

Appliance para Sistemas de Resposta a Audiência em Dispositivos Móveis

Vinícius Ribeiro¹, Jomara Mota Bindá¹, Renato R. Lopes², Eduardo Valle Jr.¹

¹DCA — Departamento de Computação e Automação Industrial - Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brasil

²Departamento de Comunicações - Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas (UNICAMP)
Campinas, SP – Brazil

vribeiro@dca.fee.unicamp.br, jbinda@dca.fee.unicamp.br,
rlopes@decom.fee.unicamp.br, dovalle@dca.fee.unicamp.br

Resumo. *Sistemas de resposta a audiência permitem coletar, em tempo real, respostas de uma audiência durante uma aula. Ordinariamente, sistemas de resposta a audiência são implementados com pequenos dispositivos eletrônicos chamados clickers. Esses dispositivos eletrônicos são caros por conta do custo de aquisição, instalação e manutenção. Neste artigo, apresentaremos os desafios de desenvolvimento de uma solução de baixo custo para sistemas de resposta a audiência, incluindo os desafios tecnológicos e de experiência do usuário. Nós propomos uma solução baseada no conceito de “traga seu próprio dispositivo” e apresentamos os resultados obtidos.*

1. Introdução

Um importante fator no ensino e aprendizado é a capacidade de manter os estudantes interessados e ativos durante a aula, participando e contribuindo com suas respostas e opiniões [Fulford and Zhang 1993].

Práticas pedagógicas ativas, como *peer instruction*, são mais fáceis de serem implementadas se o professor tiver uma maneira rápida, não invasiva, e pessoal de obter *feedback* e respostas dos alunos durante a aula [Kay and LeSage 2009]. A combinação de *peer instruction* com as tecnologias disponíveis resulta em abordagens educacionais inovadoras que melhoram o relacionamento professor-aluno na sala de aula, aumentando as interações entre eles [Fulford and Zhang 1993].

A tecnologia tem sido utilizada cada vez mais no processo de aprendizagem em escolas e universidades. Sistemas de resposta a audiência (SRA) tem um grande papel nesse contexto. SRAs permite a coleta em tempo real das reações e respostas de uma audiência durante uma apresentação ou aula.

Os sistemas de resposta a audiência disponíveis atualmente são tanto hardware e software que permitem estudantes darem respostas instantâneas a questões propostas pelos professores utilizando pequenos dispositivos eletrônicos chamados de *clickers*. A seguir, as respostas são processadas, e elas podem ser visualizadas graficamente. Esses sistemas possuem um alto preço intrínseco por conta do custo de aquisição, instalação e manutenção. O valor desses dispositivos é um grande obstáculo para escolas em áreas menos favorecidas. Além disso, *clickers* possuem limitações de hardware e não são muito flexíveis. A maioria

dos aparelhos oferecem um número limitado de tipos de respostas, o que muitas vezes não satisfaz as necessidades do professor.

O Virtualclicker mostra-se uma solução acessível para escolas com baixos orçamentos. Nosso objetivo é proporcionar a escolas rurais e periféricas a oportunidade de trabalhar com técnicas pedagógicas ativas. O projeto é baseado no conceito “traga seu próprio dispositivo” (do inglês, *bring your own device*) (BYOD), em que utiliza-se o hardware dos próprios estudantes para acessar o sistema. Dessa maneira, nós eliminamos o dispositivo eletrônico, proibitivo para nosso mercado.

2. Tecnologias e Métodos

Peer Instruction (PI) melhora a interatividade e a participação dos alunos durante a aula [Mazur 1997]. PI difere de formatos tradicionais de cursos através do questionamento dos estudantes para revelar dúvidas ocultas e assuntos mal entendidos.

PI inclui enquetes no seu procedimento. Questionar os alunos é um recurso comum para obter a visão e os pensamentos dos estudantes e isso pode ser feito através de levantar as mãos ou questionando o aluno diretamente. Entretanto, enquetes tem limitações uma vez que alunos tímidos tem dificuldades de apresentar suas ideias [Cross et. al 2012].

Sistemas de resposta a audiência em sala de aula oferecem um novo modelo de interatividade habilitando a prática do *peer instruction* [Beatty et. al 2006]. Sistemas de resposta a audiência tem sido usados em cursos de diferentes formatos, desde palestras formais a aprendizado cooperativo, e em diferentes níveis, desde aulas introdutórias a cursos avançados de graduação [Fies and Marshall 2006, Caldwell 2007].

Diversos estudos [Yourstone et. al 2008, Caldwell 2007, Bergtrom 2006] sugerem que estudantes ficam mais atentos durante a aula quando sistemas de resposta a audiência são aplicados. Além disso, SRAs permitem respostas anônimas, livres de julgamentos dos colegas de classe. Estudos [Kay and LeSage 2009, Caldwell 2007, Bergtrom 2006] mencionam que a interação frequente e positiva ocorre quando SRAs são utilizados durante sessões de discussões e resultam em boas articulações e raciocínio dos alunos [Beatty 2005] uma vez que o estudante necessita inserir sua resposta e defender seu ponto de vista quando solicitado, ele deve tentar convencer seus colegas de classe de que suas respostas estão corretas.

Além disso, existem benefícios potenciais ao professor pelo fato de que ele pode analisar o nível de aprendizado durante a aula [Yourstone et. al 2008] e mudar sua abordagem dependendo do *feedback* dos estudantes [Caldwell 2007, Cutts 2006]. Se o resultado mostra que existem mal entendimento e más interpretações a respeito de um determinado tópico na maioria da sala de aula, o professor pode testar diferentes explicações e abordagens para sanar a questão. Pesquisas [Yourstone et. al 2008, Caldwell 2007] sugerem que a performance de aprendizado cresce por conta dos benefícios de ter *feedback* imediato.

O Virtualclicker utiliza-se do conceito de “traga seu próprio dispositivo”, um modelo em que os estudantes utilizam seus dispositivos pessoais na escola de forma a utilizá-los durante a aula [Stavert 2013]. Essa solução substitui a necessidade de um *clicker* pessoal fornecendo um software ou aplicação para utilizar em um dispositivo cliente em rede.

Modelos de BYOD possuem inúmeras vantagens por conta do grande alcance de dispositivos que podem ser utilizados, e.g. equipamentos obsoletos, uma vez que as aplicações de software não requerem grande poder computacional. Ainda assim, essas soluções encontram diversos desafios dependendo do equipamento que é utilizado. Usando-se abordagens com dispositivos móveis, assumimos que os estudantes possuem seus próprios aparelhos e se sentem confortáveis em se deslocar até a escola com eles [Stavert 2013]. Além disso, estudantes que não possuem esses dispositivos não podem participar completamente da aula. A abordagem que utiliza dispositivos móveis colocam no aluno a responsabilidade de levar os aparelhos para a sala de aula, sobrecarregando o estudante [Caldwell 2007].

Utilizar dispositivos fixos requer infraestrutura adequada para suportar muitas conexões de rede e dependem da confiabilidade da rede local no laboratório da escola. Além disso, algumas vezes é necessário utilizar soluções baseadas em nuvem, o que pode causar problemas técnicos e organizacionais [Lennon 2012].

Apesar dos desafios encontrados quando utilizando BYOD, encontramos soluções simples para encará-los. O projeto Virtualclicker utiliza um sistema web hospedado em um Raspberry Pi com o servidor web Apache embarcado na plataforma e distribuído em uma rede local através de um roteador sem fio. Dessa maneira, o estudante pode conectar-se ao sistema através de qualquer dispositivo equipado com um navegador web, e.g. notebook, netbook, smartphone ou tablet. Com essa abordagem, o mais simples e barato smartphone pode acessar a aplicação, e a infraestrutura requerida é mínima.

A linguagem de programação escolhida para o desenvolvimento do projeto foi a PHP. Ela foi selecionada por ser largamente utilizada no desenvolvimento de aplicações web e possuir documentação e comunidade online grandes e acessíveis, proporcionando um aprendizado mais rápido, prático e eficiente. Interfaces ricas puderam ser adquiridas utilizando-se HTML5, CSS3 e Javascript. Linguagens que, assim como a PHP, são amplamente utilizadas no desenvolvimento web e possuem grande comunidade online. Além disso, essas linguagens são suportadas pelos principais navegadores web disponíveis no mercado. O banco de dados utilizado para o desenvolvimento do projeto foi o MySQL, que possui licença gratuita para aplicações não comerciais, o que é compatível com a nossa proposta de projeto. Esse banco de dados pode ser integrado ao PHP utilizando-se biblioteca própria, além de ser fácil de instalar e configurar.

O hardware escolhido para hospedar a aplicação foi o Raspberry Pi Modelo B equipado com servidor web Apache. O Raspberry Pi opera com uma versão de Linux embarcado, o que possibilita simples processos de instalação e configuração do Apache, MySQL e PHP. Os processos de instalação e configuração podem ser facilmente encontrados na web.

Como parte do processo de desenvolvimento, foi criado um cenário para o Virtualclicker. Cenário é uma simples história que descreve uma situação hipotética para levantar questionamentos e fazer o grupo de desenvolvedores pensar como o projeto proposto irá funcionar em situações adversas. Utilizar cenários durante o processo de desenvolvimento é uma boa prática uma vez que o desenvolvimento de software vai além da escrita do código [Rosson and Carroll 2001]. O estudo de caso a seguir ilustra uma aplicação para o

Virtualclicker em uma situação real, quando a rede sem fio local já está estabelecida e as configurações prévias estão funcionando.

Cenário:

O professor está em sala de aula após a chegada dos alunos. O professor entra na interface de instrutor, escolhendo uma aula para iniciar. Uma nova reunião é criada no banco de dados e os alunos estão prontos para começar. Os alunos podem fazer o *check in* na aula criada e o evento é registrado no banco de dados. Enquanto os estudantes estão entrando no sistema, o professor pode observar a presença dos alunos através da tela de presenças. Depois disso, durante a aula, o professor pode iniciar uma nova questão para verificar o entendimento dos alunos sobre o conteúdo apresentado. Depois da tela de presenças, o instrutor vai a tela de questões, que permite que ele crie uma nova questão através da escolha do número da pergunta. Uma vez que a questão está aberta no banco de dados, os alunos podem responde-la na interface do estudante.

O aluno vai a tela de resposta em que ele possui uma questão de múltipla escolha. O estudante pode inserir sua resposta depois de clicar nela, ou ele pode mudar sua resposta antes de finalizá-la. Enquanto os estudantes estão respondendo as questões, o professor pode observar as respostas na tela de respostas. Depois que todos já responderam, instrutor pode finalizar a questão e gerar um gráfico de setores mostrando as estatísticas das respostas da turma. Na última tela, o instrutor pode voltar a tela de inserção de questões para reiniciar o processo. A qualquer momento, o professor pode fechar a aula clicando no botão de saída. As interfaces de aluno e de professor compartilham o mesmo banco de dados, então a comunicação entre elas é feita através dele.

O protótipo do Virtualclicker é dividido em duas interfaces independentes: a interface do instrutor e a interface do aluno. Cada uma possui características distintas pensadas para suprir as necessidades de professores e estudantes com ferramentas específicas. A Figura 1 mostra os protótipos preliminares para as duas interfaces.

A interface do aluno é otimizada para trabalhar em uma tela pequena de um smartphone. Nós não esperamos que o aluno tenha uma tela grande, então as ferramentas desenvolvidas para a interface do aluno são mais simples e não ocupam muito espaço. Porém, a interface do aluno ainda encaixa-se perfeitamente na tela de um notebook. Uma vez que o instrutor possui ferramentas como tela de presenças, respostas e gráficos mostrando o desempenho do aluno, ela foi otimizada para trabalhar em uma tela maior de um tablet ou notebook, encaixando-se perfeitamente a uma tela de 10 polegadas.

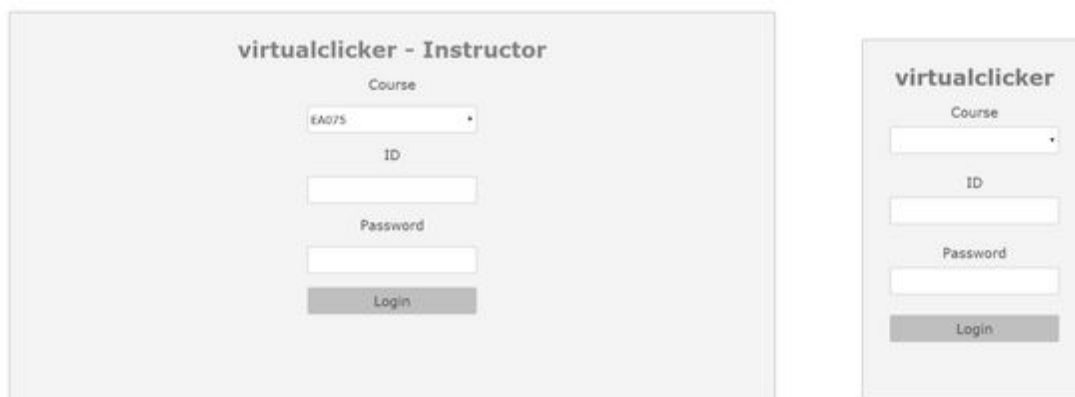


Figure 1. (A) Instructor interface (B) Learner interface

3. Resultados

Durante o desenvolvimento do projeto, a interface do usuário foi uma preocupação constante para a equipe de desenvolvimento. Cada peça de design foi pensada para ser simples e intuitiva para o usuário. O projeto é pensado para ser utilizado em áreas menos desenvolvidas, então não podemos esperar grande conhecimento sobre tecnologia dos nossos usuários. Para o sucesso da aplicação, nós tivemos que torná-la simples e de rápida aprendizagem.

O protótipo foi desenvolvido da forma mais intuitiva. A equipe de pesquisadores realizou um completo processo de *brainstorm* de forma a elaborar a *storyboard* do Virtualclicker tanto para a interface do aluno quanto para a do professor. Como resultado, nós pudemos desenvolver uma interface amigável e intuitiva.

No final do projeto, o sistema completo foi embarcado no Raspberry Pi, a rede local foi configurada e distribuída através de um roteador sem fio, e o protótipo pode ser testado com usuários reais. Assim, pudemos receber um *feedback* pela experiência deles. Os experimentos foram conduzidos da seguinte maneira.

Primeiramente, o sujeito interpretou o papel do professor e testou a interface do instrutor enquanto um membro da equipe de pesquisa interpretou o papel do estudante. O sujeito possuía um roteiro para seguir requisitando que ele trabalhasse com o protótipo. O sujeito era solicitado a inserir uma nova aula no sistema, esperar que o estudante entrasse, inserir uma nova questão, esperar que o estudante respondesse a pergunta e gerar um gráfico contendo a resposta do estudante.

Na segunda parte do experimento, o sujeito era solicitado a interpretar o papel do estudante e testar a interface do aluno enquanto um membro da equipe do projeto interpretava o papel do professor. Novamente, o sujeito deveria seguir um roteiro para interagir com o sistema. O sujeito era solicitado a entrar no sistema, responder uma pergunta proposta pelo professor, alterar a resposta e finalizar sua participação na aula. Durante todo o experimento, o sujeito deveria avaliar a experiência e pensar em como melhorar a interface.

Ao final dos experimentos de usuário, nós solicitamos que o sujeito participasse de uma pequena entrevista para relatar sua experiência de tal forma que pudéssemos obter uma

opinião subjetiva a respeito do projeto. Em seguida, solicitamos que o participante respondesse um formulário mais objetivo a respeito do que poderia ser melhorado no futuro do projeto.

Das respostas dos usuários, pudemos concluir que a plataforma é boa e simples de ser utilizada, entretanto pode ser melhorada de forma a tornar-se mais intuitiva. A plataforma possui fluxo único, assim, os usuários reclamaram que era difícil desfazer uma ação indesejada. Além disso, os usuários encontraram dificuldades durante o fluxo do programa para inserir questões durante a aula.

Ao final do experimento, a equipe do projeto percebeu que o protótipo em desenvolvimento funciona bem e não apresenta grandes problemas técnicos durante a execução. Entretanto, a equipe do projeto possui um grande trabalho para melhorar a experiência do usuário e tornar a plataforma mais intuitiva.

4. Discussão e Conclusão

Como um projeto em desenvolvimento, a equipe possui um grande trabalho futuro tornando a interface mais amigável para o usuário. É importante melhorar alguns aspectos de design e torná-la mais intuitiva. O *feedback* recebido através dos experimentos de usuário serão muito úteis no futuro do Virtualclicker.

Além disso, será necessário desenvolver uma interface offline para que o professor possa visualizar o desenvolvimento dos alunos fora da sala de aula, inserir novos alunos e novas turmas no sistema web e trabalhar com os dados adquiridos a qualquer momento. O trabalho futuro inclui tornar a plataforma mais flexível e robusta, melhorar aspectos de design e experiência de usuário e desenvolver diferentes tipos de questões, respostas e gráficos.

O Virtualclickers é parte de uma ampla pesquisa sobre interação homem-computador que inclui outra solução para sistemas de resposta a audiência em sala de aula chamada Paperclickers. Essa abordagem utiliza conceitos de processamento de imagens para capturar as respostas dos alunos identificando *top codes* na sala de aula.

A proposta do Paperclickers mantém a ideia do BYOD, mas nesse caso o professor é responsável por levar o dispositivo para a sala de aula ao invés do aluno. O instrutor utilizará a câmera do smartphone para escanear a sala de aula, então ele irá visualizar os resultados na tela do dispositivo. Para reconhecer o código impresso, será necessário instalar uma aplicação móvel no smartphone do professor.

A aplicação Paperclickers foi desenvolvida para reconhecer dois tipos de símbolos codificados. O primeiro símbolo usado é um código de barras bidimensional chamado QRCode. Nós escolhemos o QRCode por sua popularidade e uso crescente. QRCode possui um bom decodificador de código aberto chamado Zxing. Esse código pode armazenar números e caracteres alfabéticos, e ainda possui capacidade de correção de erros. Utilizando esse código, garantimos privacidade ao estudante e identificação única. Além disso, existem três padrões quadrangulares localizados no topo-esquerdo, topo-direito e inferior-esquerdo do código possibilitando a identificação da orientação o que proporciona respostas de múltipla escolha [Belussi and Hirata 2013]. O segundo símbolo é conhecido como TopCodes (*Tangible Object Placement Codes*). TopCodes foram desenvolvidos para reconhecer objetos tangíveis em uma superfície. Esse código proporciona identificação e orientação únicas, mas

podem reconhecer apenas 99 códigos. A biblioteca principal dos TopCodes é gratuita e de código aberto, e possui classes de Java disponíveis [TopCode 2015].

TopCodes e QrCodes possuem vantagens complementares. Enquanto QrCodes permitem a criação de identificadores únicos por estudante para um grande número de alunos, TopCodes permitem apenas 99 códigos diferentes, demandando um código único por turma, não por aluno. Apesar disso, TopCodes são extremamente robustos e de rápida detecção, enquanto QrCodes possuem problemas por conta de transformações geométricas e possuem requisitos computacionais caros para detecção.

A Figura 2 mostra a aplicação Paperclickers para definições preliminares e visualização de gráficos.

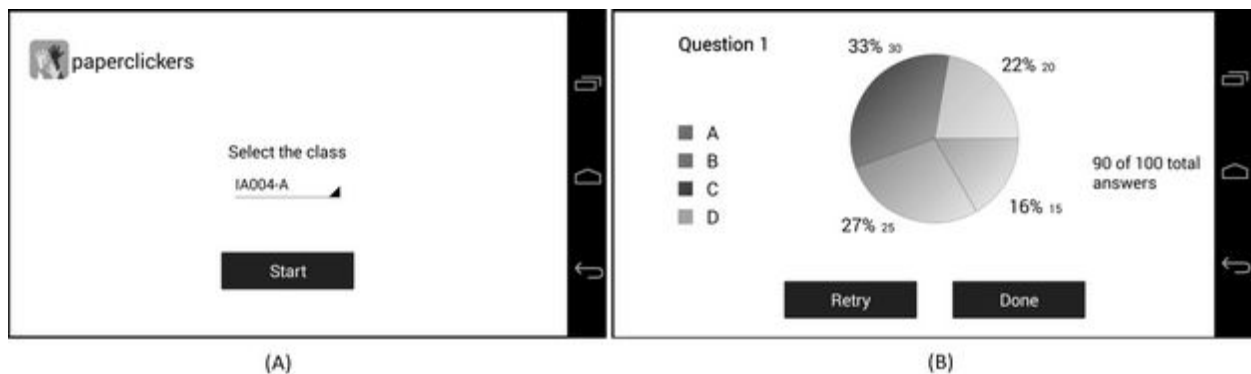


Figure 3. (A) Preliminary Definition screen; (B) GraphView screen.

O projeto do Paperclickers foi desenvolvido pela estudante de mestrado Jomara Mota Bindá, membro da equipe de pesquisadores e desenvolvedores. Após todo o projeto de pesquisa, as duas abordagens atingiram as expectativas propostas. Virtualclicker e Paperclickers mostram-se soluções promissoras para os problemas encontrados com sistemas de resposta a audiência.

5. Referências Bibliográficas

- Fulford, C. P. and Zhang, S. (1993) "Perceptions of Interaction: The critical predictor in distance education," *American Journal of Distance Education*, vol. 7. pp. 8–21.
- Kay, R. H. and LeSage, A. (2009) "Examining the benefits and challenges of using audience response systems: A review of the literature," *Comput. Educ.*, vol. 53, pp. 819–827.
- Cross, A. Cutrell, E. and Thies, W. (2012) "Low-cost audience polling using computer vision," *Proc. 25th Annual ACM Symp. User interface Software Technology - UIST '12*, p. 45.
- Mazur, E. (1997) "Peer Instruction: Getting Students to Think In Class," *The Changing Role of Physics Departments in Modern Universities: Proceedings of ICUPE*. pp. 981–988.
- Beatty, I.D. Gerace, W. J. Leonard, W. J. and Dufresne, R. J. (2006) "Designing effective questions for classroom response system teaching," *Am. J. Phys.*, vol. 74, no. 1, p. 31.
- Fies, C. and Marshall, J. (2006) "Classroom response systems: A review of the literature," *Journal Scientific Education Technology*, vol. 15, pp. 101–109.

Caldwell, J. E. (2007) “Clickers in the large classroom: Current research and best-practice tips,” CBE Life Sciences Education, vol. 6. pp. 9–20.

Yourstone, S. a Kraye, H. S. and Albaum, G. (2008) “Classroom Questioning with Immediate Electronic Response: Do Clickers Improve Learning?,” Decis. Sci. J. Innov. Educ., vol. 6, no. 1, pp. 75–88.

Bergtrom, G. (2006) “Clicker Sets as Learning Objects Clickers Promote Learning,” Interdisciplinary Journal Knowledge Learning Objects, vol. 2, pp. 106–110.

Beatty, I. D. (2005) “Transforming student learning with classroom communication systems,” Aug. 2005.

Cutts, Q. I. (2006) “Practical Lessons from Four Years of Using an ARS in Every Lecture of a,” pp. 65–79.

Stavert, B. (2013) “BYOD in Schools Literature Review” p. 34.

Lennon, R. G. (2012) “Bring your own device (BYOD) with Cloud 4 education,” Proc. 3rd Annual Conference Syst. Program. Applied Software Humanit. - SPLASH '12, p. 171.

Rosson, M. B. and Carroll, J. M. (2001) Usability engineering: scenario-based development of human-computer interaction. Elsevier.

Raspberry Pi Documentation (Online) – Acessível: <http://www.raspberrypi.org/>.

Data-Driven Documents (D3.js) Documentation (Online) – Acessível: <http://d3js.org/>.

PHP Documentation (Online) – Acessível: http://php.net/manual/pt_BR/.

MySQL Documentation (Online) – Acessível: <http://dev.mysql.com/doc/>.

W3Schools, SQL References (Online) – Acessível: <http://www.w3schools.com/sql/>.

Niederauer, Juliano, “PHP para quem conhece PHP: recursos avançados para criação de websites dinâmicos“, 3rd ed. – São Paulo: Novatec, 2008.

Welling, Luke, “PHP e MySQL desenvolvimento web” / Luke Welling, Laura Thompson; translation Edson Furmankiewicz e Adriana Kramer. – Rio de Janeiro: Elsevier, 2005 – 6th reprint.

QRCode (Online) - Acessível: <http://www.qrcode.com/en/>

Zxing (Online) - Acessível: <https://github.com/zxing/>

8. Agradecimentos

Essa pesquisa é financiada pelo PIBITI/CNPq e foi desenvolvida no Laboratório LCA/FEEC/UNICAMP. Agradecemos ao Prof. Dr. Renato R. Lopes e à estudante de mestrado Jomara Mota Bindá pelas importantes contribuições para o trabalho. Também agradecemos ao estudante de graduação Wilson José Prata Neto pela ajuda na concepção e condução dos experimentos de usuário.