



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE QUIXADÁ**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**

**THALITA MARIA FIRMINO DE MOURA**

**DESENVOLVIMENTO DE UMA LOUSA INTERATIVA MÓVEL PARA USO EM  
AULAS PRESENCIAIS**

**QUIXADÁ**  
**2013**

**THALITA MARIA FIRMINO DE MOURA**

**DESENVOLVIMENTO DE UMA LOUSA INTERATIVA MÓVEL PARA USO EM  
AULAS PRESENCIAIS**

Trabalho de Conclusão de Curso submetido à  
Coordenação do Curso Bacharelado em  
Sistemas de Informação da Universidade  
Federal do Ceará como requisito parcial para  
obtenção do grau de Bacharel.

Área de Concentração: Computação.

Orientador: Prof. Msc. Ricardo Reis Pereira.

**QUIXADÁ**

**2013**

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do Campus de Quixadá

---

M884d Moura, Thalita Maria Firmino de  
Desenvolvimento de uma lousa interativa móvel para uso em aulas presenciais / Thalita Maria  
Firmino de Moura. – 2013.  
57 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de  
Sistemas de Informação, Quixadá, 2013.  
Orientação: Prof. Me. Ricardo Reis Pereira  
Área de concentração: Computação

1. Software de aplicação - desenvolvimento 2. Computação móvel 3. Software - desenvolvimento I.  
Título.

**THALITA MARIA FIRMINO DE MOURA**

**DESENVOLVIMENTO DE UMA LOUSA INTERATIVA MÓVEL PARA USO EM  
AULAS PRESENCIAIS**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: Computação.

Aprovado em: 08/01/2014.

**BANCA EXAMINADORA**

---

Prof. Msc. Ricardo Reis Pereira (Orientador)  
Universidade Federal do Ceará – UFC

---

Prof. Dr. Críston Pereira de Souza  
Universidade Federal do Ceará – UFC

---

Prof. Dr. David Sena Oliveira  
Universidade Federal do Ceará – UFC

Dedico este trabalho primeiramente a Deus, pela saúde, fé e perseverança que tem me dado. E aos meus queridos pais, que com luta, mas principalmente com muita dedicação e amor, me deram a educação sem a qual eu não teria chegado a lugar algum.

## **AGRADECIMENTOS**

A Deus, quando algumas vezes, sentindo-me desacreditada e perdida nos meus objetivos, me concedeu forças para continuar lutando pelos meus sonhos.

A toda minha família, em especial: à minha mãe, Maria Antonete, minha avó, Francisca Ceci, e minha tia, Maria Antoneide, pelo amor, força e apoio em todos os momentos.

Ao meu orientador, Prof. Ricardo Reis Pereira, pela ajuda e dedicação para que esse trabalho fosse concluído com sucesso. Aos Professores, Críston Pereira e Davi Sena, por terem aceitado o convite de compor a banca.

Ao meu grande amigo José Diógenes, pela paciência, pelo carinho e pelos conhecimentos compartilhados que foram de grande importância para a minha jornada acadêmica, nesses quatro anos.

Aos demais colegas da Universidade que participaram, de alguma forma, na construção e realização deste tão desejado sonho de carregar o canudo de minha formatura.

A todos vocês, muito obrigada!

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo.  
Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas  
admiráveis.”

(José de Alencar)

## RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação para dispositivos móveis que simula uma lousa móvel. A aplicação foi dividida em dois módulos, um módulo cliente e outro servidor. Ambos foram implementados utilizando o *framework Qt*. O aplicativo cliente permite desenho à mão livre e, uma vez conectado ao aplicativo servidor, compartilha seu conteúdo com outros clientes conectados. O aplicativo servidor assegura estas conexões e redireciona dados entre os clientes, possibilitando que compartilhem uma experiência de página de desenho compartilhada. Além do desenvolvimento da lousa móvel, este trabalho apresenta um estudo sobre a evolução da comunicação móvel, mostrando o potencial dessa área.

Palavras chave: Dispositivos móveis. *Framework Qt*. Lousa móvel.



## **ABSTRACT**

This work presents the development of a mobile application which simulates a mobile whiteboard. The application is divided into two modules, client and server. Both were implemented using the Qt framework. The client application allows freehand drawing and if connected to the server application, shares its content with other connected clients. The application server maintains these connections and exchange data between clients creating a sharing drawing page experience. Besides the development of mobile whiteboard, this work presents a study on the evolution of mobile communication, showing the potential of this area.

**Keywords:** Mobile Devices. Qt Framework. Mobile Whiteboard.

## LISTA DE FIGURAS

FIGURA 1 – AMBIENTE DE UTILIZAÇÃO DA LOUSA MÓVEL.....	27
FIGURA 2 – PONTO ABSOLUTO.	FIGURA 3 – PONTO NORMALIZADO..... 29
FIGURA 4 – PROCESSO DE CONSTRUÇÃO DE TRAÇO CORRENTE. ....	30
FIGURA 5 – PROCESSO DE CONSTRUÇÃO DE TRAÇOS REMOTOS.....	31
FIGURA 6 – MODELO DE INTERAÇÃO SIMULTÂNEA. ....	37
FIGURA 7 – MODELO DE INTERAÇÃO SIMULTÂNEA COM <i>THREAD</i> .....	38

## **LISTA DE FLUXOGRAMAS**

FLUXOGRAMA 1 – RELACIONAMENTO ENTRE DOIS CLIENTES (EMISSOR E RECEPTOR).....	34
FLUXOGRAMA 2 – RELACIONAMENTO CONCORRENTE ENTRE DOIS OU MAIS CLIENTES .....	35

## LISTA DE QUADROS

QUADRO 1 – BLOCO.....	32
QUADRO 2 – CASO DE TESTE 1.....	36
QUADRO 3 – CASO DE TESTE 2.....	37
QUADRO 4 – CASO DE TESTE 3.....	39

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>14</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b> .....	<b>17</b>
2.1	COMUNICAÇÃO MÓVEL.....	17
2.1.1	<i>Histórico da Comunicação Móvel</i> .....	17
2.1.2	<i>Dispositivos Móveis</i> .....	18
2.1.2.1	Vantagens dos Dispositivos Móveis .....	19
2.1.2.2	Desafio dos Dispositivos Móveis .....	20
2.2	LOUSA DIGITAL .....	22
2.2.1	<i>Contextualização</i> .....	22
2.2.2	<i>Informações Técnicas da Lousa Digital</i> .....	22
2.3	O FRAMEWORK QT.....	23
2.3.1	<i>Definição</i> .....	23
2.3.2	<i>Uma breve história do Qt</i> .....	24
<b>3</b>	<b>PROCEDIMENTOS METODOLÓGICOS</b> .....	<b>26</b>
<b>4</b>	<b>DESENVOLVIMENTO DA APLICAÇÃO</b> .....	<b>27</b>
4.1	AMBIENTE DE UTILIZAÇÃO .....	27
4.2	CONCEITOS BÁSICOS .....	27
4.3	NORMALIZAÇÃO E DESNORMALIZAÇÃO DE PONTOS .....	28
4.4	GERENCIAMENTO DE DADOS NO CLIENTE .....	29
4.5	MODELO DE ENVIO DE DADOS.....	31
4.6	PROCESSAMENTO DE DADOS PROVENIENTES DA REDE NO CLIENTE .....	32
4.7	FUNÇÃO DA APLICAÇÃO SERVIDOR .....	33
4.8	FLUXOGRAMAS DA APLICAÇÃO .....	33
<b>5</b>	<b>TESTE DA APLICAÇÃO</b> .....	<b>36</b>
5.1	CASO DE TESTE – COMUNICAÇÃO ENTRE DOIS CLIENTES COM INTERAÇÃO RESTRITA ...	36
5.2	CASO DE TESTE – COMUNICAÇÃO ENTRE DOIS CLIENTES COM INTERAÇÃO SIMULTÂNEA	36
5.3	CASO DE TESTE – COMUNICAÇÃO ENTRE QUATRO CLIENTES COM INTERAÇÃO	
	SIMULTÂNEA USANDO <i>THREADS</i> .....	38
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>40</b>
6.1	CONCLUSÃO.....	40
6.2	TRABALHOS FUTUROS .....	40
	<b>REFERÊNCIAS</b> .....	<b>42</b>

## 1 INTRODUÇÃO

Nos últimos anos, o mercado de dispositivos móveis vem crescendo de maneira acelerada. Desta forma, as novidades tecnológicas estão progredindo rapidamente e o futuro da computação móvel está se tornando cada vez mais promissor. De acordo com o site Idgnow (2013), o número de *smartphones*, *tablets*, *laptops* e telefones com conexão a internet será maior do que o número de pessoas no mundo até o final de 2013, ou seja, a quantidade de dispositivos móveis irá ultrapassar os 7 bilhões e, até 2017, existirão aproximadamente 1,4 dispositivos por habitante.

*Smartphones e tablets* chegaram ao mercado com uma proposta tímida, desprovida de grandes ambições (VIRGÍLIO, 2013). Entretanto, apesar do poder computacional limitado, a cada geração ocorrem melhorias na portabilidade e aumento na capacidade de processamento e armazenamento. Paralelamente, o aparecimento de modelos com preços mais acessíveis facilitou a popularização desses dispositivos em diversas camadas da sociedade. Visto isso, é notório o crescente número de consumidores que buscam por mais recursos e novas funcionalidades, como telas maiores e com novos formatos e sensibilidade ao toque, diferentes recursos multimídias, conectividade, jogos e aplicações específicas. Portando um dispositivo, um usuário pode utilizar diversas aplicações, seja fazendo anotações em um editor de textos, checando suas tarefas no calendário ou acessando um mapa para descobrir sua localização.

A expansão desse mercado impulsiona o desenvolvimento de novas aplicações. O sistema operacional *Android*, presente em grande número de dispositivos móveis, constitui um exemplo de plataforma que permite o desenvolvimento de aplicações livres e de código aberto. De acordo com Schemberger e Freitas (2009), para operadoras de telefonia, fabricantes de dispositivos móveis e usuários avançados, a possibilidade de desenvolvimento é bastante vantajosa, haja vista existirem sistemas operacionais gratuitos e de fácil personalização.

Para oferecer uma experiência de uso ainda mais rica e interessante ao usuário, as aplicações para dispositivos móveis podem ser interativas. A interatividade permite que duas ou mais pessoas possam exercer algum tipo de influência, através de comportamentos e ações, sobre o conteúdo armazenado em dispositivos individuais ou em uma rede compartilhada. Desta forma, o processo de comunicação e compartilhamento de conteúdos torna-se mais dinâmico e prático. É neste contexto que entra a lousa interativa digital. Para Junior (2009), a lousa interativa (ou digital) permite que se integrem em um único instrumento todas as

técnicas de comunicação e de processamento da informação conhecidas. Tais características são primordiais para a concepção e proposição de aulas mais dinâmicas, ilustradas e, principalmente, produtivas.

Baseando-se nessas considerações, o objetivo principal deste trabalho é desenvolver uma aplicação que simula uma lousa interativa para uso em sala de aula, executando-a em dispositivos móveis, visando potencializar a dinâmica de ensino e tornar o aprendizado mais participativo e significativo. Para isso, foi necessário (i) identificar os módulos da tecnologia Qt com ênfase em dispositivos móveis, (ii) construir um modelo simplificado da aplicação lousa interativa móvel e (iii) estender e testar a aplicação. Diferentemente das lousas interativas convencionais, onde a tecnologia é concentrada em uma tela grande e de difícil locomoção, por serem pesadas ou por serem fixas, a lousa móvel tem tecnologia concentrada no software, tornando-se mais flexível e de baixo-custo.

Em Junior (2009), foi desenvolvida uma lousa interativa baseada na análise de imagem por intermédio de um software que interpreta e atende às intenções do usuário. A lousa funciona a partir do uso de um projetor convencional que exibe uma imagem em uma superfície qualquer. A intenção do usuário é reconhecida através de uma caneta eletrônica emissora de infravermelho que funciona como um *mouse*, e de uma câmera infravermelho que recebe esse sinal.

Neste trabalho, a lousa proposta foi desenvolvida utilizando o *framework* Qt, visto que o mesmo oferece os recursos necessários para o desenvolvimento de aplicações gráficas para dispositivos móveis. A aplicação está dividida em um módulo cliente e um módulo servidor. O módulo servidor é responsável por atender as solicitações de conexão dos clientes (alunos e professor) e retransmitir os dados entre eles. No módulo cliente, o usuário, através de seu dispositivo móvel, produz conteúdo que é então compartilhado/visualizado por outros clientes em seus dispositivos. O processo comunicativo, o ciclo emissor–mensagem–receptor, tem o diferencial de possibilitar que o receptor torne-se também um produtor de mensagens, ou seja, tanto um professor pode produzir conteúdo e este ser visualizado pelos alunos, quanto um aluno pode produzir conteúdo e este ser visualizado pelo professor e outros alunos, aumentando assim o poder de interação durante a aula.

Além da introdução, este trabalho está estruturado da seguinte maneira: a seção 2 apresenta a revisão bibliográfica, mostrando o histórico da comunicação móvel e as vantagens e desafios dos dispositivos móveis. O conceito de lousa digital e as informações técnicas da mesma também são mostrados na seção 2. Ainda nesta mesma seção, é mostrado uma breve história e conceitos do *framework* Qt. Na seção 3, são mostrados os procedimentos

metodológicos utilizados para a execução deste trabalho e o cronograma contendo todas as atividades do projeto. Na seção 4, é mostrado o processo de desenvolvimento da aplicação, focando nos resultados obtidos. Na seção 5, são mostrados os testes realizados na aplicação. Por fim, na seção 6 são apresentadas as considerações finais acerca do trabalho desenvolvido e os trabalhos futuros.



## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Comunicação Móvel

#### 2.1.1 *Histórico da Comunicação Móvel*

De acordo com Dias e Sadok (2001), um sistema de comunicação móvel tem como característica a possibilidade de movimento relativo entre as partes como, por exemplo, a comunicação entre o telefone celular e a estação base na telefonia celular. Sistemas móveis usam a tecnologia sem fio para possibilitar uma comunicação transparente enquanto o usuário se desloca.

Ainda de acordo com Dias e Sadok (2001), o desejo da humanidade em comunicar-se livre de fios ocorre desde os primórdios da civilização. Na Grécia antiga o uso de sinais de fumaça é mencionado como forma de comunicação. No final do século XVIII, Claude Chape inventa a telegrafia óptica, possibilitando a comunicação sem fio para longas distâncias. Em 1820, Hans Christian Oersted descobre experimentalmente que a corrente elétrica produz um campo magnético. As comunicações sem fio tiveram uma contribuição fundamental com Michel Faraday demonstrando a indução eletromagnética em 1831. Em 1864, James C. Maxwell lança os fundamentos teóricos sobre campos magnéticos com suas famosas equações. Em 1876, Alexander Graham Bell inventa o telefone. Finalmente, Heinrich Hertz foi o primeiro a demonstrar, através de um experimento em 1887, as equações de Maxwell sobre ondas eletromagnéticas.

Em 1905, ocorreu a primeira transmissão de voz e música em um canal sem fio por Reginald Fessenden, e em 1946, o primeiro sistema de telefonia móvel comercial, operado pela empresa americana Bell, foi implantado em St. Louis, Missouri, nos Estados Unidos.

A era da telefonia celular teve seu início efetivo no início dos anos 90, quando o usuário podia portar o aparelho embora suas dimensões iniciais fossem grandes. Originalmente, telefones celulares surgiram como dispositivos para conversação por voz, exclusivamente. Porém, com o avanço da tecnologia e a evolução das gerações da telefonia celular, esses dispositivos adquiriram também capacidade de processamento e comunicação através da integração da rede celular com rede de dados, em especial a internet. Foram adotados vários padrões em diferentes países para a telefonia celular e ficaram conhecidos como a Primeira Geração (1G). Segundo Dias e Sadok (2001), os sistemas de 1G utilizam a transmissão de dados de modo analógicos e a técnica utilizada de acesso ao meio é a FDMA

(*Frequency Division Multiple Access* – Acesso Múltiplo por Divisão de Frequência), que possuía uma baixa qualidade e uma incompatibilidade com os diversos sistemas existentes.

Depois da 1G ocorreu o surgimento da Segunda Geração (2G), com o objetivo de aumentar a capacidade da primeira geração. A 2G utilizava o TDMA (*Time Division Multiple Access* TDMA – Acesso múltiplo por divisão de tempo) e o CDMA (*Code Division Multiple Access* – Acesso múltiplo por divisão de código). Na Europa surgiu o GSM (*Global System for Mobile Communications*). Uma inovação encontrada no sistema GSM é a utilização do SIM (*Subscriber Identification Module*) que contém algumas identificações do usuário assim como chave de código de privacidade. O SIM é conectado a um terminal GSM, pode ser removido de um aparelho e conectado em outro. Sem o SIM o terminal fica impossibilitado de operar.

Posteriormente, surgiu a Terceira Geração (3G), com o objetivo de disponibilizar telefonia móvel de longo alcance, possibilitando o suporte a aplicações multimídia e acesso à internet. Atualmente, a 4G já se encontra em uso e até 2020 espera-se a disponibilização dos serviços da Quinta Geração (5G). Segundo o Site Info (2013), esta nova tecnologia para dispositivos móveis permitirá aos usuários alcançar velocidades pela internet que possibilitarão baixar, por exemplo, um filme completo em menos de um segundo.

As facilidades trazidas com a mobilidade e o poder computacional crescente dos dispositivos fizeram com que o grau de dependência por estes dispositivos aumentassem, e fizessem com o que os celulares se tornassem parte essencial no dia-a-dia de muitas pessoas, seja como ferramenta de trabalho e contato com clientes ou como ferramenta de comunicação instantânea.

### **2.1.2 Dispositivos Móveis**

Dos anos 90 para cá, pudemos notar um grande crescimento no desenvolvimento de tecnologias para comunicação celular móvel, comunicação via satélite e redes locais sem fio. A popularização dessas tecnologias tem permitido o acesso a informações remotas onde quer que se esteja, abrindo um leque muito grande de facilidades, aplicações e serviços para os usuários.

Muito mais do que assistentes pessoais ou agendas eletrônicas, os dispositivos móveis passaram a ser computadores que podem ser facilmente levados a qualquer lugar, criados para atender profissionais e pessoas em movimento que necessitam de rapidez, facilidade e segurança no acesso a informações corporativas e pessoais. De acordo com Figueiredo e

Nakamura (2003), um dispositivo móvel deve ter a capacidade de realizar processamento, trocar informações via rede e ser capaz de ser transportado facilmente por seu usuário. Para isso, é importante que o dispositivo computacional tenha tamanho reduzido e não necessite de cabos para conectá-lo à rede de dados ou fonte de energia elétrica. Assim, equipamentos deste tipo devem ter as seguintes características: ser bem menor que as estações de trabalho que costumamos usar, geralmente manipulados no colo ou na palma das mãos; possuir uma bateria, para evitar a necessidade de conexões à rede elétrica através de cabos que limitariam muito a mobilidade; e ter acesso a dados através de tecnologias de redes sem fio.

Os celulares ainda possuem como principal função o serviço de voz, porém a transmissão de dados vem tornando cada vez mais frequente. Como exemplo, pode-se citar a possibilidade de ter acesso à informação e serviços a qualquer hora acessando a internet. Paralelamente à evolução dos celulares, surgiram outros dispositivos móveis com funcionalidades específicas, como os PDAs (*Personal Digital Assistants* - Assistentes Digitais Pessoais). A melhoria nas técnicas utilizadas na fabricação destes dispositivos e o preço dos componentes de hardware em queda possibilitaram o surgimento de dispositivos como *smartphones* que reúnem as principais funcionalidades de celulares e PDAs.

À medida que a demanda por funcionalidades aumenta, as empresas fabricantes de *smartphones* vêm acrescentando novas tecnologias a tais aparelhos, variando seus tamanhos, fazendo modificações desde hardware até software, estimulando nos consumidores o desejo de possuir o mais recente modelo do aparelho.

#### **2.1.2.1 Vantagens dos Dispositivos Móveis**

Permitir que usuários acessem informações contextualmente relevantes a qualquer momento e em qualquer lugar é uma grande vantagem dos dispositivos móveis. As pessoas estão cada vez mais dependentes das informações disponibilizadas na internet, o que antes poderia ser feito apenas em uma infraestrutura fixa, agora pode ser acessado via dispositivo móvel. Elas estão mudando a maneira pela qual acessam a rede mundial, ou seja, não apenas de seus computadores, mas também de dispositivos móveis.

A mobilidade é outra característica que deve ser levada em consideração. A capacidade de poder continuar uma comunicação e manter o envio de dados constante mesmo quando em movimento pode ser considerada uma das melhores vantagens de um dispositivo móvel.

Algumas vantagens dos dispositivos móveis em relação aos computadores são listadas a seguir:

- Tamanho: bastante reduzidos e muito mais leves do que os computadores comuns, podendo ser transportados de forma muito mais prática;
- Consumo de energia: por serem menores e mais econômicos gastam menos energia que os computadores visto que o tempo de recarga é menor;
- Fácil manuseio: os dispositivos móveis possuem uma interface gráfica simples de manusear se comparado aos computadores;
- Custos operacionais: como os dispositivos móveis são mais compactos e possuem atividades específicas, estes aparelhos não possuem alguns periféricos internos, como discos rígidos e discos flexíveis, diminuindo consideravelmente os custos com a manutenção.

Sob o ponto de vista empresarial, os dispositivos também não deixam a desejar. Segundo Schaefer (2004), para aqueles que consomem grande parte do seu tempo trabalhando remotamente, estes equipamentos são versáteis, dedicados, multifuncionais e de uso genérico. Do ponto de vista empresarial, eles são ótimos geradores de informações, podendo ser utilizados desde a automação de processos até a coleta de informações estratégicas.

### **2.1.2.2 Desafio dos Dispositivos Móveis**

O mercado de dispositivos móveis é dominado por diferentes plataformas tecnológicas, incluindo sistemas operacionais e plataformas de desenvolvimento de software, o que gera uma variedade de diferentes soluções disponíveis no mercado. Sendo assim, um dos principais desafios da computação móvel é o desenvolvimento de software em um ambiente de tecnologia altamente fragmentado e em rápida evolução. Desta forma, boa parte das principais dificuldades está relacionada às próprias plataformas móveis: como não há uniformidade no mundo *mobile*, um aplicativo está sujeito a ser executado em dispositivos e sistemas operacionais bem diferentes uns dos outros. Outras dificuldades podem surgir, por exemplo, quando se decide desenvolver para várias plataformas simultaneamente. Nem toda equipe é fluente em várias linguagens e tecnologias.

É necessário criar uma experiência para o usuário que torne agradável o uso do aplicativo. Fornecer uma boa experiência para o usuário significa utilizar os recursos de uma plataforma ou dispositivo para entregar ao usuário a informação da maneira mais eficiente e intuitiva possível. Isto porque cada plataforma tem seus recursos específicos, principalmente

em relação às interfaces gráficas, e interfaces aparentemente idênticas em duas plataformas podem ter comportamentos diferentes por terem formas diferentes de interagir com o usuário. Desta forma, independente da tecnologia utilizada, quando se fala em dispositivos móveis o foco deve estar sempre na experiência que o aplicativo possibilita ao usuário. Sem isto, outros requisitos importantes podem não ser suficientes para garantir o sucesso do mundo mobile.

Outros problemas que podem ser destacados são os relacionados aos desafios no campo sociológico. Os usuários costumam a trocar o papel por uma tela, por mais fina que seja. O fato mais importante a ser observado é a maneira como se usa um celular ou dispositivo portátil, além das limitações da tela e bateria, ele não é usado da mesma forma de como se utiliza um computador, mesmo que consigamos atingir seu mesmo desempenho tecnológico.

Além dos desafios no campo sociológico, há desafios no campo técnico, como destacados a seguir:

- Baixo poder de processamento: os dispositivos móveis têm reserva de energia limitada, para garantir uma autonomia de uso razoável, a saída é cortar o poder de processamento da CPU (*Central Processing Unit* - Unidade Central de Processamento). Isto faz com que não seja prático realizar certos processamentos mais pesados nos dispositivos móveis;
- Interoperabilidade: a comunicação de sistemas móveis com sistemas já existentes nas empresas é um grande desafio, principalmente porque tais sistemas não foram construídos considerando este tipo de interação.
- Recurso de memória limitado: os dispositivos possuem pouca memória, exigindo assim que o gerenciamento de memória seja fundamental;
- Interface reduzida: normalmente esses dispositivos possuem uma tela de pequena dimensão, tornando limitada a quantidade de informações que pode ser visualizada, sem disponibilidade de janelas.

Atualmente, grandes fabricantes de dispositivos móveis fazem investimentos acentuados na área *mobile*, com o objetivo de fazer com que tenhamos mais recurso tecnológico nos dispositivos móveis, embora ainda não se faça uso de todos esses recursos. Os desafios estão sendo superados rapidamente, pois a transmissão de dados por dispositivos móveis está se tornando uma nova frente de negócios, onde diversos setores da economia buscam cada vez mais uma forma de diminuir tempo e custos e melhorar a qualidade de seus serviços.

## **2.2 Lousa Digital**

### **2.2.1 Contextualização**

A lousa digital é uma ferramenta de ensino destinada a facilitar a exposição de conhecimentos em salas de aula, conferências e reuniões corporativas. Tal dispositivo confere uma ergonomia de interfaceamento com o computador adequada à estes ambientes, elevando a interação dos participantes, em geral professores e alunos, com o conteúdo ministrado (MOREIRA, 2009).

Em vez de giz, uma caneta eletrônica, no lugar do quadro-negro, uma plataforma sensível ao toque. A lousa digital é um recurso que veio para revolucionar as salas de aulas: uma grande tela que permite que os alunos possam visualizar o mesmo conteúdo, havendo interação com o recurso de tela sensível ao toque, permitindo postar documentos na internet, compartilhar arquivos na rede local ou enviar informações por e-mail. Nesse sentido, de acordo com Nakashima e Amaral (2006), ao utilizar a lousa digital o professor pode acessar páginas na internet, escrever, desenhar, editar, gravar e enviar para os seus alunos via *e-mail* tudo o que foi escrito e realizado no quadro durante as aulas.

A substituição do quadro negro por uma plataforma sensível ao toque conferiu mais dinamismo e interação aos usuários. Para Moreira (2009), a característica amigável da lousa digital confere satisfação ao usuário, pois pode, por exemplo, ser utilizada para proporcionar experiências lúdicas em sala de aula alternativamente à ferramenta tradicional, estática e monótona.

Apesar da diversidade de recursos oferecidos pelas lousas digitais, que proporcionam um ambiente de aprendizagem mais inovador e dinâmico, estas ainda assim possuem algumas desvantagens. Para Júnior (2009), as lousas interativas convencionais não oferecem ao usuário flexibilidade, pois sua tecnologia é concentrada na tela especial, de difícil locomoção, já que são pesadas e, muitas vezes, fixas; além disso, necessitam de cabos para alimentação.

### **2.2.2 Informações Técnicas da Lousa Digital**

De acordo com Nakashima e Amaral (2006), atualmente existem vários modelos de lousas digitais, variando o tamanho, a marca e o custo, mas a maioria é composta por uma tela conectada a um computador e um projetor multimídia. A superfície dessa tela é sensível ao toque, isto é, quando alguém executa algum movimento sobre ela, o computador registra o que se fez em um software específico que acompanha a lousa digital.

Os modelos mais atuais têm a CPU acoplados na parte inferior, com tamanho bem discreto. É neste local que ficam os dispositivos que permitem o funcionamento da lousa digital, tal qual um computador normal (processador, memória e disco rígido ficam ali alojados). Existem modelos que possuem entradas USB (*Universal Serial Bus*) neste mesmo local, e outros modelos que possuem entradas laterais (na própria lousa digital) de conexões USB para transferência de arquivos com unidades removíveis compatíveis (como um pen drive, por exemplo). Desta forma, tudo o que se pensar em termos de recursos de um computador, de multimídia, simulação de imagens e navegação na internet é possível com a lousa. Ou seja, funciona como um computador, mas com uma tela melhor e maior.

Ainda de acordo com Nakashima e Amaral (2006), Há também a opção de utilizar acessórios, como canetas específicas que possuem uma ponta de borracha, juntamente com um apagador especial que não danifica a superfície do quadro. As canetas ficam em um suporte integrado à lousa e cada uma delas possui uma “tinta eletrônica”. Ao retirar uma delas do suporte há sensores ópticos que detectam automaticamente a escolha da cor da caneta e os traços que forem feitos na lousa serão da cor correspondente. Para que os traços saiam da forma desejada é necessário ter firmeza no traçado, tendo o cuidado para não apoiar a mão no quadro, evitando que essas marcas indesejadas fiquem registradas. Para apagar aquilo que foi escrito na lousa, basta retirar o apagador do suporte e movimentá-lo suavemente sobre as anotações, que automaticamente serão apagadas.

O software de gerenciamento da lousa digital fornece algumas ferramentas que ajudam na manipulação da mesma. Uma delas é a lente de aumento (*zoom*), que permite ampliar o tamanho do conteúdo ou da imagem selecionada, a fim de verificar seus detalhes. A ferramenta *holofote* permite focar uma determinada área da lousa (permite que toda tela fique escura), deixando o que se quer mostrar em destaque. Outro recurso interessante é o teclado digital que exclui a necessidade de utilizar o teclado convencional, permitindo digitar qualquer caractere diretamente no quadro interativo.

## 2.3 O Framework Qt

### 2.3.1 Definição

Qt é um *framework* de desenvolvimento de software completo com ferramentas projetadas para simplificar a criação de aplicações com interfaces de usuário de alta interatividade para desktops, sistemas embarcados e plataformas móveis (QT DIGIA, 2013).

O Qt permite desenvolver aplicações multiplataforma, ou seja, foi projetado para desenvolver aplicações em vários sistemas operacionais, como Windows, Linux, Mac OS X e

*Android*. Desta forma, o código-fonte de aplicações desenvolvidas em Qt é implementado uma única vez e pode ser compilado em várias plataformas sem a necessidade de alteração.

O *framework* Qt está disponível em duas licenças: uma licença para softwares comerciais, na qual os desenvolvedores pagam taxas de licenciamento para utilizar as ferramentas oferecidas pelo Qt, e uma edição livre, na qual as versões possuem código-fonte aberto, podendo ser utilizadas por qualquer desenvolvedor sem cobrança de taxas (BLANCHETTE; SUMMERFIELD, 2006).

A estrutura do *framework* Qt, atualmente, é mantida em módulos. Os módulos utilizados para este trabalho foram: Core, Gui, Network e Multithreading. Estes são detalhados mais adiante.

### 2.3.2 *Uma breve história do Qt*

O *framework* Qt tornou-se disponível ao público em maio de 1995. Foi inicialmente desenvolvido por Haavard Nord (CEO da Trolltech) e Eirik Chambe-Eng (Presidente da Trolltech). Em 1991, Haavard começou a escrever as primeiras classes do Qt e, em 1993, Haavard e Eirik desenvolveram os primeiros componentes gráficos do Qt (BLANCHETTE; SUMMERFIELD, 2006).

A empresa Trolltech foi fundada em 4 de março de 1994, originalmente com o nome *Quasar Technologies*. Em Abril de 1995, esta empresa foi contratada pela *Metis*, uma empresa norueguesa, para desenvolver um software baseado em Qt, surgindo a partir daí seu primeiro cliente. Em março de 1996, a Agência Espacial Europeia tornou-se seu segundo cliente, comprando dez licenças de softwares comerciais (BLANCHETTE; SUMMERFIELD, 2006).

Em 2008, a Nokia adquiriu os direitos sobre o *framework* Qt. Desde então, o desenvolvimento de aplicações em Qt foi voltado para os dispositivos da Nokia, como a plataforma *Symbian*. Entretanto, em 2011, esta mesma empresa anunciou sua decisão de abandonar as tecnologias baseadas no *Symbian* e usar tecnologias *Microsoft* em seus aparelhos. Desta forma, a Digia, uma empresa finlandesa, passou a assumir o licenciamento comercial e serviços de suporte do Qt. Atualmente, o *framework* é mantido pelo Qt *Project*, uma iniciativa de software livre, formada por desenvolvedores individuais e provenientes de empresas já citadas, como Nokia e Digia.

A primeira versão do Qt (Qt 1.0) foi lançada em 24 de setembro de 1996. No final deste mesmo ano, a versão 1.1 foi lançada, vendendo 18 licenças. Em junho de 1999, Qt 2.0



foi lançado, oferecendo uma nova licença de código aberto, a *Q Public License* (QPL). O Qt 3.0 foi lançado em 2001, com versões disponíveis para Windows, Mac OS X, Unix e Linux. Em relação ao Qt 2.0, a versão 3.0 trouxe 42 novas classes. Em 2005, o Qt 4.0 foi lançado contando com cerca de 500 classes e mais de 9000 funções. Houve um grande avanço em relação às versões anteriores com melhorias que incluem entre uma avançada e eficiente arquitetura *model/view* (modelo que separa a lógica de programação da interface de usuário) e uma rápida e flexível estrutura de desenho bidimensional. Qt 4 foi a primeira edição a disponibilizar as duas licenças de software, *open source* e comercial, em todas as plataformas suportadas (BLANCHETTE; SUMMERFIELD, 2006).

O Qt está em sua quinta versão (Qt 5), lançada em dezembro de 2012. Os principais recursos oferecidos nesta nova versão são: bibliotecas OpenGL e OpenGL ES (*OpenGL for Embedded Systems*). Essas bibliotecas dão suporte a criação de aplicações que utilizam gráficos tridimensionais e, no caso do OpenGL ES, os recursos gráficos são oferecidos para sistemas embarcados utilizados em dispositivos como telefones celulares, PDAs e consoles de vídeo games. Além disso, a quinta versão oferece suporte ao desenvolvimento de aplicações para a plataforma *Android*, iOS (comercial), *Blackberry*.

Desde a fundação da Trolltech, a popularidade do *framework* Qt tem crescido continuamente. Este sucesso é uma reflexão tanto da qualidade deste *framework* quanto da usabilidade que ele proporciona aos desenvolvedores de software. O Qt deixou de ser um produto utilizado por poucos para ser usado diariamente por milhares de clientes e desenvolvedores de código aberto em todo o mundo (BLANCHETTE; SUMMERFIELD, 2006).

### 3 PROCEDIMENTOS METODOLÓGICOS

O desenvolvimento deste trabalho consiste em etapas progressivas de implementação. A primeira etapa consiste na identificação de módulos relevantes do *framework* Qt. São identificados módulos fundamentais com capacidade de implementação gráfica e suporte a rede em dispositivos móveis. São eles:

- Core: necessário em qualquer projeto Qt. Não oferece suporte gráfico, mas contém classes fundamentais, como a classe QObject;
- Gui: oferece suporte a GUI (Graphical User Interface), incluindo desde componentes de entrada elementar (como QLineEdit e QPushButton) até visualizadores de dados baseados na arquitetura *Model, View e Control* (QListView, QTableView e QTreeView);
- Network: contém classes que dão suporte a gerenciamento de redes (QNetworkAccessManager), ao modelo cliente servidor (QTcpSocket e QTcpServer) e a outros serviços básicos relacionados a redes (QHttp, QFtp, entre outros.);
- Multithreading: possui classes que permitem computação concorrente (QThread) e comunicação segura entre processos (QMutex, QMutexLocker, QSemaphore).

A segunda etapa consiste na elaboração de um documento de requisitos. Neste documento, estão descritos os requisitos funcionais e não funcionais da aplicação desenvolvida, além de uma descrição geral da mesma.

Após a elaboração do documento de requisitos, a próxima etapa consiste no desenvolvimento de um protótipo que simula a comunicação básica entre várias instâncias da mesma aplicação. Estas instâncias faziam solicitações de conexão com o servidor, permitindo troca de mensagens.

A próxima etapa trata da transmissão de dados gráficos simples (formato vetorial), ou seja, a transmissão de mensagens é expandida à transmissão de desenhos simples. Esta etapa possibilita a primeira simulação real da lousa com capacidade de desenho livre.

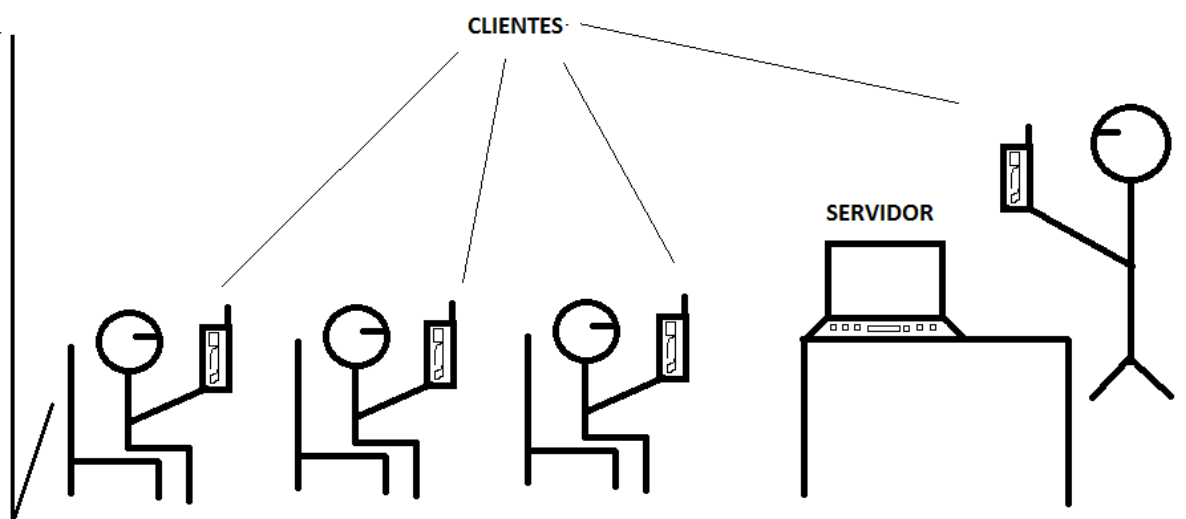
As etapas seguintes consistem da construção da aplicação propriamente dita, incluindo um lado cliente e outro servidor. Esta construção requer a definição de um ambiente de utilização, de conceitos básicos que permitam uma melhor implementação, de um critério de processamento de dados que possibilite a exibição de uma mesma imagem em *displays* de dimensões diferentes, de estruturas de dados eficazes e capazes de armazenar construções gráficas primitivas e de um modelo de envio e processamento de dados.

## 4 DESENVOLVIMENTO DA APLICAÇÃO

### 4.1 Ambiente de utilização

A lousa móvel foi projetada para ser utilizada em aulas presenciais. Para a simulação do aplicativo, são necessários uma máquina para execução do lado servidor da aplicação e pelo menos dois dispositivos móveis (*notebooks*, *smartphones* ou *tablets*) que executam o lado cliente. A aplicação do lado servidor é responsável por fazer a comunicação entre os clientes (que incluem alunos e professor). A aplicação cliente é responsável por manipular entrada de usuário (“traço livre” gerado com *mouse* ou toque/arraste em tela sensível) além de redirecioná-la, via aplicativo servidor, aos demais clientes conectados.

Figura 1 – Ambiente de Utilização da Lousa Móvel.



Fonte: elaborada pela autora.

Como mostrado na Figura 1, o ambiente físico resume-se a uma sala contendo os equipamentos citados anteriormente. Por meio de uma rede interna, alunos e professor, portando seus dispositivos móveis, podem solicitar conexão ao aplicativo servidor. Este aplicativo mantém uma lista dos clientes conectados (*sockets*), o que permite o apropriado redirecionamento de dados entre eles.

### 4.2 Conceitos Básicos

A seguir é descrita a terminologia utilizada no desenvolvimento da lousa móvel. A aplicação possui dois lados: cliente e servidor. **Cliente** refere-se à aplicação, cujas funcionalidades são conexão ao servidor, desenho, e visualização do conteúdo desenhado por

outros clientes. **Servidor** refere-se à aplicação que permite gerenciamento de conexões e tráfego de dados entre clientes. Os dados gerados por cada cliente conectado são replicados, via servidor, nos demais clientes, permitindo a sensação de página compartilhada onde todos podem desenhar ao mesmo tempo.

**Emissor** refere-se ao cliente quando está enviando dados pela rede. **Receptor** refere-se ao cliente quando está recebendo dados pela rede. Um mesmo cliente pode ser ao mesmo tempo emissor (quando envia desenho que constrói) e receptor (quando recebe desenho construído por outros clientes através da rede).

**Ponto** refere-se a menor unidade gráfica representável na lousa. Representa uma coordenada cartesiana  $(x,y)$  no *display* do dispositivo, onde  $x$  denota coluna e  $y$  a linha. A origem  $(0,0)$  situa-se no topo esquerdo, e os eixos coordenados  $x$  e  $y$  dispõem-se respectivamente da esquerda para a direita e de cima para baixo.

**Tela de desenho** refere-se a interface dos clientes onde são feitos os rabiscos. Nela é possível desenhar utilizando o *mouse*, ou o toque no caso de telas sensíveis. **Traço** representa um grupo de pontos conectados. Através do click/arraste do *mouse* ou do toque/arraste em telas sensíveis é possível gerar uma quantidade finita de pontos, gerando assim um traço. **Página** representa um grupo de traços. A página que contém o foco do desenho é chamada de **página corrente**.

### 4.3 Normalização e Desnormalização de Pontos

Devido às diferentes dimensões e resoluções de *displays* existentes, o valor absoluto das coordenadas de um ponto num dispositivo pode não ser apropriado em outro. A solução para este problema é a normalização<sup>1</sup> das coordenadas dos pontos.

Para normalizar a abscissa  $x$  de um ponto é necessário dividir seu valor pela largura do *display*, ou seja,

$$\bar{x} = \frac{x}{w}$$

onde  $\bar{x}$  denota abscissa normalizada e  $w$  a largura do *display*. Similarmente, para normalizar a ordenada  $y$  deve-se dividir seu valor pela altura do *display*, ou seja,

$$\bar{y} = \frac{y}{h}$$

onde  $\bar{y}$  denota a ordenada normalizada e  $h$  a altura do *display*.

---

<sup>1</sup> Normalizar um conjunto de valores numéricos significa aplicar-lhes uma transformação de forma que os valores finais fiquem entre 0 e 1.

No cliente emissor os pontos capturados pelo *mouse*/toque são normalizados e enviados, via aplicação servidor, para o cliente receptor. No receptor, os pontos precisam ser desnormalizados, ou seja, readaptados às dimensões do dispositivo de destino, para enfim serem desenhados. Para desnormalizar as coordenadas de um ponto normalizado devemos aplicar,

$$x = \bar{x} * w$$

$$y = \bar{y} * h$$

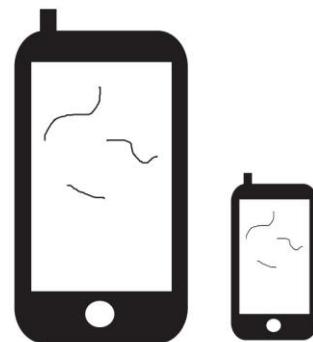
onde  $x$  e  $y$  denotam as coordenadas absolutas no dispositivo destino,  $\bar{x}$  e  $\bar{y}$  as coordenadas normalizadas provenientes da rede, e  $w$  e  $h$  a largura e a altura do *display* destino, respectivamente.

Nas figuras 2 e 3, é ilustrado o envio de pontos (conectados como traços) entre dispositivos de dimensões diferentes. Na Figura 2, os pontos estão sendo enviados em coordenadas absolutas (sem normalização/desnormalização). Os traços gerados no dispositivo de *display* maior são redenhados na mesma proporção no dispositivo de *display* menor, fazendo com que apenas uma parte do desenho seja visualizada. Na Figura 3, onde os pontos são normalizados/desnormalizados, a proporção é corrigida apropriadamente permitindo que todo conteúdo seja visualizado.

Figura 2 – Ponto Absoluto.



Figura 3 – Ponto Normalizado.



Fonte: elaboradas pela autora.

#### 4.4 Gerenciamento de Dados no Cliente

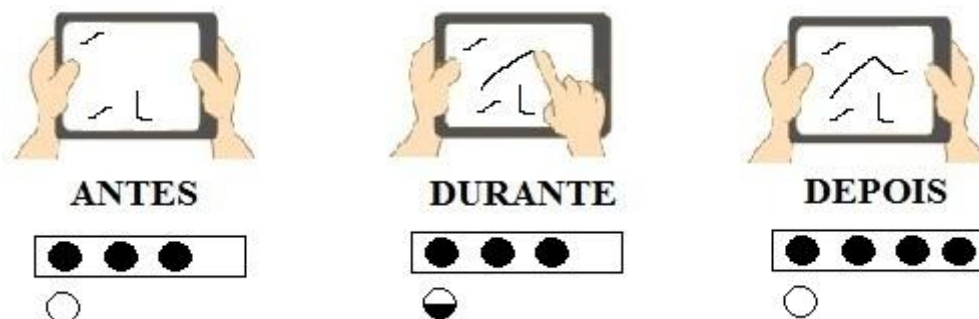
Os dados no cliente são mantidos por três variáveis: a página corrente, o traço corrente e os traços remotos. A página corrente mantém uma lista de traços já construídos. Tais traços são redenhados na tela de desenho sempre que necessário. O **traço corrente** refere-se ao

traço construído localmente pelo usuário. Ele é desenhado sempre que o *mouse*/dedo é arrastado, permitindo ao usuário visualizar em tempo real o que está fazendo. **Traços remotos** refere-se a uma tabela de dispersão, cuja chave é o descritor do *socket* emissor e o valor é uma lista de traços construídos por outros clientes da mesma conexão. O traço corrente de cada usuário emissor define uma instância de traço remoto em cada uma das tabelas de dispersão dos clientes receptores, permitindo que eles armazenem e exibam conteúdo gerado remotamente. Os traços remotos são redesenhados sempre que alguma informação chega pela rede, permitindo a cada usuário visualizar em tempo real o que os demais clientes emissores estão desenhando.

O *mouse*/dedo possui três estados funcionais: pressionado/tocado, arrastado e solto. Quando pressionado, a variável traço corrente é esvaziada e logo preparada para receber novos pontos. Quando em arraste, as coordenadas capturadas são normalizadas e guardadas no traço corrente. Quando solto, uma cópia de traço corrente é gravada na página corrente, finalizando um ciclo de construção de traço.

A Figura 4 ilustra o processo descrito anteriormente. Os retângulos horizontais de bolas cheias ilustram a página corrente. As três bolas cheias denotam os três traços ilustrados no *display*. Quando o *display* é pressionado/tocado (estágio ANTES), a variável traço corrente, denotada pela bola vazia, é preparada para receber pontos. Durante o arraste (estágio DURANTE), o traço corrente (bola parcialmente cheia) recebe os pontos capturados pelo *mouse*. Finalmente, quando solto (estágio DEPOIS), o traço corrente é copiado para a página corrente (retângulos de bolas cheias) e em seguida esvaziado (volta a ser bola vazia e uma nova bola cheia entra na página corrente).

Figura 4 – Processo de Construção de Traço Corrente.



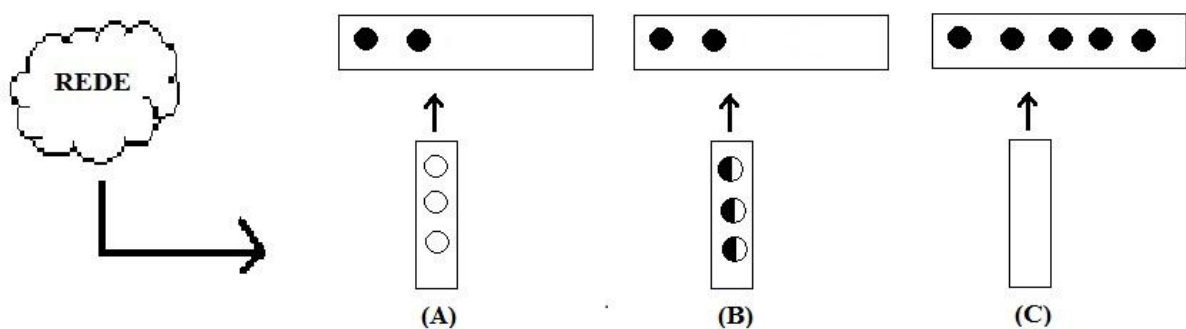
Fonte: elaborada pela autora.

A rede possui três estados funcionais no que diz respeito à construção de traços remotos: início de traço, componente de traço e fim de traço. Quando o *mouse*/dedo é

pressionado/tocado em um *display* de cliente emissor, ele envia pela rede, para os clientes receptores conectados, uma notificação de início de traço que implicará na construção de uma nova instância de traço remoto em cada uma das tabelas de dispersão (traços remotos) destes receptores. Enquanto o *mouse*/dedo estiver arrastando, os pontos capturados são enviados pela rede como notificações de componentes de traços e anexados nos traços remotos correspondentes em cliente receptor. Quando o *mouse*/dedo é solto, uma nova notificação é enviada pelo emissor aos receptores indicando que o traço remoto deve ser copiado para a página corrente e depois eliminado.

A Figura 5 ilustra o processo anterior. Os retângulos horizontais de bolas cheias ilustram estados da página corrente. Os retângulos verticais retratam estados da variável traços remotos (que é uma lista de traços). Em (A), são mostradas três instâncias de traços, em traços remotos, que denotam três clientes emissores. Estas instâncias são criadas sempre que notificações de início de traço, enviadas por clientes emissores, chegam pela rede. Em (B), quando notificações de componente de traço chegam pela rede, cada uma das instâncias em traços remotos recebe pontos da rede (bolas parcialmente cheias). Em (C), quando notificações de fim de traço chegam pela rede, as instâncias em traços remotos são copiadas para a página corrente e eliminadas. Note que só existem instâncias de traços, em traços remotos, somente enquanto o cliente emissor relacionado estiver desenhando.

Figura 5 – Processo de Construção de Traços Remotos.



Fonte: elaborada pela autora.

#### 4.5 Modelo de Envio de Dados

O modelo *bloco-comando* representa uma abstração criada para tratar a transmissão de informações entre clientes. Consiste na definição de um grupo de comandos de tamanho fixo (blocos) que são enviados pela rede com intuito de executar uma ação remota.

Um **Bloco** representa uma estrutura de dados de onze *bytes* cujos campos são apresentados no Quadro 1. O identificador de comando denota um inteiro que representa um dos possíveis comandos a ser executado. O descritor de *socket* especifica qual cliente fez a requisição. O conteúdo denota o que deve ser enviado. Ele contém oito *bytes* para que possa abranger um ponto com coordenadas de quatro bytes cada, podendo, entretanto representar outros dados.

Quadro 1 – Bloco.

<b>Tamanho (<i>bytes</i>)</b>	<b>Descrição</b>
1	Identificador de Comando
2	Descritor do Socket
8	Conteúdo

Fonte: elaborado pela autora.

Foram construídos três comandos: *Press*, *Move* e *Release*. Eles estão relacionados aos eventos do *mouse*/toque. *Press* ocorre quando o *mouse*/dedo é clicado/tocado e indica que a lousa será riscada. O bloco enviado contém o comando, o descritor do cliente e os oitos *bytes* referentes ao ponto. *Move* ocorre quando o *mouse*/dedo é arrastado. Um bloco é construído e enviado cada vez que a posição do *mouse*/dedo muda. Para todos eles, o identificador e o descritor são os mesmos, mas os oitos *bytes* finais são definidos com o valor da coordenada normalizada. *Release* ocorre quando o *mouse*/dedo é solto. O bloco é construído e enviado com identificador distinto que encerra o ciclo de construção.

O envio de blocos pela rede passa por três etapas: teste de conexão com a aplicação servidor, montagem do bloco e envio propriamente dito. Os blocos recebidos do cliente receptor, pela aplicação servidor, são redirecionados, através de um laço, a todos os clientes receptores.

#### **4.6 Processamento de Dados Provenientes da Rede no Cliente**

Dados provenientes da rede chegam fragmentados e precisam ser montados. Como cada bloco possui onze *bytes*, então é necessário enfileirar os dados que chegam e processá-los somente quando, pelo menos, um bloco já estiver sido formado. De fato, cada vez que a fila responsável pelo armazenamento de dados recebe novos dados, ela é submetida a um laço que extrai e processa o máximo de blocos possíveis, ou esquematicamente:



```

Enquanto tamanho (fila) >= 11 faça
    bloco <- desenfileira bloco de 11 bytes
    processa bloco

```

O processamento de um bloco consiste em verificar o valor do identificador de comando e aplicar o processamento apropriado. O pseudocódigo a seguir ilustra o processamento dos comandos *press*, *move* e *release*:

```

Caso (identificador de comando do bloco) seja:
    id de press: início de traço
    id de move: tratar pontos e redesenhar
    id de release: encerrar traço

```

Esta estrutura é expansível, ou seja, permite que outros comandos possam ser inseridos futuramente.

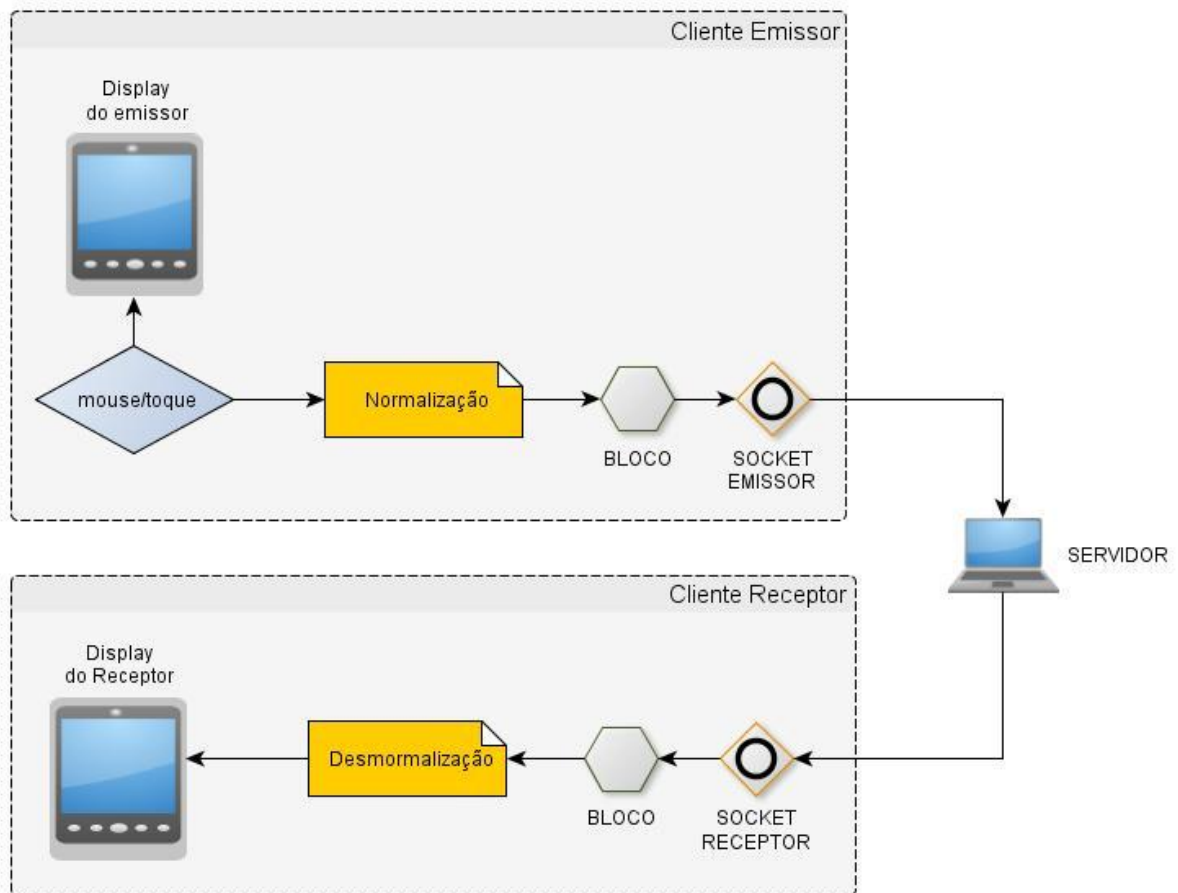
#### 4.7 Função da Aplicação Servidor

A aplicação servidor possui duas funcionalidades: gerenciar conexões e permitir o tráfego de dados entre clientes. Várias instâncias da aplicação cliente conectam-se a uma única instância da aplicação servidor. Os dados são enviados pelos clientes diretamente ao servidor que os reenvia aos demais clientes conectados, exceto aquele que enviou os dados. Os clientes são diferenciados, no lado servidor, pelos valores de seus descritores incorporados nos blocos. Os clientes processam os dados que recebem, simultaneamente, e mantêm a sensação de página compartilhada.

#### 4.8 Fluxogramas da Aplicação

O Fluxograma 1 ilustra o relacionamento, na aplicação da lousa móvel, entre dois clientes, receptor e emissor, e o servidor. O cliente emissor, através do *mouse*/toque, produz traços correntes em seu *display*. Estes têm seus pontos normalizados e empacotados em blocos. Os blocos são enviados para o servidor por meio do *socket* (emissor). No servidor, os blocos são redirecionados para o cliente receptor. No cliente receptor, o *socket* correspondente recebe os blocos e os reencaminha para desnormalização dos pontos e posterior reconstrução dos traços. Os traços desnormalizados são por fim desenhados no *display* do receptor.

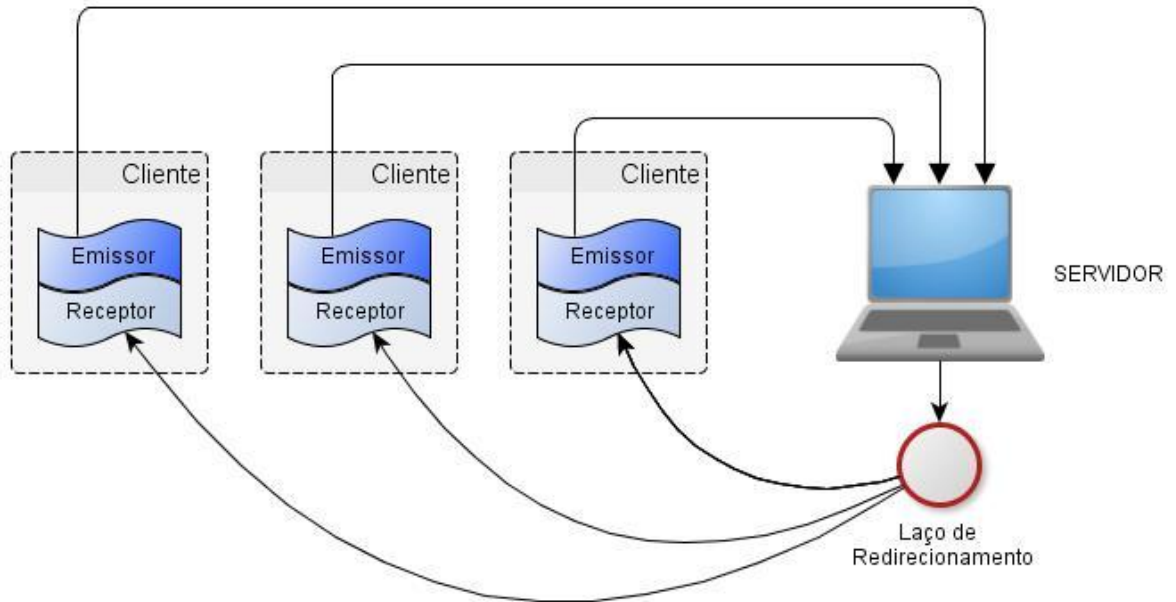
Fluxograma 1 – Relacionamento entre dois Clientes (Emissor e Receptor).



Fonte: elaborado pela autora.

O Fluxograma 2 ilustra a concorrência quando dois ou mais clientes estão envolvidos no relacionamento. Cada cliente é simultaneamente emissor e receptor (emissor quando desenha e receptor recebe desenho dos demais clientes). Como emissor, os dados oriundos dos eventos de desenho são enviados para o servidor. Note que todos os clientes podem enviar seus dados ao mesmo tempo. Os descritores guardados nos blocos enviados pelos clientes emissores os identificam na rede. O servidor redireciona os dados para todos os clientes receptores utilizando um laço. Em cada iteração desse laço, uma cópia do conteúdo recebido no servidor é enviada para um cliente receptor, permitindo que todos os clientes receptores recebam os mesmos dados.

Fluxograma 2 – Relacionamento Concorrente entre dois ou mais Clientes



Fonte: elaborado pela autora.

## 5 TESTE DA APLICAÇÃO

### 5.1 Caso de Teste – Comunicação entre dois clientes com interação restrita

O objetivo deste caso de teste é identificar como a aplicação se comportou durante a comunicação entre dois clientes (um emissor e outro receptor), conectados ao mesmo aplicativo servidor. Nesta comunicação, o cliente emissor escreve na lousa e ambos visualizam o conteúdo gerado. O caso de teste está descrito no Quadro 2.

Quadro 2 – Caso de Teste 1.

<b>Caso de Teste:</b>	CT01	<b>Ambiente de Teste</b>	Android/Linux
<b>Aplicação:</b>	Lousa Móvel.		
<b>Descrição:</b>	Testar a comunicação entre dois clientes (um emissor e outro receptor), onde o cliente emissor escreve e ambos visualizam o conteúdo gerado.		
<b>Pré-Condições:</b>	Os clientes devem estar conectados ao servidor da aplicação.		
<b>Passo(s):</b>	<b>Ação</b>	<b>Resposta Esperada</b>	<b>Passou/Falhou</b>
1	O cliente emissor faz um traço na lousa.	Os dois clientes visualizam o traço.	Passou
<b>Pós-Condições:</b>	A comunicação entre dois clientes (um emissor e outro receptor), onde apenas um risca e ambos visualizam, foi realizada com sucesso.		

Fonte: elaborado pela autora.

Na execução deste caso de teste, os dois clientes (um emissor e outro receptor) conseguiram se conectar ao servidor da aplicação por meio de seus dispositivos móveis. O cliente emissor produziu traço na lousa (traço corrente), enquanto o outro cliente (receptor) pôde visualizá-lo em seu dispositivo (traço remoto). A aplicação se comportou como esperado durante a comunicação e geração de conteúdo. Durante a produção do traço, não houve atraso na replicação do mesmo no dispositivo do outro cliente.

### 5.2 Caso de Teste – Comunicação entre dois clientes com interação simultânea

O objetivo deste caso de teste é identificar como a aplicação se comportou durante a comunicação entre dois clientes (emissores e receptores ao mesmo tempo), conectados ao mesmo aplicativo servidor. Nesta comunicação, os dois clientes escrevem simultaneamente na lousa e ambos visualizam o conteúdo gerado. O caso de teste está descrito no Quadro 3.

Quadro 3 – Caso de Teste 2.

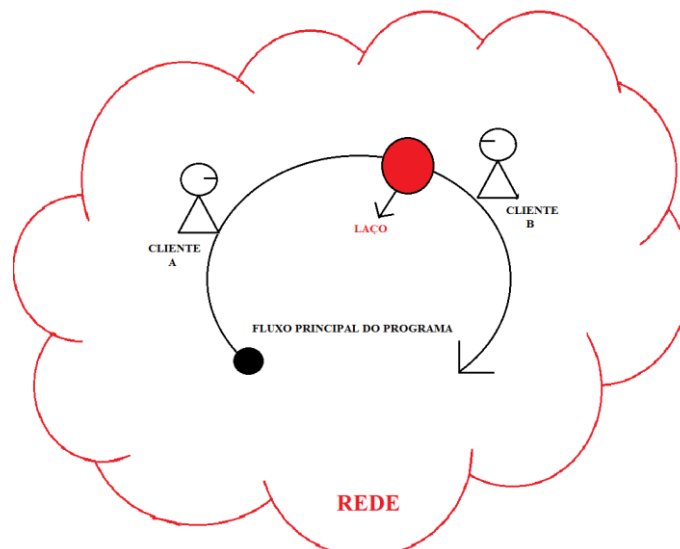
<b>Caso de Teste:</b>	CT02	<b>Ambiente de Teste</b>	Android/Linux
<b>Aplicação:</b>	Lousa Móvel.		
<b>Descrição:</b>	Testar a comunicação entre dois clientes (emissores e receptores ao mesmo tempo), onde ambos escrevem simultaneamente e visualizam o conteúdo gerado.		
<b>Pré-Condições:</b>	Os clientes devem estar conectados ao servidor da aplicação.		
<b>Passo(s):</b>	<b>Ação</b>	<b>Resposta Esperada</b>	<b>Passou/Falhou</b>
1	Os dois clientes fazem traços na lousa.	Os dois clientes visualizam os traços.	Falhou
<b>Pós-Condições:</b>	A comunicação entre dois clientes, onde ambos riscam e visualizam, foi realizada com sucesso.		

Fonte: elaborado pela autora.

Na execução deste caso de teste, os dois clientes (emissor e receptor ao mesmo tempo) conseguiram se conectar ao servidor da aplicação por meio de seus dispositivos móveis. Ambos produziram traços na lousa, entretanto, a aplicação não se comportou como esperado. Neste caso, quando um dos clientes escrevia, a aplicação não permitia ao outro escrever.

O processamento de dados da fila de *bytes*, que chegam pela rede, é executado dentro de um laço. Este laço executa enquanto a quantidade de *bytes* for maior ou igual ao tamanho de um bloco (onze *bytes*). Se novos blocos chegam durante a execução do laço ou uma quantidade muito grande de blocos precisa ser processada, então este laço poderá demandar de muito tempo, comprometendo o laço principal da aplicação e, conseqüentemente, a repintura. Sem repintura, os traços correntes demoram a ser redesenhados, prejudicando a visualização.

Figura 6 – Modelo de Interação Simultânea.

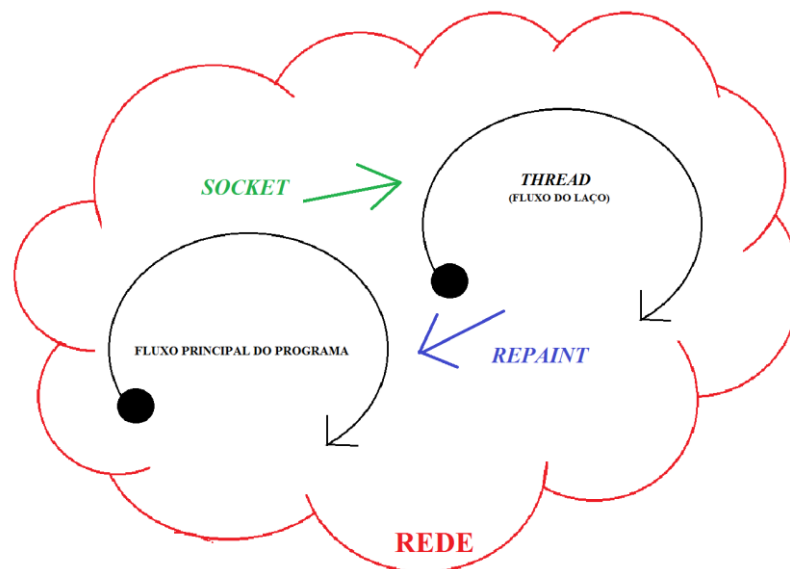


Fonte: elaborada pela autora.

A Figura 6 ilustra dois clientes conectados, em uma determinada rede, por meio do servidor da aplicação. Ambos podem produzir e visualizar traços numa página compartilhada. A pintura de traços pelo Cliente A ocorre no fluxo principal da aplicação, enquanto do Cliente B ocorre no laço de tratamento de dados da rede. Desta forma, sempre que houver blocos a serem processados, a execução do laço não é finalizada e o fluxo principal é retardado ou mesmo interrompido. Na imagem, o ponto em vermelho representa o momento em que o controle do fluxo principal é interrompido, impedindo a repintura no dispositivo.

Para resolução do problema, foi proposta a utilização de *threads* conforme ilustrado no caso de teste seguinte.

Figura 7 – Modelo de Interação Simultânea com *Thread*.



Fonte: elaborada pela autora.

### 5.3 Caso de Teste – Comunicação entre quatro clientes com interação simultânea usando *threads*

O objetivo deste caso de teste é identificar como a aplicação se comportou durante a comunicação entre quatro clientes conectados (emissores e receptores ao mesmo tempo) ao mesmo servidor. Nesta comunicação, os clientes escrevem simultaneamente na lousa e todos visualizam o conteúdo gerado.

Um modelo baseado em *threads* foi adotado (Figura 7). O fluxo principal agora independe do laço que processa dados da rede o qual ocorrerá um *thread* independente. O *socket* de cada cliente é mantido por seu respectivo *thread* de forma que os blocos

provenientes da rede são processados concorrentemente às operações locais como entrada de *mouse* e repintura.

O caso de teste está descrito no Quadro 4.

Quadro 4 – Caso de Teste 3.

<b>Caso de Teste:</b>	CT03	<b>Ambiente de Teste</b>	Android/Windows/Linux
<b>Aplicação:</b>	Lousa Móvel.		
<b>Descrição:</b>	Testar a comunicação entre quatro clientes (emissores e receptores ao mesmo tempo), onde todos escrevem simultaneamente e visualizam o conteúdo gerado.		
<b>Pré-Condições:</b>	Os clientes devem estar conectados ao servidor da aplicação.		
<b>Passo(s):</b>	<b>Ação</b>	<b>Resposta Esperada</b>	<b>Passou/Falhou</b>
1	Os clientes fazem traços na lousa.	Os clientes visualizam os traços.	Passou
<b>Pós-Condições:</b>	A comunicação entre quatro clientes, onde todos riscam e visualizam, foi realizada com sucesso.		

Fonte: elaborado pela autora.

Com o novo modelo, a comunicação entre quatro clientes conectados foi executada com êxito, ou seja, os clientes conectados conseguiram, simultaneamente, desenhar e visualizar os traços construídos remotamente. A latência de atualização de traços remotos demonstrou ser sensível à qualidade da rede e à capacidade de processamento dos dispositivos utilizados. Quanto melhor a rede e a capacidade de processamento do dispositivo, mais satisfatório é o efeito de página de desenho compartilhada.

## 6 CONSIDERAÇÕES FINAIS

Nesta seção são feitas as considerações finais acerca do trabalho realizado, focando nas contribuições, limitações e trabalhos futuros.

### 6.1 Conclusão

No presente trabalho foram realizados estudos detalhados sobre o *framework Qt*, identificando os principais módulos, com ênfase em dispositivos móveis, utilizados no desenvolvimento da lousa móvel.

A aplicação desenvolvida nesse trabalho comprovou, através de testes realizados, ter cumprido seus objetivos, ou seja, permitiu que vários usuários se conectassem e interagissem com a lousa dentro de uma rede local. Por meio da aplicação servidor, os usuários se conectaram com sua aplicação cliente e compartilharam informações entre si. Os dados gerados pelos clientes foram enviados ao servidor e este, por sua vez, os redistribuiu entre os clientes conectados.

À medida que cresce o mercado de dispositivos móveis, aumenta também a diversidade de aplicativos que são desenvolvidos para esses aparelhos. Empresas de desenvolvimento criam cada vez mais aplicativos que trocam dados na rede e se comunicam com servidores. Contudo, apesar de todo avanço tecnológico, a transferência de grandes quantidades de dados via rede é sempre problemático devido às limitações da própria rede ou mesmo do aparelho.

O problema acima se trata de uma limitação encontrada no presente trabalho. Durante os testes, verificou-se que a geração de muitos traços ocasionou um pequeno atraso na replicação dos mesmos nos dispositivos dos outros clientes. Desta forma, na aplicação do trabalho em questão, é essencial que os dados sejam simples para que o tráfego dos mesmos ocorra com sucesso.

### 6.2 Trabalhos Futuros

Durante a execução do trabalho proposto foram identificadas algumas possibilidades de trabalhos futuros, são eles:

- a) incorporar algumas funcionalidades básicas características de uma lousa na aplicação cliente, tais como desfazer traço, inserir nova página, salvar, copiar/colar, entre outras.



Tais funcionalidades estão descritas no documento de requisitos que se encontra no apêndice;

- b) aprimorar o gerenciamento de usuários na aplicação servidor. Permitir que os usuários sejam identificados por meio de um apelido e não mais apenas por um descritor. No momento da conexão, além de informar o IP, o usuário informaria também o seu apelido que seria então identificado pela aplicação servidor e mostrado na lista de clientes conectados;
- c) permitir que um cliente (professor) controle a interação dos demais clientes na lousa. A lista de clientes conectados, inicialmente, mostraria todos eles bloqueados para escrita. Apenas o professor iniciaria desbloqueado e este teria permissão de bloquear/desbloquear um determinado cliente (aluno) para uso da lousa;
- d) permitir a formação de grupos de alunos. O professor poderia criar páginas específicas para um grupo de alunos, assim os trabalhos em grupos se tornariam mais dinâmicos.

## REFERÊNCIAS

- BLANCHETTE, Jasmin; SUMMERFIELD, Mark. **C++ GUI Programming with Qt 4**. 2ed. Massachusetts: Prentice Hall, 2006. 556p.
- DIAS, K. L.; SADOK, D. F.H. **Internet Móvel: Tecnologias, Aplicações e QoS**. XIX Simpósio Brasileiro de Redes de Computadores, Florianópolis, 2001.
- FIGUEIREDO, C. M. S.; NAKAMURA, E. **Computação Móvel: Novas Oportunidades e Novos Desafios**. T & C Amazônia, 2003.
- IDGNOW. **Dispositivos móveis vão superar número de pessoas em 2013, diz estudo**. Disponível em < <http://idgnow.uol.com.br/mobilidade/2013/02/11/aparelhos-moveis-conectados-vaosuperar-numero-de-pessoas-em-2013-diz-estudo/>>. Acesso em: 18 mai. 2013.
- INFO. **Samsung espera finalizar a tecnologia 5G em 2020**. Disponível em < <http://info.abril.com.br/noticias/ti/samsung-espera-finalizar-a-tecnologia-5g-em-2020-12052013-2.shl>>. Acesso em: 17 mai. 2013.
- JUNIOR, J. C. C. **Lousa interativa de alta resolução**. São Paulo: FATEC/Tatuí, 2009.
- MOREIRA, G. L. S. M. **Desenvolvimento de uma lousa digital utilizando transdutores ultrassônicos**. Projeto de Fim de Curso I (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Minas Gerais, Minas Gerais, 2009.
- NAKASHIMA, R., H. R.; AMARAL, S. F. **A linguagem audiovisual da lousa digital interativa no contexto educacional**. Campinas: Educação Temática Digital, 2006.
- SCHAEFER, C. **Protótipo de aplicativo para transmissão de dados a partir de dispositivos móveis aplicados a uma empresa de transporte**. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2004.
- SHEMBERGER, E. E.; FREITAS, I.; VANI, R. **Plataforma Android**. Universidade Estadual do Oeste do Paraná, Paraná, 2009.
- QT DIGIA. **Qt**. Disponível em < <http://qt.digia.com/Product/>>. Acesso em: 12 jun. 2013.
- VÍRGILIO, E. **Aplicações para dispositivos móveis para Gestão de Pessoas**. Disponível em < <http://www.rh.com.br/Portal/Mudanca/Artigo/8492/aplicacoes-para-dispositivos-moveis-para-gestao-de-pessoas.html>>. Acesso em: 15 mai. 2013.

# APÊNDICES

**APÊNDICE A – DOCUMENTO DE REQUISITOS**

## **Documento de Requisitos - Lousa Móvel**

**Versão 2.0**

## Histórico de Alterações

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
16/07/2013	1.0	Criação do documento	Thalita Maria
03/01/2014	2.0	Remoção de alguns casos de uso	Thalita Maria

## Sumário

<b>1. INTRODUÇÃO .....</b>	<b>4</b>
1.1 VISÃO GERAL DO DOCUMENTO .....	4
1.2 CONVENÇÕES, TERMOS E ABREVIACÕES.....	4
1.2.1 Identificação dos requisitos.....	4
1.2.2 Prioridades dos requisitos.....	4
<b>2. DESCRIÇÃO GERAL DA APLICAÇÃO.....</b>	<b>5</b>
<b>3. REQUISITOS FUNCIONAIS (CASOS DE USO) .....</b>	<b>6</b>
3.1 AULAS.....	6
[RF001] Criar aula .....	6
[RF002] Abrir aula.....	6
[RF003] Salvar aula.....	6
[RF004] Enviar aula .....	6
[RF005] Gerar aula.....	7
3.2 FERRAMENTAS.....	7
3.2.1 Ferramentas de desenho.....	7
[RF001] Fornecer ferramenta caneta .....	7
[RF002] Fornecer ferramenta marca-texto.....	8
[RF003] Fornecer ferramenta borracha .....	8
[RF004] Fornecer ferramenta preenchimento.....	8
[RF005] Fornecer ferramenta tamanho .....	9
3.2.2 Ferramentas de texto.....	9
[RF006] Fornecer ferramenta de edição de texto.....	9
[RF007] Fornecer ferramenta de caixa de texto .....	9
3.2.3 Ferramentas Multimídia.....	10
[RF008] Fornecer ferramenta importar.....	10
[RF009] Fornecer ferramenta modo vidro.....	10
3.3 CONEXÃO.....	10
[RF001] Estabelecer conexão.....	10
[RF002] Sincronizar conteúdo ao conteúdo dos dispositivos .....	11
3.4 GERENCIADOR DE USUÁRIOS .....	11
[RF001] Exibir usuários pareados.....	11
[RF002] Desbloquear usuário .....	11
[RF003] Bloquear usuário.....	12
3.5 RELATÓRIOS.....	12
[RF001] Gerar relatórios .....	12
<b>4. REQUISITOS NÃO FUNCIONAIS .....</b>	<b>12</b>
[NF001] Usabilidade.....	12
[NF002] Desempenho.....	13

## 1. Introdução

Este documento especifica oficialmente os requisitos funcionais e não funcionais da aplicação Lousa Móvel. Serve como base para a equipe de desenvolvimento, fornecendo informações necessárias para a implementação das funcionalidades ao longo de todo o ciclo de vida do produto.

### 1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

- **Seção 2 – Descrição geral da aplicação:** apresenta uma visão geral da aplicação, caracterizando qual é o seu escopo e descrevendo seus usuários;
- **Seção 3 – Requisitos funcionais (casos de uso):** especifica todos os casos de uso da aplicação, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado;
- **Seção 4 – Requisitos não-funcionais:** especifica todos os requisitos não funcionais da aplicação, divididos em requisitos de usabilidade e de confiabilidade.

### 1.2 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

#### 1.2.1 Identificação dos requisitos

As abreviações usadas no documento são:

- RF00X – Requisito Funcional;
- RNF00X – Requisito Não Funcional;

#### 1.2.2 Prioridades dos requisitos



Para estabelecer a prioridade dos requisitos, nas seções 4 e 5, foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial** é o requisito sem o qual a aplicação não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante** é o requisito sem o qual a aplicação entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, a aplicação poderá ser implantado e usado mesmo assim.
- **Desejável** é o requisito que não compromete as funcionalidades básicas da aplicação, isto é, a aplicação pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores da aplicação, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

## 2. Descrição geral da aplicação

A Lousa Móvel tem por objetivo permitir a criação de aulas dinâmicas e interativas. Ela está dividida em dois módulos: módulo cliente e módulo servidor. O usuário, em seu dispositivo móvel, poderá gerar conteúdos por meio da aplicação cliente. Esta por sua vez poderá ser sincronizada aos dispositivos móveis de outros usuários, permitindo a sensação de página compartilhada. A aplicação servidor servirá para guardar os clientes conectados e redistribuir conteúdos compartilhados entre eles. A próxima seção mostrará todas as funcionalidades da aplicação.

### 3. Requisitos funcionais (casos de uso)

#### 3.1 Aulas

##### [RF001] Criar aula

**Descrição do caso de uso:** Este caso de uso permite que o usuário crie uma aula.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** não tem.

**Pós-condição:** o usuário cria a aula desejada.

##### [RF002] Abrir aula

**Descrição do caso de uso:** Este caso de uso permite ao usuário abrir uma aula para edição.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** a aula deve está salva.

**Pós-condição:** o usuário consegue abrir a aula desejada.

##### [RF003] Salvar aula

**Descrição do caso de uso:** Este caso de uso permite que o usuário salve uma aula.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** a aula deve ter sido criada ou modificada.

**Pós-condição:** o usuário consegue salvar a aula desejada.

##### [RF004] Enviar aula

**Descrição do caso de uso:** Este caso de uso permite que o usuário envie uma aula por e-mail.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** a aula deve está salva.

**Pós-condição:** o usuário consegue enviar a aula desejada.

<b>[RF005] Gerar aula</b>
---------------------------

**Descrição do caso de uso:** Este caso de uso permite que ao usuário gerar uma aula no formato PDF.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** a aula deve está salva.

**Pós-condição:** o usuário consegue gerar a aula desejada.

## 3.2 Ferramentas

### 3.2.1 Ferramentas de desenho

<b>[RF001] Fornecer ferramenta caneta</b>
---

**Descrição do caso de uso:** Este caso de uso fornece diversas cores de caneta para o usuário. Esta ferramenta ficará ativada até que o usuário escolha outra.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** uma aula deve ter sido criada ou aberta.

**Pós-condição:** o usuário consegue escrever ou desenhar com as diversas opções de caneta oferecidas pela ferramenta.

**[RF002] Fornecer ferramenta marca-texto**

**Descrição do caso de uso:** Este caso de uso oferece marca-texto com diversas cores. Esta ferramenta ficará ativada até que o usuário escolha outra.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** uma aula deve ter sido criada ou aberta.

**Pós-condição:** o usuário consegue utilizar as diversas opções de marca-texto oferecidas pela ferramenta.

**[RF003] Fornecer ferramenta borracha**

**Descrição do caso de uso:** Este caso de uso permite que o usuário apague a tinta digital feita pela ferramenta caneta. Esta ferramenta ficará ativada até que o usuário escolha outra.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** uma aula deve ter sido criada ou aberta.

**Pós-condição:** o usuário consegue utilizar a ferramenta borracha.

**[RF004] Fornecer ferramenta preenchimento**

**Descrição do caso de uso:** Este caso de uso permite que o usuário aplique uma cor a uma forma ou ao fundo da tela. Esta ferramenta ficará ativada até que o usuário escolha outra.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** uma aula deve ter sido criada ou aberta.

**Pós-condição:** o usuário consegue utilizar a ferramenta preenchimento.

**[RF005] Fornecer ferramenta tamanho**

**Descrição do caso de uso:** Este caso de uso permite que o usuário selecione o tamanho, ou seja, a espessura da caneta.

**Prioridade:**      Essencial                    Importante                    Desejável

**Pré-condições:** uma aula deve ter sido criada ou aberta.

**Pós-condição:** o usuário consegue utilizar a ferramenta tamanho.

### 3.2.2 Ferramentas de texto

**[RF006] Fornecer ferramenta de edição de texto**

**Descrição do caso de uso:** Este caso de uso permite que o usuário insira texto via teclado. Fornece opção para edições: tipo, tamanho e cor de fonte, negrito, itálico e sublinhado.

**Prioridade:**      Essencial                    Importante                    Desejável

**Pré-condições:** uma aula deve ter sido criada ou aberta.

**Pós-condição:** o usuário consegue inserir e editar um texto.

**[RF007] Fornecer ferramenta de caixa de texto**

**Descrição do caso de uso:** Este caso de uso permite que ao usuário escolher o tipo de caixa de texto, com ou sem borda.

**Prioridade:**      Essencial                    Importante                    Desejável

**Pré-condições:** deve ter sido inserido um texto na aula.

**Pós-condição:** o usuário consegue escolher o tipo de caixa de texto desejado.

### 3.2.3 Ferramentas Multimídia

#### [RF008] Fornecer ferramenta importar

**Descrição do caso de uso:** Este caso de uso permite que ao usuário importar recursos de multimídia, como uma imagem ou textos em PDF quaisquer.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** não tem.

**Pós-condição:** o usuário consegue importar o arquivo desejado.

#### [RF009] Fornecer ferramenta modo vidro

**Descrição do caso de uso:** Este caso de uso permite ao usuário, por meio da caneta, fazer anotações diretas sobre o conteúdo importado.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** deve ter sido importado algum arquivo.

**Pós-condição:** o usuário consegue fazer anotações no arquivo importado.

### 3.3 Conexão

#### [RF001] Estabelecer conexão

**Descrição do caso de uso:** Este caso de uso permite que o usuário (aplicação cliente) estabeleça uma conexão, através de uma rede interna, com o servidor da aplicação.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** não tem.

**Pós-condição:** uma conexão é estabelecida.

<b>[RF002] Sincronizar conteúdo ao conteúdo dos dispositivos</b>
--

**Descrição do caso de uso:** Este caso de uso permite que o conteúdo gerado pelo usuário (aplicação cliente) seja sincronizado ao conteúdo dos dispositivos móveis de outros usuários.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** uma conexão deve ter sido estabelecida.

**Pós-condição:** o usuário consegue sincronizar o conteúdo ao conteúdo dos dispositivos móveis de outros usuários.

### 3.4 Gerenciador de usuários

<b>[RF001] Exibir usuários pareados</b>
---

**Descrição do caso de uso:** Este caso de uso permite listar, na aplicação servidor, todos os usuários (aplicação cliente) conectados.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** pelo menos uma conexão deve ter sido estabelecida.

**Pós-condição:** são exibidos os usuários conectados.

<b>[RF002] Desbloquear usuário</b>
------------------------------------

**Descrição do caso de uso:** Este caso de uso permite ao usuário (professor) desbloquear um determinado usuário (aluno) para que este possa interagir com a lousa, podendo desenhar, escrever, etc.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** conexões devem ter sido estabelecidas.

**Pós-condição:** o professor consegue desbloquear o aluno desejado.

### [RF003] Bloquear usuário

**Descrição do caso de uso:** Este caso de uso permite ao usuário (professor) bloquear um determinado usuário (aluno) para que este fique apenas em modo leitura.

**Prioridade:**      Essencial                       Importante                       Desejável

**Pré-condições:** conexões devem ter sido estabelecidas.

**Pós-condição:** o professor consegue desbloquear o aluno desejado.

## 3.5 Relatórios

### [RF001] Gerar relatórios

**Descrição do caso de uso:** Este caso de uso permite ao usuário gerar relatórios diversos, como relatórios de aulas registradas, quantidade de alunos por aula registrada, etc.

**Prioridade:**      Essencial                       Importante                       Desejável

**Pré-condições:** devem haver aulas salvas.

**Pós-condição:** o usuário consegue gerar os relatórios desejados.

## 4. Requisitos não funcionais

### [NF001] Usabilidade

A interface com o usuário é de vital importância para o sucesso da aplicação. Principalmente por ser uma aplicação móvel, onde o espaço é limitado para informações. Desta forma, uma interface amigável ao usuário é um requisito essencial.

**Prioridade:**      Essencial                       Importante                       Desejável



<b>[NF002] Desempenho</b>
---------------------------

Embora não seja um requisito essencial para a aplicação, deve ser considerada por corresponder a um fator de qualidade de software.

**Prioridade:**     Essencial                     Importante                     Desejável