mb); assim não seria possível

utilizar uma conexão para um número muito grande de dispositivos transmitindo altas taxas de dados. O melhor, nesse caso, seria utilizar o Wi-Fi, principalmente em servidores com conteúdos multimídia.

Outro problema que foi observado com a utilização de celulares para a criação de *piconets* é que a maioria dos celulares suporta apenas uma ou duas conexões Bluetooth ativas devido a estratégias para baixo consumo de energia, como o SonyEricsson K790i. Alguns suportam mais conexões, como o Nokia N96 e N97, contudo não são celulares populares e estes modelos possuem Wi-Fi. Assim, não seria vantagem utilizar essa arquitetura com celulares que possuem limitações quanto ao número de conexões ou celulares que já possuem Wi-Fi.

Os resultados encontrados com a utilização da terceira arquitetura foram bastante adequados. Ela mostrou-se uma arquitetura eficiente e que pode ser utilizada sem a necessidade de um complexo gerenciamento de conexões, diminuindo o tempo para a comunicação entre o servidor e os celulares.

7 CONCLUSÃO

Com o término da implementação e com os testes realizados com o Mobile Interact, foi possível chegar à conclusão de que o objetivo inicial, que era criar um CRS que possibilitasse a interação de vários alunos simultaneamente com um servidor através de celulares, foi cumprido e que o sistema pode ser utilizado em ambientes educacionais, auxiliando os professores com metodologias de ensino interativas.

O uso do CRS proposto utilizando celulares mostrou que os professores poderiam criar aulas mais dinâmicas, envolvendo os alunos de forma mais interativa, como é proposto pela metodologia pedagógica do *Active Learning*. Módulos para o servidor e aplicativos para os celulares foram desenvolvidos, sendo que no servidor, três arquiteturas foram desenvolvidas para o gerenciamento das conexões Bluetooth.

Realizados os testes com os celulares junto às arquiteturas criadas, de forma inesperada a arquitetura do servidor Bluetooth que parecia ter um baixo desempenho, a arquitetura baseada em filas de espera, mostrou-se eficiente na conexão dos celulares, superando até mesmo a arquitetura que parecia ser a mais confiável e adequada, a arquitetura baseada em scatternets, que nos testes apresentou tempos elevados para se estabelecer uma conexão.

Logo nos primeiros testes da segunda arquitetura notou-se um baixo desempenho em relação à organização da rede, trazendo tempos de conexão superiores aos da primeira arquitetura, mesmo quando um servidor estava com todas as conexões ocupadas e um cliente entrava na fila de espera. Assim, um novo caminho foi criado baseando-se na otimização da primeira arquitetura, sendo desenvolvida uma nova arquitetura baseada na primeira, mas contando com mais de um servidor. Com a elaboração desta terceira arquitetura, a arquitetura baseada em scatternets foi deixada de lado e o desenvolvimento se concentrou na obtenção de melhores resultados com a nova arquitetura.

As arquiteturas Bluetooth que utilizaram fila de espera apresentaram um bom desempenho, entretanto a arquitetura baseada em *sockets* que utiliza Wi-Fi mostrou-se muito mais confiável e eficiente. Essas vantagens são devidas ao grande alcance de um ponto Wi-Fi comparado ao Bluetooth, e seu limite de conexões, que geralmente varia com os modelos de roteadores, são consideravelmente maiores que os limites do Bluetooth.

O problema em se usar Wi-Fi é referente ao pequeno número de celulares que possuem esta tecnologia. A maioria dos celulares que possuem alguma tecnologia de comunicação sem fio inclui somente o Bluetooth; somente os aparelhos mais modernos possuem Wi-Fi, não sendo viável para a criação de um sistema que tenha como objetivo ser utilizado por todos.

Outro problema também está relacionado ao uso da tecnologia, pois não são todos os usuários que possuem um celular que suporte Java e tenha Bluetooth ou Wi-Fi integrado. Nos testes realizados, praticamente todos os alunos tinham celulares, no entanto apenas cerca de 30% possuíam celulares que pudessem ser utilizados nos testes junto ao Mobile Interact.

Um ponto importante de ser mencionado é que apesar da confiabilidade do sistema, seu uso não deve ser feito de maneira única e obrigatória, uma vez que alunos podem não possuir tecnologia adequada para interagir com o sistema. O Mobile Interact deve ser utilizado de forma opcional, complementando conteúdos vistos em sala de aula, levando mais interatividade e dinâmica no ensino, fazendo com que os alunos interajam mais com o professor.

Com o modelo proposto neste trabalho, aplicações finais poderão ser desenvolvidas e utilizadas em ambientes reais, além de que também se faria possível a utilização da arquitetura desenvolvida em outras áreas, como na TVDI, o que traria novas possibilidades para os usuários. O Mobile Interact é um sistema que cumpriu seus objetivos mas o ideal é que ele seja testado de maneira geral antes de sua implantação a fim de se conhecer seu comportamento em determinado ambiente e com características específicas de hardware.

7.1 TRABALHOS FUTUROS

Com a adoção de um padrão de TV Digital desenvolvido no Brasil, várias pesquisas vem sendo feitas com base nas possibilidades de interação dos usuários com os programas televisivos através de um receptor de TVDI e dispositivos periféricos. O *middleware* Ginga, que é o *middleware* desenvolvido e adotado pelo padrão brasileiro, vem adotando várias tecnologias e conceitos a fim de permitir uma maior interatividade por parte dos usuários. Muitos projetos estão sendo desenvolvidos utilizando todos esses recursos disponíveis.

Um problema da interação proposta pelo ambiente de TVDI é a interação simultânea por vários usuários, problema que não poderia ser gerido por apenas um dispositivo de controle, como um controle remoto por exemplo. O ideal seria que cada usuário pudesse interagir individualmente, obtendo resultados de forma particular, sem interferência de outros usuários que estão acessando o conteúdo interativo no mesmo instante. É possível imaginar a seguinte situação: três telespectadores estão assistindo a um programa quando um questionário é disponibilizado, eles devem respondê-lo para que aprendam mais sobre suas personalidades. Se todos os usuários utilizarem um mesmo meio para responder, ou mesmo se os resultados forem exibidos na mesma tela, alguma confusão nos dados pode ser gerada.

Um sistema no qual cada usuário responda e veja seus resultados pode ser implantado pelo uso de um dispositivo de interação com tela para cada um deles. Utilizando celulares neste ambiente da TVDI, seria possível gerar uma maior individualidade na interação com aplicações digitais. Os celulares possuem um poder de processamento que atendem aos requisitos básicos para execução de aplicativos que possam ser distribuídos em programas da TVDI, além de ser um dispositivo pessoal e bem disseminado.

Com a finalização das normas do *middleware* Ginga e a partir da criação de ambientes de desenvolvimento completos para o padrão brasileiro, será possível adaptar e utilizar a ferramenta Mobile Interact para prover o gerenciamento de

múltiplas conexões e possibilitar que vários usuários utilizem seus celulares para interagir com programas televisivos.

A principal dificuldade de adaptar a ferramenta atualmente é a falta de um ambiente de desenvolvimento completo, principalmente utilizando as normas do Java DTV, criadas pela Sun para serem utilizadas pelo Ginga, que ainda não possui nenhuma implementação disponível. Também não há um emulador eficaz da plataforma, sendo impossível simular um ambiente real.

O Mobile Interact foi desenvolvido em Java, sendo totalmente compatível com as normas do Java DTV. As únicas alterações necessárias serão referentes à interface gráfica do sistema interativo, uma vez que os padrões de interface para televisão são diferentes dos padrões de interfaces para computadores. Os módulos principais referentes ao gerenciamento das conexões e ao Bluetooth Marketing não necessitarão de modificações, podendo ser facilmente portados.

Outro possível projeto seria o uso do Mobile Interact em servidores online que poderiam ser acessados utilizando a internet móvel dos celulares. Assim, o módulo servidor que utiliza *sockets* para criar uma comunicação por Wi-Fi poderia ser executado como um *Web Service* em um servidor online, e seu endereço disponibilizado para os celulares. Nada muda no aplicativo instalado nos celulares; apenas o endereço físico do servidor, que antes estava localizado numa rede local, na qual um celular podia ingressar utilizando Wi-Fi, passa agora a ser um endereço online. A única exigência é que o celular possua um plano de dados para conexão com a internet, possibilitando a conexão em qualquer lugar.

O Mobile Interact poderia ser utilizado para testes feitos a qualquer momento, sendo empregado em ambientes de ensino à distância, em questionários divertidos de conhecimentos gerais ou mesmo como avaliações físicas e outras avaliações do dia-a-dia. Inúmeras possibilidades são abertas com a utilização do sistema Mobile Interact.

REFERÊNCIAS

ABRAHAMSON, A. L. Teaching with Classroom Communication System – What it Involves and Why it Works. *VII Taller Internacional Nuevas Tendencias en la Ensenanza de la Fisica*. Puebla, México, 1999.

ANATEL**.** *Controle de Acessos de Serviço Móvel Pessoal*. Disponível na Internet: http://sistemas.anatel.gov.br/SMP/Administracao/consulta/AcessosMoveisOpDensidade/tela.asp. Acesso em: 03 de 03 de 2010.

BAQER, M.; KAMAL, A**.** ACRS: An Advanced Classroom Response System Architecture for Learning Enhancement*. Second International Conference on Developments in eSystems Engineering (DESE)*, 2009. 10.1109/DeSE.2009.44.

BEATTY, Ian. Transformim Student Learning with Classroom Communication Systems. *Research Bulletin. Center for Applied Research*. Vol. 2004. Edição 3. 2004.

BLUECOVE. Bluecove Documentation. Disponível na Internet: http://code.google.com/p/bluecove/wiki/Documentation. Acesso em: 21 de junho de 2009.

BOYLE, J.; et al. Experience with classroom feedback systems to enable socratic dialogue in large engineering classes. *Engineering Education 2002: Professional Engineering Scenarios.* 2002, Vol. 1, 1.

CHEN, Yu-Fen et al. *Elementary science classroom learning with wireless response devices implementing active and experiential learning.* s.l. : IEEE International Workshop on Wireless and Mobile Technologies in Education, 2005. 10.1109/WMTE.2005.22.

CUNHA, Felipe; SAITO, Thomaz. A REDE WIRELESS: Uma nova ferramenta como auxílio ao sistema de informação da produção. 2005. Disponível na internet: http://www.malima.com.br/article_read.asp?id=242. Acesso em: 9 de abril de 2010.

EVERS, Christine et. al. *Bluetooth vx. Infrared - A Comparison*. Disponível na internet: http://www.faculty.iu-bremen.de/birk/lectures/PC101-2003/17bluetooth/bluetooth/intro.html. Acesso em: 10 de julho de 2009.

FERREIRA, Leydson Pontes. *Utilizando Dispositivos Móveis e Bluetooth para Aplicação de Avaliações em Meio Digital*. Relatório de conclusão de curso (Engenharia da Computação). Natal: UFRN, 2009.

HAKE, R. Interactive-engagement Versus Traditional Methods: A Six-thousand-student Survey of Mechanics Test Data for Introductory Physics Courses. *American Journal of Physics*, vol. 66, Jan. 1998.

INFOWESTER. *Tecnologia Bluetooth*. Disponível na internet: http://www.infowester.com/bluetooth.php. Acesso em: 3 de agosto de 2009

INFRARED DATA ASSOCIATION. IrDA Data Specification. Disponível na internet: http://www.irda.org. Acesso em: 2 de agosto de 2009.

JEDDAH, Ahmed; ZAGUIA, Nejib; JOURDAN, Guy-Vincent. 2009**.** Analyzing the Device Discovery Phase of Bluetooth Scatternet Formation Algorithms. IEEE. 2009.

JOHNSON, David. Hardware and software implications of creating Bluetooth Scattemet devices. IEEE, 2004.

KALETA, Roberta; JOOSTEN, Tanya. Student Response System: A University of Wisconsin System Study of Clickers. *Research Bulletin. Center for Applied Research*. Vol. 2007. Edição 10. 2007.

LINDQUIST, David; et al. Exploring the Potential of Mobile Phones for Active Learning in the Classroom in *SIGCSE '07: Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, Mar 2007.

LOWRY, P.B., ROMANO JR., N.C.; GUTHRIE, R.. Explaining and Predicting Outcomes of Large Classrooms Using Audience Response Systems. Proceedings of the 39th Annual Hawaii International Conference on System Sciences. 2006. 10.1109/HICSS.2006.173.

MARKETT, C.; et al. *Using short message service to encourage interactivity in the classroom*. Computers & Education. Vol. 46, págs. 280 a 293, 2006.

MURILO, Nelson. Aspectos de segurança em redes Wi-Fi. 2006. Disponível na internet: http://www.malima.com.br/article_read.asp?id=287. Acesso em: 9 de abril de 2010.

ORTIZ, Enrique C. Using the Java APIs for Bluetooth Wireless Tecnology. Disponível na internet: http://developers.sun.com/mobility/apis/articles/bluetoothintro/. Acesso em: 1 de agosto de 2009.

PAPYRUS. PAPYRUS COMPUTER TECHNOLOGIES LTD - Infrared (irDA). Disponível na internet: http://www.papyrus.co.il/FAQ/infrared_%28irda%29.htm. Acesso em: 15 de julho de 2009.

PETR, D.W. Experience with a multiple-choice audience response system in an engineering classroom.*Proceedings 35th Annual Conference Frontiers in Education*, 2005. 10.1109/FIE.2005.1612282.

PRINCE, Craig. *Integrating Diverse Student Devices into the Digital Classroom*. General Exam Report (Dept. of Computer Science and Engineering). University of Washington. Jan, 2007

REIS, Terence. Bluetooth Marketing - Visão Geral | MobilePedia. Disponível na internet: http://www.mobilepedia.com.br/artigos/bluetooth-marketing-visao-geral. Acesso em: 10 de março de 2010.

SIAU, Keng; SHENG, Hong; NAH, Fiona. Use of a classroom response system to enhance classroom interactivity. *IEEE JOURNALS.*Vol. 49, 3. 2006.

SIG. Bluetooth.com - The Official Bluetooth Tecnology Info Site. Disponível na internet: http://www.bluetooth.org. Acesso em: 1 de agosto de 2009.

SILVA, Celia.**.** *A timidez em sala de aula: um mal-estar subjetivo no Mundo Contemporâneo.* Dissertação (Mestrado em pedagogia) Fortaleza-CE, UNIFOR, 2000.

SUNKAVALLI, Siva; RAMAMURTHY, Byrav. MTSF: A Fast Mesh Scatternet Formation Algorithm for Bluetooth Networks. IEEE, 2004.

WI-FI ALLIANCE; Wi-Fi Alliance Articles: Wi-Fi Specification. Disponível na internet: http://www.wi-fi.org/knowledge_center_overview.php?type=7. Acesso em: 10 de abril de 2010.

WIKIPEDIA. Bluetooth. Disponível na internet: http://en.wikipedia.org/wiki/Bluetooth. Acesso em: 05 de junho de 2009.

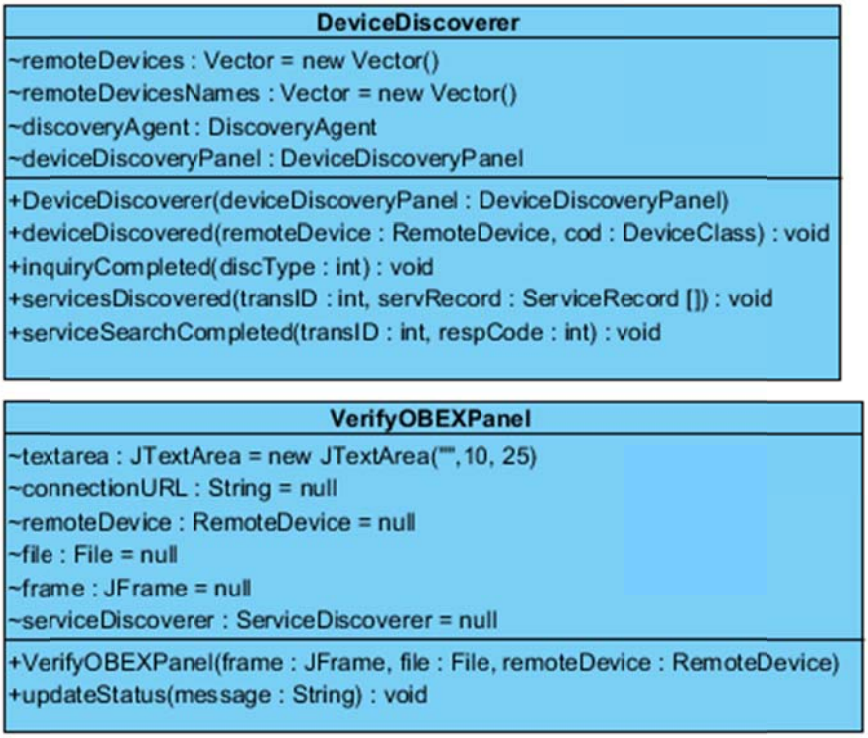
WIKIPEDIA. Infrared Data Association. Disponível na internet: http://en.wikipedia.org/wiki/Infrared_Data_Association. Acesso em: 05 de junho de 2009.

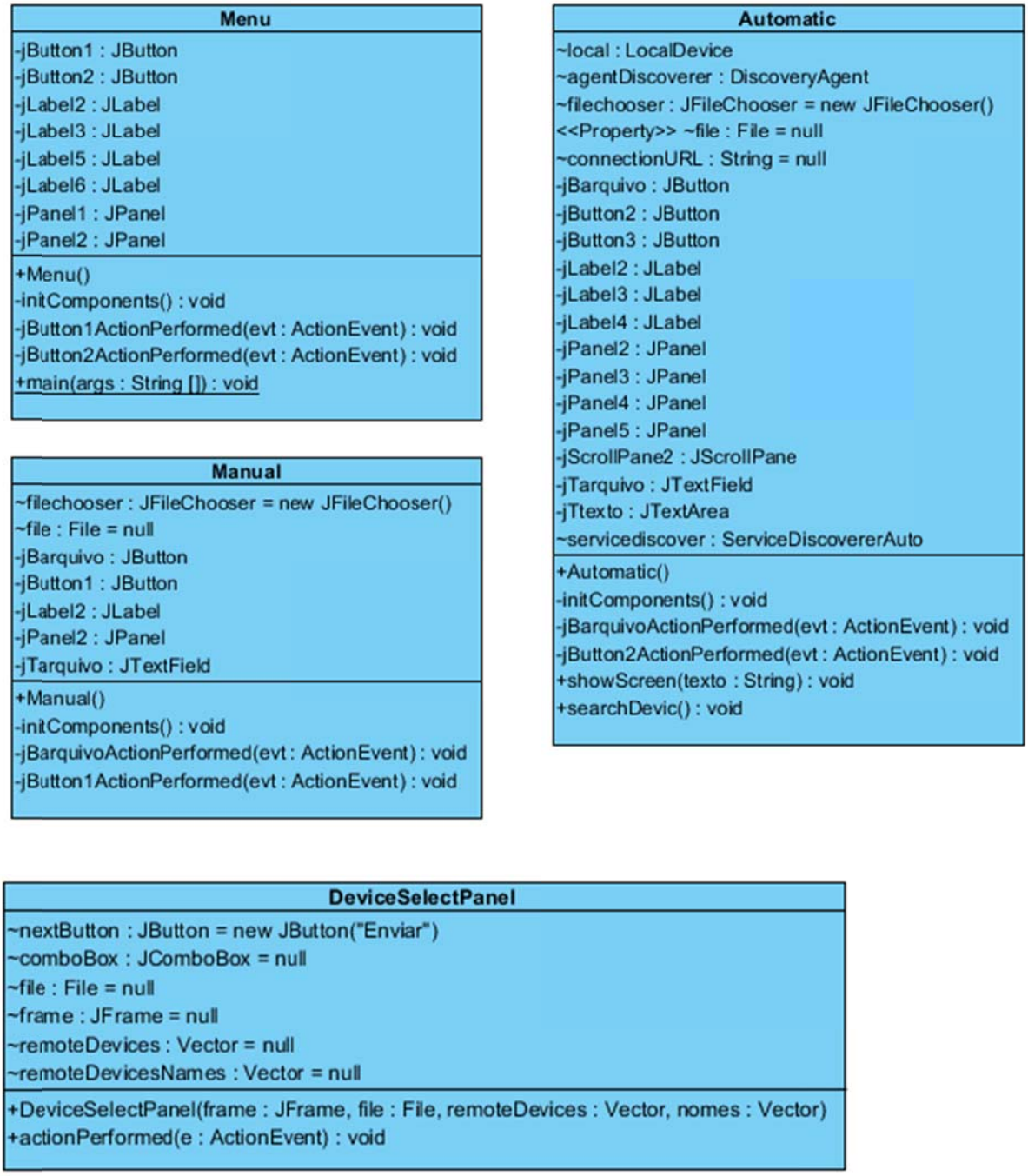
YANG, Chorng-Horng; CHEN, Yi-Sheng. BLUETOOTH SCATTERNET FORMATION FOR SUPPORTING DEVICE MOBILITY. IEEE, 2005.

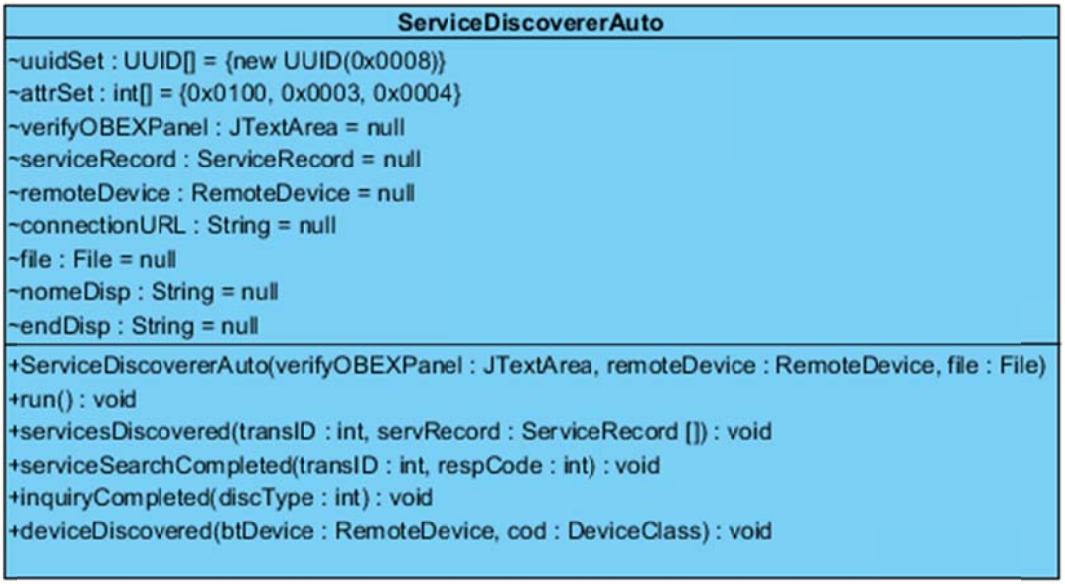
ANEXO I – Diagrama de Classes do Mobile Interact

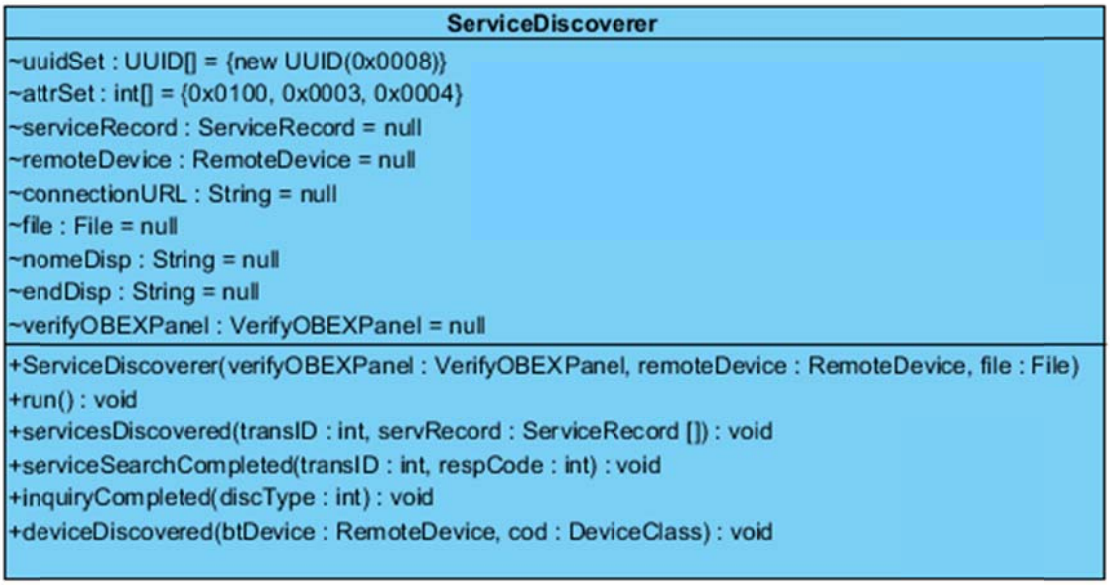
As classes mostradas nos diagramas de classes de cada módulo do sistema Mobile Interact são detalhadas a seguir.

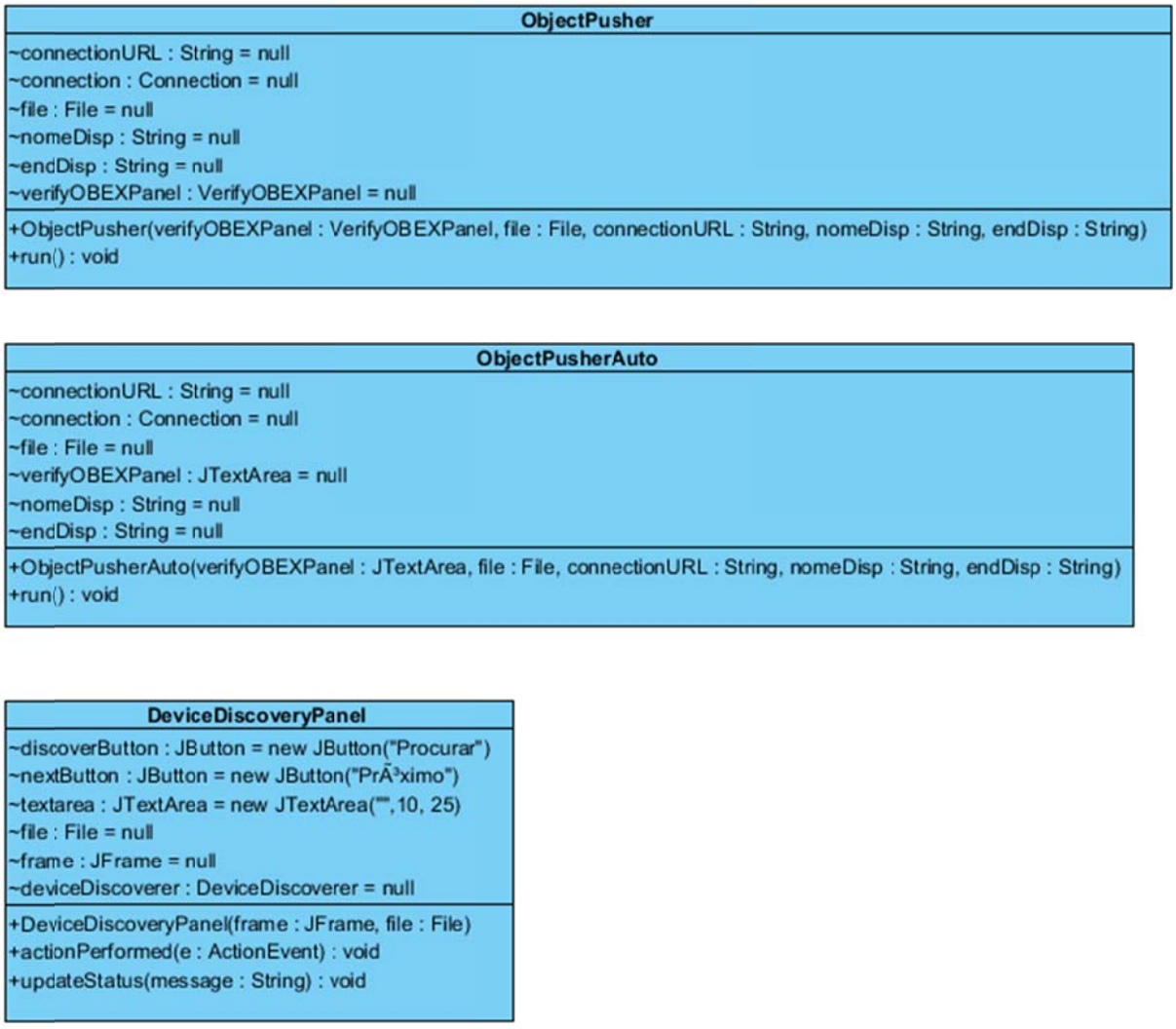
1 MÓDULO BLUETOOTH MARKETING



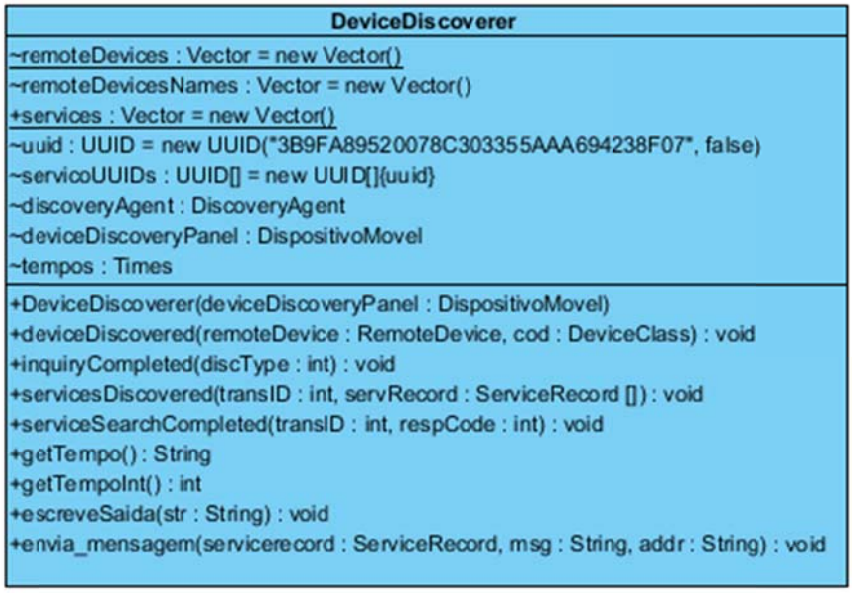

Menu
-jButton1 : JButton
-jButton2 : JButton
-jLabel2 : JLabel
-jLabel3 : JLabel
-jLabel5 : JLabel
-jLabel6 : JLabel
-jPanel1 : JPanel
-jPanel2 : JPanel
+Menu()
-initComponents() : void
-jButton1ActionPerformed(evt : ActionEvent) : void
-jButton2ActionPerformed(evt : ActionEvent) : void
+main(args : String []) : void
Automatic
~local : LocalDevice
~agentDiscoverer : DiscoveryAgent
~filechooser : JFileChooser = new JFileChooser()
<<Property>> ~file : File = null
~connectionURL : String = null
-jBarquivo : JButton
-jButton2 : JButton
-jButton3 : JButton
-jLabel2 : JLabel
-jLabel3 : JLabel
-jLabel4 : JLabel
-jPanel2 : JPanel
-jPanel3 : JPanel
-jPanel4 : JPanel
-jPanel5 : JPanel
-jScrollPane2 : JScrollPane
-jTarquivo : JTextField
-jTtexto : JTextArea
~servicediscover : ServiceDiscovererAuto
+Automatic()
-initComponents() : void
-jBarquivoActionPerformed(evt : ActionEvent) : void
-jButton2ActionPerformed(evt : ActionEvent) : void
+showScreen(texto : String) : void
+searchDevic() : void
Manual
~filechooser : JFileChooser = new JFileChooser()
~file : File = null
-jBarquivo : JButton
-jButton1 : JButton
-jLabel2 : JLabel
-jPanel2 : JPanel
-jTarquivo : JTextField
+Manual()
-initComponents() : void
-jBarquivoActionPerformed(evt : ActionEvent) : void
-jButton1ActionPerformed(evt : ActionEvent) : void
DeviceSelectPanel
~nextButton : JButton = new JButton("Enviar")
~comboBox : JComboBox = null
~file : File = null
~frame : JFrame = null
~remoteDevices : Vector = null
~remoteDevicesNames : Vector = null
+DeviceSelectPanel(frame : JFrame, file : File, remoteDevices : Vector, nomes : Vector)
+actionPerformed(e : ActionEvent) : void

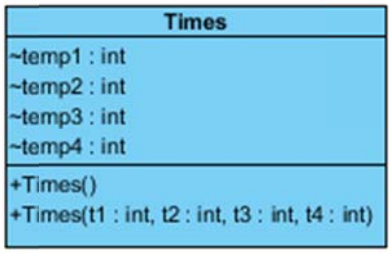

ServiceDiscovererAuto
~uuidSet : UUID[] = {new UUID(0x0008)}
~attrSet : int[] = {0x0100, 0x0003, 0x0004}
~verifyOBEXPanel : JTextArea = null
~serviceRecord : ServiceRecord = null
~remoteDevice : RemoteDevice = null
~connectionURL : String = null
~file : File = null
~nomeDisp : String = null
~endDisp : String = null
+ServiceDiscovererAuto(verifyOBEXPanel : JTextArea, remoteDevice : RemoteDevice, file : File)
+run() : void
+servicesDiscovered(transID : int, servRecord : ServiceRecord []) : void
+serviceSearchCompleted(transID : int, respCode : int) : void
+inquiryCompleted(discType : int) : void
+deviceDiscovered(btDevice : RemoteDevice, cod : DeviceClass) : void


ServiceDiscoverer
~uuidSet : UUID[] = {new UUID(0x0008)}
~attrSet : int[] = {0x0100, 0x0003, 0x0004}
~serviceRecord : ServiceRecord = null
~remoteDevice : RemoteDevice = null
~connectionURL : String = null
~file : File = null
~nomeDisp : String = null
~endDisp : String = null
~verifyOBEXPanel : VerifyOBEXPanel = null
+ServiceDiscoverer(verifyOBEXPanel : VerifyOBEXPanel, remoteDevice : RemoteDevice, file : File)
+run() : void
+servicesDiscovered(transID : int, servRecord : ServiceRecord []) : void
+serviceSearchCompleted(transID : int, respCode : int) : void
+inquiryCompleted(discType : int) : void
+deviceDiscovered(btDevice : RemoteDevice, cod : DeviceClass) : void


ObjectPusher
~connectionURL : String = null
~connection : Connection = null
~file : File = null
~nomeDisp : String = null
~endDisp : String = null
~verifyOBEXPanel : VerifyOBEXPanel = null
+ObjectPusher(verifyOBEXPanel : VerifyOBEXPanel, file : File, connectionURL : String, nomeDisp : String, endDisp : String)
+run() : void
ObjectPusherAuto
~connectionURL : String = null
~connection : Connection = null
~file : File = null
~verifyOBEXPanel : JTextArea = null
~nomeDisp : String = null
~endDisp : String = null
+ObjectPusherAuto(verifyOBEXPanel : JTextArea, file : File, connectionURL : String, nomeDisp : String, endDisp : String)
+run() : void
DeviceDiscoveryPanel
~discoverButton : JButton = new JButton("Procurar")
~nextButton : JButton = new JButton("PrÃ³ximo")
~textarea : JTextArea = new JTextArea("",10, 25)
~file : File = null
~frame : JFrame = null
~deviceDiscoverer : DeviceDiscoverer = null
+DeviceDiscoveryPanel(frame : JFrame, file : File)
+actionPerformed(e : ActionEvent) : void
+updateStatus(message : String) : void

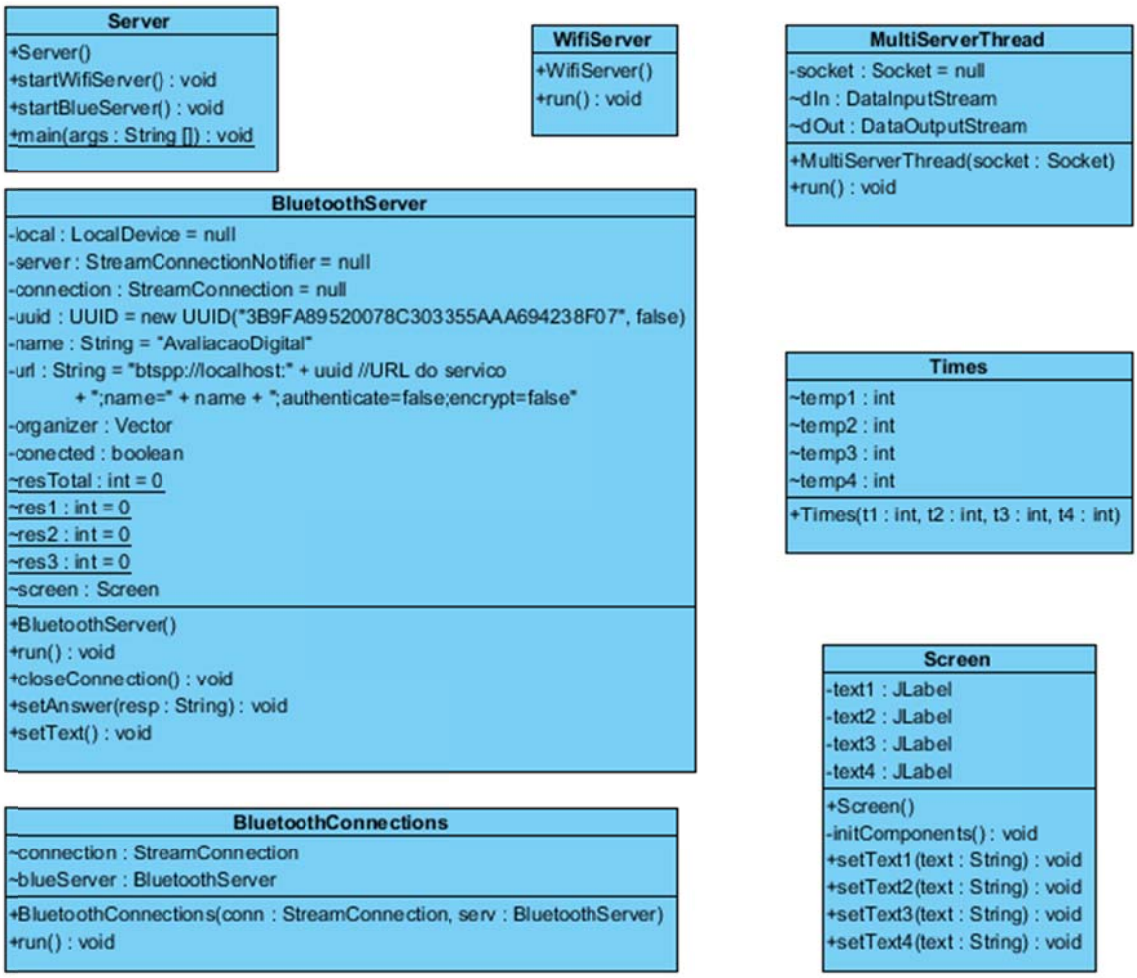
2 MÓDULO DO APLICATIVO PARA CELULAR






DispositivoMovel
-midletPaused : boolean = false
<<Property>> -listMenu : List
<<Property>> -listConectar : List
<<Property>> -formAjuda : Form
<<Property>> -listMenuConectado : List
<<Property>> -alertConexao : Alert
<<Property>> -listMenuConectadoBlue : List
<<Property>> -formReceberAvaliacao : Form
<<Property>> -stringItemAvaliacao : StringItem
<<Property>> -choiceQuestao : ChoiceGroup
<<Property>> -form : Form
<<Property>> -listQuestao : List
<<Property>> -backCommand : Command
<<Property>> -exitCommand : Command
<<Property>> -backCommand1 : Command
<<Property>> -backCommand2 : Command
<<Property>> -backCommand3 : Command
<<Property>> -backCommand4 : Command
<<Property>> -backCommand5 : Command
<<Property>> -okCommand : Command
<<Property>> -backCommand6 : Command
<<Property>> -backCommand7 : Command
<<Property>> -okCommand1 : Command
~socket : SocketConnection
~dIn : DataInputStream
~dOut : DataOutputStream
~quest : String
~op1 : String
~op2 : String
~op3 : String
~op4 : String
~selectQuest : int = 1
~REQUEST_QUESTION : int = 1
~questauto : boolean = true
~resp : boolean = false
~sendQuestWifi : boolean = false
<<Property>> ~inicialTime : long
~discoveryAgente : DiscoveryAgent
~local : LocalDevice
~estadoBluetoothInicial : int
~devic : DeviceDiscoverer
+DispositivoMovel()
-initialize() : void
+startMIDlet() : void
+resumeMIDlet() : void
+switchDisplayable(alert : Alert, nextDisplayable : Displayable) : void
+commandAction(command : Command, displayable : Displayable) : void
+listMenuAction() : void
+listConectarAction() : void
+listMenuConectadoAction() : void
+listMenuConectadoBlueAction() : void
+listQuestaoAction() : void
+getTempo() : String
+getTempoInt() : int
+getDisplay() : Display
+exitMIDlet() : void
+startApp() : void
+pauseApp() : void
+destroyApp(unconditional : boolean) : void
+receiveWifi(numQuestao : int) : void
+showQuest() : void
+showScreen(texto : String) : void
+getQuestAuto() : boolean
+getResp() : boolean
+showQuest(quest : String, op1 : String, op2 : String, op3 : String, op4 : String) : void
+showQuest3(quest : String, op1 : String, op2 : String, op3 : String) : void
+getAnswer() : String

3 MÓDULO DO SERVIDOR BLUETOOTH E WI-FI



Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Administração
Baixar livros de Agronomia
Baixar livros de Arquitetura
Baixar livros de Artes
Baixar livros de Astronomia
Baixar livros de Biologia Geral
Baixar livros de Ciência da Computação
Baixar livros de Ciência da Informação
Baixar livros de Ciência Política
Baixar livros de Ciências da Saúde
Baixar livros de Comunicação
Baixar livros do Conselho Nacional de Educação - CNE
Baixar livros de Defesa civil
Baixar livros de Direito
Baixar livros de Direitos humanos
Baixar livros de Economia
Baixar livros de Economia Doméstica
Baixar livros de Educação
Baixar livros de Educação - Trânsito
Baixar livros de Educação Física
Baixar livros de Engenharia Aeroespacial
Baixar livros de Farmácia
Baixar livros de Filosofia
Baixar livros de Física
Baixar livros de Geociências
Baixar livros de Geografia
Baixar livros de História
Baixar livros de Línguas

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo