

## **UMA PROPOSTA DE REDE SOCIAL UBÍQUA BASEADA EM REDES P2P MÓVEIS**

**Cristiano Bacelar de Oliveira<sup>(1)</sup>**

Graduando em Tecnologia em Telemática e Técnico em Telecomunicações, ambos pelo Centro Federal de Educação Tecnológica do Ceará. cristianobac@hotmail.com.

**Iluska Amorim Gurgel do Amaral<sup>(1)</sup>**

Graduando em Tecnologia em Telemática pelo Centro Federal de Educação Tecnológica do Ceará. iluskaamorim@yahoo.com.br.

**Fernando Augusto Matos de Almeida<sup>(1)</sup>**

Graduando em Tecnologia em Telemática pelo Centro Federal de Educação Tecnológica do Ceará. fernando\_aug@hotmail.com.

**Cidcley Teixeira de Souza<sup>(1)</sup>**

Professor da Gerência de Telemática do Centro Federal de Educação Tecnológica do Ceará. Coordenador do NASH (Núcleo Avançado em engenharia de Software distribuído e sistemas Hipermídia). E-mail: cidcley@cefetce.br.

**Endereço<sup>(1)</sup>:** Centro Federal de Educação Tecnológica - CEFETCE. Av. 13 de Maio, 2081 - Benfica - Fortaleza/CE. CEP:60040-531. Fone +55 (85) 3307-3642 Fax (85) 3307-3666

### **RESUMO**

As redes sociais estão atualmente se tornando cada vez mais presentes no cotidiano dos que utilizam a Internet. Contudo, a implementação dessas redes se dá, basicamente, através de servidores centralizados que devem ser utilizados através de navegadores Web. Neste trabalho descrevemos uma proposta de implementação de uma rede social que utiliza computação ubíqua. Esta rede baseia-se em contextos de localização e conteúdo, no qual os participantes podem compartilhar contextos diferentes de modo dinâmico e, por conseguinte, estar em contato com outras pessoas que tenham os mesmos interesses. Nesse trabalho são mostrados os cenários de utilização deste modelo de rede, além da descrição da estrutura e do funcionamento das entidades envolvidas.

**PALAVRAS-CHAVE:** Rede social, computação ubíqua, computação ciente de contexto, computação pervasiva e P2P móvel.

### **1. INTRODUÇÃO**

A computação móvel tem tido um crescimento bastante considerável, tanto em relação ao número de usuários adeptos, como na melhoria técnica e aumento de recursos dos dispositivos móveis. Hoje em dia é cada vez mais comum usarmos um *handheld* ou celular para tarefas como acesso à internet, edição de arquivos, etc.

A computação ubíqua [7] se assemelha bastante a computação distribuída, sendo uma forma de computação distribuída em redes sem fio. Porém a grande diferença a computação ubíqua é que os programas estão embarcados no ambiente, conseguindo informações deste para determinar o seu comportamento, tendo o conhecimento do dispositivo em que o mesmo está executando e de suas capacidades.

A computação ubíqua nos permite fazer uso constante da interligação de computadores, porém de uma forma um pouco diferente das aplicações comuns em redes de computadores, onde vários programas trocam dados através da rede um programa utiliza a rede para obter diversas informações sobre o ambiente e utilizando essas informações para determinar sua execução.

No contexto da Internet, uma aplicação que vêm se desenvolvendo no sentido de integração e compartilhamento de informações são as redes sociais. Uma rede social [8] é uma das formas de representação dos relacionamentos afetivos ou profissionais dos seres humanos entre si ou entre seus agrupamentos de interesses mútuos. Essa rede é responsável pelo compartilhamento de idéias entre pessoas que possuem

interesses e objetivo em comum. Assim, um grupo de discussão é composto por indivíduos que possuem identidades semelhantes.

Contudo, uma das grandes desvantagens dessas redes sociais é que essas são implementadas atualmente de forma estática. Ou seja, interessaem que surgem em virtude de necessidades momentâneas deveriam permitir a criação de comunidades dinâmicas. Além disso, o princípio da localização física não é considerado nas implementações atuais de redes sociais. Nesse sentido, propomos nesse trabalho, uma nova estrutura de rede social que utilizando os conceitos presentes na computação ubíqua, permite a criação de uma rede social móvel e orientada a contexto. Esta rede funcionará como uma série de comunidades, onde cada uma é definida por um contexto específico e limitada a uma determinada região, criando assim uma relação baseada, também, na localização de cada usuário.

Na nossa proposta, denominada de Rede Social Ubíqua (RSU), serão formados grupos de discussões, onde a localização e a orientação à contexto reservará determinado assunto a uma área restrita, onde realmente só existam usuários em busca de algo em comum.

Este artigo está dividido em quatro partes principais. Na seção 2 traremos alguns conceitos envolvidos na computação ubíqua. Na seção 3 são apresentados alguns conceitos relacionados às redes sociais. Já na seção 4, serão abordadas as funcionalidades da Rede Social Ubíqua, com a apresentação de cenários e detalhes sobre seu funcionamento. Na seção 5 é apresentado um protótipo de implementação da RSU. Por fim, na seção 6, temos uma breve conclusão.

## **2. COMPUTAÇÃO UBÍQUA**

Os avanços da eletrônica, principalmente o surgimento da tecnologia de VLSI, trouxe também um grande avanço computacional, destacando-se o aumento da velocidade dos processadores e o aumento da capacidade das memórias. Esta evolução aliada à evolução das telecomunicações, principalmente das redes sem fio permitiu o surgimento de novos paradigmas computacionais, entre estes se destaca a computação ubíqua. A computação ubíqua integra a mobilidade das redes sem fio à execução dos softwares, trazendo a computação distribuída para as redes sem fio[7]. Dessa forma a computação move-se para fora dos computadores, podendo migrar entre eles, escolhendo qual computador apresenta as melhores características para a execução de um determinado programa. Contudo, o conceito de computação ubíqua é um pouco mais complicado e é uma fusão de características da computação móvel e da computação pervasiva.

A computação móvel consiste em mover serviços computacionais, porém os serviços não mudam nem interagem de forma alguma com o ambiente, de fato os programas não têm idéia dessa movimentação. Um exemplo seria o uso de programas em dispositivos móveis.

Também nessa mesma linha, a computação pervasiva é utilizada o conceito de software cliente de contexto. Onde os programas conseguem identificar o ambiente no qual estão inseridos e baseados nesse ambiente o comportamento do software muda para melhor se adequar às características do ambiente e às necessidades do usuário. É importante também que se detecte outros computadores inseridos nesse ambiente, podendo-se escolher a máquina que melhor executa uma determinada aplicação.

Já a computação ubíqua consiste no conceito da computação pervasiva aplicada em dispositivos móveis, sendo uma junção da computação pervasiva com a computação móvel[3].

## **3. REDES SOCIAIS**

As redes sociais [8] assemelham-se às redes de computadores em alguns aspectos. Dentre eles podemos destacar o fato principal de ambas serem formadas por nós que estão interligados a fim de compartilharem algum recurso. No caso das redes sociais, os indivíduos representam os nodos que compartilham uma informação, através de algum tipo de relacionamento entre si. Para isso, no entanto, é preciso um meio de comunicação que atinja a todos que estão dentro da mesma rede.

Embora tenham se popularizado com a Internet, devido justamente à facilidade da troca de informações através da Web, as redes sociais já são objeto de estudo de várias pesquisas na área das Ciências Humanas há algum

tempo. Todavia, essas redes que antes eram limitadas por questões geográficas, éticas ou culturais, dentre outras, são hoje bem mais abrangentes e, consequentemente, mais influentes, diante da perspectiva de que hoje uma informação viaja pelo mundo todo em poucos segundos. Com isso, temos pessoas querendo cada vez mais informação em cada vez menos tempo. Essa tendência é refletida de, ou para, as redes de computadores.

A RSU tem por objetivo atuar como um meio que facilite a troca de informações entre seus usuários, permitindo a busca constante por assuntos de interesses comuns entre eles, agilizando contatos e, principalmente, acompanhando o seu deslocamento.

#### **4. REDE SOCIAL UBÍQUA**

Neste trabalho é proposta uma aplicação que faça uso da computação ubíqua para formar uma rede social, em que pessoas com interesses em comuns possam manter contato. A idéia de computação ubíqua foi dividida em contextos, onde pessoas inseridas nos mesmos contextos formam uma rede social. Cabe a aplicação identificar os contextos no qual o usuário está inserido num dado momento, e se modificar para atender suas necessidades.

A modificação da aplicação consiste basicamente em filtrar usuários que possuem contextos diferentes, bem como inserir novos usuários que possuem os mesmos contextos. Assim a aplicação vai inserindo e excluindo usuário da rede formada pelos contextos em que o usuário da aplicação está inserido, bem como determinando os contextos em que o próprio usuário está inserido.

Para esta aplicação foram vislumbradas duas formas de contexto. O contexto de conteúdo, que identifica usuários que tenham interesses em um conteúdo em comum, e o contexto de localização, que identifica usuários que estão em uma mesma área geográfica de localização. A rede social é formada por usuário que estão inseridos ao mesmo tempo nos mesmos contextos de conteúdo e localização.

Com a Rede Social Ubíqua (RSU), um apaixonado por Mustang, por exemplo, que está em um museu de carros e que sonha em fazer uma exposição com Mustangs, pode abrir um grupo em que o contexto de conteúdo seria Mustangs e já começar a organizar a lista com os interessados. Em uma outra situação, um fazendeiro que esteja na exposição de "Arte e Ciência no Instituto Biológico", em São Paulo, aplicada ao estudo de pragas e doenças na agricultura e que tem sua plantação de mandioca empestada de insetos e ácaros pode criar um grupo referente ao assunto e obter recomendações de pessoas que se encontram próximas e que já passaram pelo mesmo problema. Ou então, alguém que está no shopping e que deseja conhecer críticas a respeito de um filme em cartaz, pode abrir um tópico com o assunto citado e trocar mensagem com os usuários que têm acesso ao serviço, que estão no mesmo contexto de localização e que se interessarem em emitir opiniões. Como já foi dito, trata-se de uma espécie de rede móvel de relacionamentos, onde os tópicos, ou grupos, criados se compararam às comunidades e seus membros estarem localizados dentro do mesmo raio de alcance.

Em termos de arquitetura, a RSU é uma rede P2P móvel [10], que consiste num conjunto de *peers* que estão distribuídos na rede e interagem no compartilhamento de informações. Na nossa proposta há a presença de um servidor centralizado utilizado apenas para o registro do serviço por parte dos *peers*. Entretanto, este servidor não faz parte efetivamente da rede de comunicação entre os *peers*, uma vez que esta é limitada pelos contextos dinâmicos em que os usuários estão inseridos. Nas próximas seções são detalhadas as arquiteturas de software tanto do servidor como dos *peers*.

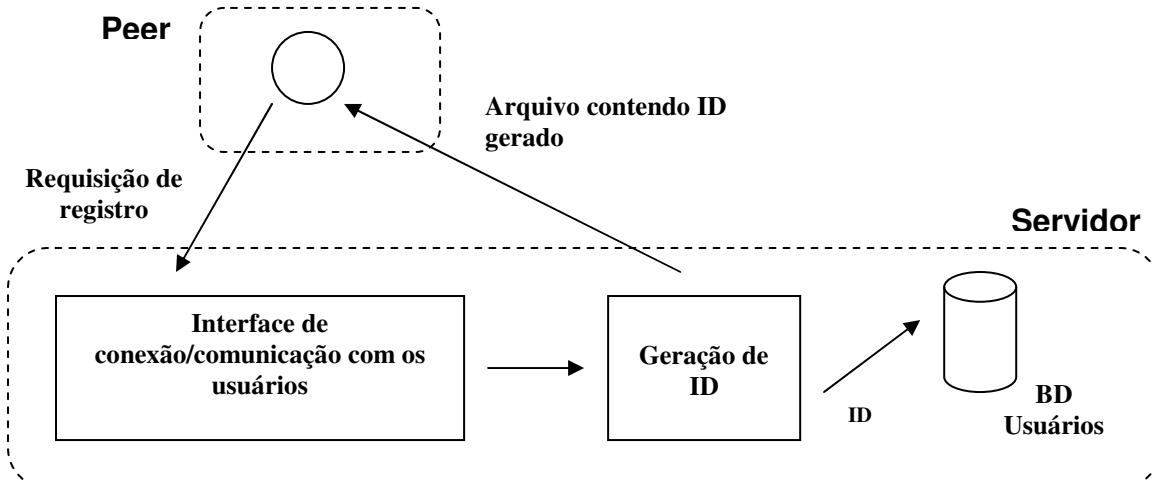
##### **4.1 ARQUITETURA DE SOFTWARE DO SERVIDOR**

Na proposta da RSU, o servidor realiza o controle escalar do serviço, ou seja, ele permite o controle de *peers* que entram e saem da rede, assim como realiza o gerenciamento dos os contextos que foram criados, além de proporcionar um certo nível de autenticação, uma vez que os *peers* são reconhecidos pela aplicação em função do ID, gerado por esse servidor.

Em termos de software, o servidor pode ser dividido da seguinte forma:

- a. **Interface de conexão:** Módulo responsável pelo monitoramento das requisições e envio das respostas. A resposta consiste basicamente da informação de ID único do usuário em um arquivo formatado, ou alguma mensagem de não reconhecimento da solicitação. Este ID é fornecido uma única vez ao usuário, sendo função da aplicação dos *peers* armazená-lo. É importante que este servidor esteja disponível a todos que tenha interesse no serviço, logo, preferimos optar por um site que possua uma interface gráfica simples o suficiente para ser bem visualizada por qualquer dispositivo conectado a web. É através deste site que o usuário tem acesso a fazer as requisições e o download do arquivo de registro.
- b. **Geração de ID:** Após o servidor receber uma requisição de registro, ele precisará gerar a informação de ID e sobre o arquivo a ser enviado. Este ID deve ser único e, para evitar que seja gerado em duplicidade, possui uma lei de formação baseada na data e hora (tempo real) em que o servidor está processando o pedido. Uma vez gerado o código ID, este é gravado em um arquivo que será transmitido ao usuário. Se a transmissão houver ocorrido com sucesso, o servidor então atualiza sua base de dados, adicionando a ela o ID enviado.
- c. **BD de usuários:** O servidor armazena todos os IDs criados em uma base dados. Como citado anteriormente, isto se faz importante para controle e acompanhamento do número de usuários cadastrados. Outras informações, como, por exemplo, horário do registro, também podem ser armazenadas para análises estatísticas futuras.

**Figura 1: Solicitação e geração de registro de usuário.**



#### 4.2 ARQUITETURA DE SOFTWARE DOS PEERS

Existem três elementos principais participantes de uma aplicação na RSU, como apresentado na Figura 2: O Gerente de contexto, responsável pela criação do contexto dinâmico e gerenciamento dos usuários dentro do contexto. O Gerente de conexão, responsável pela gerência de entrada e saída de usuário registrados do raio de alcance do dispositivo e o Registro, que foi apresentado na seção anteriormente. Os dois primeiros atuam como módulos independentes em um mesmo dispositivo, ou seja, cada participante da rede gerencia sua conexão e seus contextos. Em suma, o software do *peer* possui as seguintes funções:

##### a. **Gerente de conexão:**

Como existe a idéia de contexto de localização, que consiste no raio de alcance de um *peer*, a aplicação deve ficar monitorando a presença de outros usuários. O gerenciamento da entrada/saída de *peers* do raio de alcance de um dispositivo é feito pelo gerente de conexão. Uma vez que os *peers* são móveis, esta tarefa deve ser executada periodicamente, tendo-se cuidado de não sobrecarregar a rede. Existem duas formas de efetuar este gerenciamento. A primeira consiste na aplicação manter uma lista atualizada de

usuários visíveis, fazendo o processo de busca a atualização de forma automática. A outra forma seria a aplicação realizar essa busca sempre que o usuário desejar se comunicar com alguém, verificando, então, quais os *peers* disponíveis no momento da comunicação.

No primeiro caso, há a vantagem da transparência dessa operação para o usuário, no entanto passamos a ter o problema de aumento do tráfego desnecessário na rede. Imaginando que existem cerca de 50 usuários em um raio aproximado, haveriam então 50 dispositivos trocando mensagens de busca constantemente. Se este número aumentar, consequentemente o congestionamento da rede passa a ser maior. Realizando essa busca de forma manual, perde-se na questão da transparência, mas existe a melhoria em relação ao congestionamento, uma vez que qualquer busca só é feita no momento efetivo da comunicação. Em ambos os casos deve haver uma busca inicial, realização na inicialização do aplicativo, a fim de ambientar o usuário com os *peers* disponíveis no momento que o serviço é iniciado.

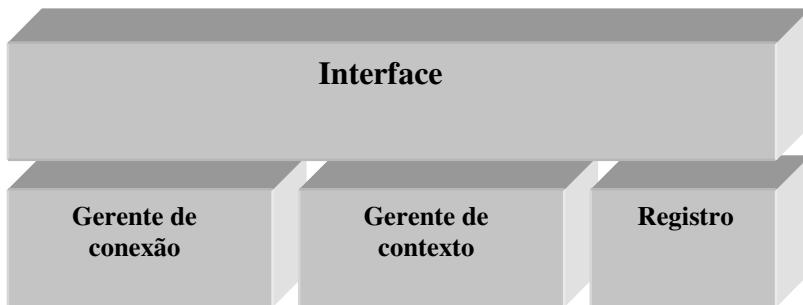
Cada *peer* localizado tem seu ID validado para confirmar seu registro. Caso tenha um ID válido, são armazenados o seu ID, sua localização, seu nick e sua lista de contextos. A validação do ID consiste apenas na verificação do formato do ID recebido. Em nenhum momento o servidor é consultado sobre o ID.

#### **b. Gerente de contexto:**

Cada ID (de usuário) pode fazer parte simultaneamente de até três contextos (com ID's de contextos diferentes), cabe então ao gerente de contexto controlar o acesso aos contextos de interesse de um *peer*. Embora existam apenas três contextos ativos por vez, existe uma fila de contextos disponíveis, ficando a critério do usuário escolher quais contextos ele quer ativar ou desativar, obedecendo ao limite especificado. Há ainda a possibilidade da criação de um novo contexto, ou grupo de interesse, por parte do usuário. Esse contexto é criado dinamicamente e obedece a um modelo padrão de informações necessária à sua criação. A aplicação gera o ID de um contexto a partir da informação do ID do usuário, o que garante que cada contexto possua um ID único, mesmo com os demais criados em *peers* diferentes. Da mesma forma que nas comunidades virtuais, quando um usuário cria um contexto ele passa a ser seu administrador, tendo alguns privilégios administrativos sobre o contexto. O gerente de contexto é responsável, também, por manter a hierarquia de usuários de um mesmo contexto.

A mobilidade dos *peers* também afeta a lista de contextos. À medida que novos usuários ficam disponíveis, surgem novos contextos inseridos à fila, e vice-versa. Daí percebe-se uma separação natural de contexto por localização. O gerente de contexto, portanto, depende de informações do gerente de conexão para atualizar sua lista de contextos disponíveis.

**Figura 2: Blocos funcionais de softwares dos *peers*.**



### **4.3 FUNCIONAMENTO**

O registro no serviço é feito em um servidor de registros, cuja função principal é gerar um ID para o usuário que deseja se registrar e manter uma base de usuários atualizada. Para fazer parte de uma RSU, o usuário deve primeiro se cadastrar no serviço e receber do servidor de registros um ID único. De posse de um ID o usuário pode iniciar o serviço, de forma independente do servidor. Daí em diante tem-se uma rede P2P, onde os usuários participantes estão dentro de um mesmo contexto de interesse e localização.

Para iniciar o serviço, o usuário deve informar o local onde se encontra e ativar (iniciar/abrir) um contexto, ou seja, criar novos contextos, além de poder também optar por buscar outros dentro do seu raio de alcance. O resultado da busca é uma lista com os contextos disponíveis e seus respectivos proprietários. O administrador do contexto é o usuário que está a mais tempo com o contexto ativo, isso garante uma certa hierarquia, independente da entrada/saída de novos usuários do raio de alcance.

No caso de contextos privados, para fazer parte de um novo grupo, o usuário precisa ser aceito pelo administrador do contexto. O usuário interessado deve enviar uma solicitação ao gerente para ingressar no contexto desejado. Caso seja aceito, o gerente do contexto adiciona-o a uma lista dos usuários ativos dentro do contexto. Já contextos públicos podem ser acessados livremente. Outra opção é a busca de usuários participantes do mesmo grupo. Essa busca é feita pelo gerente de conexão, retornando uma lista de usuários ativos. Para iniciar a comunicação, basta escolher um usuário disponível em um dos contextos ativos e enviar-lhe uma mensagem.

#### **4.4 MODELO DE CONTEXTO**

Neste artigo, admi-se uma forma de orientação a contexto diferenciada da visão padrão, onde muitas vezes utiliza-se uma plataforma de *middleware* para realizar a adaptação por parte da aplicação[1]. Neste caso, tem-se o contexto limitando o relacionamento entre usuários, sendo definido pelo interesse e localização, e restringido por escolha do usuário e pela tecnologia de rede utilizada. Assim, a aplicação acaba por interagir apenas com quem está inserido no mesmo contexto. É importante ressaltar que, para perfis de dispositivos diferentes, a lógica da aplicação continua a mesma, apenas com alteração em sua interface. Cada contexto possui as seguintes informações:

1. **ID:** Identificador do contexto. Gerado a partir do ID do criador.
2. **Descrição:** Nome do contexto.
3. **Administrador:** ID do usuário criador do contexto.
4. **Status:** Indica se o contexto está ativo ou em fila.
5. **Tempo ativo:** Tempo de atividade do usuário no contexto.
6. **Tipo:** Público ou privado.

#### **4.5 FORMATO DAS INFORMAÇÕES DE USUÁRIO**

A aplicação mantém dois tipos de registro de usuários: o próprio usuário da aplicação e os demais que estão em seu raio de alcance. Ambos os tipos possuem a mesma formatação, a diferença está no tratamento destas informações por parte dos gerentes de contexto e conexão. As seguintes são as informações de usuário:

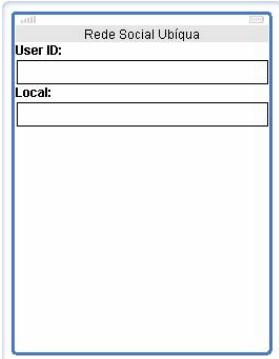
1. **ID:** Identificador do usuário fornecido pelo servidor de registro.
2. **Nome:** “apelido” do usuário, fornecido por ele na primeira vez que aplicação é usada, logo após a efetivação do registro.
3. **Local:** Local geográfico onde se encontra o *peer*. Esta informação é informada pelo usuário no momento da inicialização da aplicação. Sua utilidade é dar uma idéia aos demais usuários de onde se encontra o *peer*.

### **5. IMPLEMENTAÇÃO DA APLICAÇÃO**

A validação dessa proposta está sendo realizada através da implementação de um protótipo baseado em PC. Esse protótipo foi desenvolvido com J2ME *Wireless Toolkit* [9] e permite e avaliação dos protocolos e das restrições relacionadas à interfaces. No que diz respeito às interfaces, a RSU implementa as seguintes telas principais:

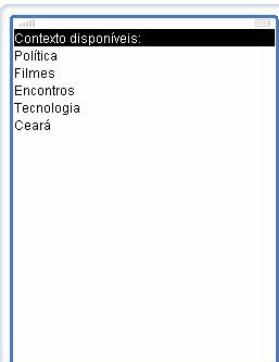
1. **Inicialização e registro:** Tela inicial do sistema. É nesta tela que o usuário pode se registrar no serviço e informar seu nick. Caso já possua ID e nick, basta informar o local onde se encontra. A Figura 3 apresenta a implementação dessa interface.

**Figura 3: Interface de login do RSU.**



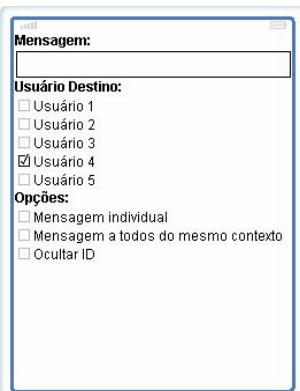
2. **Lista geral de usuários disponíveis:** Contém uma lista dinâmica de todos os usuários disponíveis dentro do raio do dispositivo.
3. **Lista de usuários disponíveis no contexto:** Contém uma lista dinâmica de todos os usuários disponíveis dentro de um contexto.
4. **Lista de contextos:** Tela contendo a relação de contextos (ativos e fila) de um *peer*. A Figura 4 apresenta a implementação dessa interface.

**Figura 4: Interface de listagem de contextos.**



5. **Detalhes do contexto:** Diversas informações sobre o contexto selecionado.
6. **Criação de contexto:** Tela onde o usuário pode criar um novo contexto, baseado no modelo especificado.
7. **Troca de mensagens:** Tela onde ocorre a comunicação com outros usuários. A Figura 5 apresenta a implementação dessa interface.

**Figura 5: Interface de troca de mensagens.**



Essas implementações ainda estão em uma fase inicial, contudo, nos testes realizados, estamos conseguindo observar a dinamicidade introduzida pela RSU e, principalmente, a versatilidade dessa proposta no que se trata de configuração de interesses e compartilhamento de informações com pessoas que realmente estejam interessados no tema em um momento e em lugar específico. Fator não contemplado atualmente por nenhuma outra proposta de rede social.

## **6. CONCLUSÃO E TRABALHOS FUTUROS**

As redes sociais são uma realidade atualmente. Elas congregam milhares de pessoas em todo o mundo ao redor do compartilhamento de interesses e de perfis. Nesse trabalho apresentamos uma nova abordagem para a implementação de redes sociais. Nossa proposta, denominada de RSU (Rede Social Ubíqua) utiliza o conceito de computação ubíqua para permitir a criação de comunidades dinâmicas e temporárias e que utilizem os conceitos de localização física e contextos de interesses.

De um modo geral, a RSU aborda uma série de conceitos, principalmente ligados a P2P móvel e computação ciente de contexto. Isto nos permite dizer que ainda há muito o que ser discutido e pensado sobre a implementação prática de uma RSU estável e popular. Portanto, nesse trabalho apresentamos a arquitetura básica da RSU. Estamos realizando uma implementação de teste e em breve uma versão será disponibilizada para avaliação. Contudo, acreditamos que essa proposta possui um caráter prático bastante interessante e que deve ser facilmente absorvida, principalmente pela população jovem que usa largamente tecnologias como Orkut e MySpaces.

Como trabalhos futuros, que na verdade já estão sendo realizado, temos como prioridade portar a implementação para dispositivos móveis reais, de modo que seja mais fácil avaliar mais precisamente as questões de mobilidade, além da exploração de uma rede com grande número de usuários e de grande número de usuários.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

1. ROCHA, Ricardo Couto Antunes da, e ENDLER, Markus, (2006) “Supporting Context-Aware Applications: Scenarios, Models And Architecture”, PUC-RJ.
2. SCHILIT, Bill N., ADAMS Norman, , and WANT, Roy .“Context-Aware Computing Applications”
3. ARAÚJO , Regina Borges de. “Computação Ubíqua: Princípios, Tecnologias e Desafios”. Universidade Federal de S. Carlos (UFSCar),Caixa Postal 676 – 13565-905 – São Carlos - SP – Brasil
4. GUIMARÃES,Eliane G., CARDOZO, Eleri, Magalhães, MAURÍCIO F., GOMES WANDER P., Pinto, Rossano P., Faina, Luís F. “CCM-tel – uma Plataforma para Aplicações Telemáticas e Ubíquas”.
5. CAMPIOLO ,Rodrigo .(2005). “Aspectos de Modelagem de Ambientes de Computação Ubíqua”. Universidade Federal de Santa Catarina – UFSC.
6. THOMSON,G., Richmond, Matthew e Nixon, Sotirios Terzisand Paddy.“An Approach to Dynamic Context Discovery and Composition”. University of Strathclyde Livingstone Tower. Glasgow,UK.
7. SOUSA, Fernando A. “Frameworks e Middleware para Computação Ubíqua”. NUSP:3286224. MAC 5743 – Introdução à Computação Móvel.
8. SILVA, Antonio Braz de Oliveira e; MATHEUS, Renato Fabiano; PARREIRAS, Fernando Silva; PARREIRAS, Tatiane Aparecida Silva. Análise de redes sociais como metodologia de apoio para a discussão da interdisciplinaridade na Ciência da Informação . Ciência da Informação . Brasília: v.35, n.1, 1º sem. 2006.
9. J2ME. <http://java.sun.com/products/sjwtoolkit/>. Acessado em Julho de 2007.
10. Kato, T., Ishikawa, N., Sumino, H., Hjelm, J., Yu, Y., and Murakami, S. A platform and applications for mobile peer-to-peer communications. 2003.