

# Infraestrutura para o Desenvolvimento de Aplicações Pervasivas Cientes de Redes Sociais

Daniel Santos<sup>1</sup>, Angelo Perkusich<sup>2</sup>, Hyggo O. Almeida<sup>1</sup>

<sup>1</sup>Departamento de Sistemas e Computação  
Universidade Federal de Campina Grande (UFCG)  
58429-140 – Campina Grande – PB – Brazil

<sup>2</sup>Departamento de Engenharia Elétrica  
Universidade Federal de Campina Grande (UFCG)  
58429-140 – Campina Grande – PB – Brazil

{danielbruno, perkusich, hyggo}@embedded.ufcg.edu.br

**Resumo.** Neste artigo apresentamos uma infraestrutura para o desenvolvimento de aplicações pervasivas para Redes Sociais Móveis, que disponibiliza os seguintes serviços: (i) aquisição e persistência das informações de contexto social e de localização, obtidas dos dispositivos móveis dos usuários e das redes sociais online; e (ii) API para execução dos serviços e acesso às informações de contexto persistidas. Como contribuição principal, também é disponibilizado um módulo para processamento distribuído das conexões sociais, representando os amigos diretos e indiretos dos usuários. A avaliação do trabalho constituiu-se do desenvolvimento de um estudo de caso e realização de experimentos que demonstram o suporte da infraestrutura para o desenvolvimento de aplicações e a escalabilidade do módulo proposto.

**Abstract.** In this paper we present an infrastructure to development of pervasive applications for Mobile Social Networking, which offers the following services: (i) acquisition and persistence of social and location context information, gathered from users' mobile devices and online social networks, and (ii) API to perform the services and to access stored context information. As main contribution is also made available a mechanism for distributed processing of users' social connections, which represents the direct and indirect friendship connections of users. The evaluation of the work consisted of developing a case study and performing experiments that demonstrate the infrastructure utility and the scalability of the proposed module.

## 1. Introdução

Nos últimos anos, tem-se observado a popularização do acesso à Internet, em conjunto com um extraordinário avanço das tecnologias de comunicação sem fio e a massificação dos dispositivos móveis no mercado consumidor, principalmente, *notebooks*, *netbooks*, aparelhos celulares e *smartphones*. Os dois últimos, em especial, estão se tornando cada vez mais baratos, acessíveis e com uma grande gama de recursos disponíveis, tais como câmera digital, vídeo, GPS, acelerômetro, conexões *Wi-Fi*, 3G, *Bluetooth*, dentre outros. Consequentemente, tem-se tornado cada vez mais comum as pessoas estarem sempre

portando seus dispositivos móveis, passando grande parte do tempo conectadas à Internet, em qualquer lugar, a qualquer momento. Esse cenário tem-se mostrado propício à viabilização do paradigma de Computação Pervasiva [Weiser 1991] e o desenvolvimento de aplicações para esse domínio.

Esse avanço no paradigma de Computação Pervasiva motivou o surgimento de uma área denominada de Redes Sociais Móveis ou Redes Sociais Pervasivas [Ben Mokhtar and Capra 2009], que estuda a integração entre este paradigma com as redes sociais da web 2.0. Esta sinergia possibilita aos usuários, que compartilham interesses em comum e frequentam os mesmos lugares no dia-a-dia, interagirem entre si para realizarem atividades de interesse comum, em qualquer lugar, a qualquer momento [Churchill and Halverson 2005, Jones and Grandhi 2005].

Para que esse paradigma funcione plenamente é primordial que as aplicações recuperem as informações do ambiente no qual os usuários se encontram, com a finalidade de prover serviços e executar tarefas em favor deles [Satyanarayanan 2001, Saha and Mukherjee 2003, Loureiro et al. 2007]. Para atingir esse objetivo, as aplicações devem possuir ciência de contexto [Dey 2001].

Aplicações pervasivas para Redes Sociais Móveis possuem ciência de pelo menos duas informações de contexto relacionadas aos usuários: (i) o contexto social (*social-awareness*); e (ii) o contexto de localização (*location-awareness*). O primeiro contém a informação sobre quem são os amigos do usuário e quais são as suas preferências ou interesses. Já o último, contém a informação sobre onde esses amigos estão localizados, a partir da aplicação de técnicas de localização [Küpper 2005].

Uma vez de posse das informações de contexto social e de localização dos usuários, essas aplicações são capazes de perceber a proximidade dos amigos, membros da comunidade, ou de alguém cujo perfil possua um alto grau de similaridade ao do usuário, em tempo real. Com isso, as aplicações podem pró-ativamente alertar o usuário através de seu dispositivo móvel, ou ainda, recomendar o novo contato como potencial amigo, caso ele ainda não faça parte da lista de amigos do usuário. Alguns exemplos de aplicações para Redes Sociais Móveis são: Serendipity [Eagle and Pentland 2005], WhozThat [Beach et al. 2008], Google Latitude [Google 2009], dentre outras.

A fim de possibilitar o desenvolvimento de aplicações pervasivas cientes de Redes Sociais, faz-se necessário o desenvolvimento de infraestruturas, *middlewares* e *frameworks* que lidem com alguns desafios, tais como: a heterogeneidade dos dispositivos móveis, tolerância a falhas nas redes, segurança e privacidade das informações dos usuários, dentre outros [da Costa et al. 2008, Saha and Mukherjee 2003]. Além disso, essas soluções devem: (i) abstrair o processo de comunicação com outras aplicações, (ii) permitir a aquisição, persistência e reutilização das informações de contexto social e de localização dos usuários e (iii) disponibilizar APIs para que os desenvolvedores possam acessar essas informações e construir rapidamente novas aplicações sociais [Bottazzi et al. 2007, Gupta et al. 2009, Toninelli et al. 2010].

Esse artigo propõe uma infraestrutura para o desenvolvimento de aplicações pervasivas no domínio de Redes Sociais Móveis. O artigo está organizado da seguinte forma: na Seção 2 são apresentadas as abordagens existentes para o desenvolvimento de aplicações pervasivas para Redes Sociais Móveis; na Seção 3 é apresentada a arqui-

tutura da infraestrutura desenvolvida, assim como, a sua API para execução dos serviços e acesso às informações de contexto disponibilizadas. Na Seção 4 é apresentado o estudo de caso desenvolvido, utilizando-se a API fornecida pela infraestrutura. Na Seção 5 são apresentados os resultados dos experimentos realizados no *Módulo de Expansão das Conexões Sociais*. Por fim, na Seção 6 são descritas as conclusões e os trabalhos futuros.

## 2. Trabalhos Relacionados

Nos últimos anos, têm surgido diversas soluções de *middlewares* para o desenvolvimento de aplicações para Redes Sociais Móveis, [Bottazzi et al. 2007, Pietiläinen et al. 2009, Ben Mokhtar and Capra 2009, Gupta et al. 2009, Low et al. 2010, Boldrini et al. 2010]. Essas soluções tentam capturar as informações dos dispositivos móveis, para construir os contextos sociais e de localização dos usuários e formar Redes Sociais Móveis.

As soluções de *middleware* existentes utilizam duas abordagens distintas para formação de Redes Sociais Móveis, são elas: (i) formação de comunidades móveis através da descoberta *Ad Hoc* dos dispositivos móveis dos usuários [Eagle and Pentland 2005, Bottazzi et al. 2007, Beach et al. 2008, Ben Mokhtar and Capra 2009, Pietiläinen et al. 2009, Gupta et al. 2009, Low et al. 2010, Boldrini et al. 2010], que compartilham interesses ou preferências em comum e/ou realizam atividades em um mesmo espaço físico; e (ii) utilização das conexões sociais existentes nas redes sociais online [Miluzzo et al. 2008, Pietiläinen et al. 2009].

Abordagens como [Gupta et al. 2009, Pietiläinen et al. 2009] apresentam modelos sintáticos para representação do perfil de pessoas e lugares. Especificamente, [Pietiläinen et al. 2009] utiliza um modelo com base nas informações de perfil dos usuários do Facebook. Outras soluções como [Bottazzi et al. 2007, Ben Mokhtar and Capra 2009, Toninelli et al. 2010] apresentam abordagens semânticas, baseadas em ontologias de domínio, *Friend of a Friend (FOAF)* e *Resource Description Framework (RDF)*, respectivamente, para modelar os interesses dos usuários e suas atividades. O trabalho apresentado em [Toninelli et al. 2010] permite a modelagem das interações sociais envolvendo conteúdos digitais produzidos pelos usuários.

Também, soluções como [Bottazzi et al. 2007, Gupta et al. 2009, Pietiläinen et al. 2009] apresentam funcionalidades que extraem e persistem as informações de contexto social e de localização dos usuários do ambiente. Todavia, apenas [Gupta et al. 2009, Pietiläinen et al. 2009] fornecem APIs para acesso às informações de contexto persistidas e suporte ao desenvolvimento de aplicações sociais.

Ainda, algumas soluções encontradas na literatura apresentam técnicas e algoritmos para extrair as informações de contexto social e de localização, com o objetivo de inferir relações sociais [Li and Chen 2009] e analisar padrões de utilização dos dispositivos móveis [Eagle and (Sandy) Pentland 2006, Eagle et al. 2009].

O serviço de *middleware* proposto em [Gupta et al. 2009] disponibiliza um módulo de aprendizagem de contexto social que utiliza algoritmos para formação de grupos sociais e descoberta de padrões geo-sociais previamente desconhecidos, tais como afinidades entre pessoas e entre pessoas e lugares. A solução desenvolvida em [Quercia et al. 2010] para construção de listas de recomendação também utiliza as informações sobre os encontros entre os usuários em conjunto com algoritmos de ex-

pansão social. Todavia, os algoritmos utilizados pelas soluções anteriores não fornecem conhecimento sobre os níveis de relacionamentos (diretos e indiretos) entre os usuários.

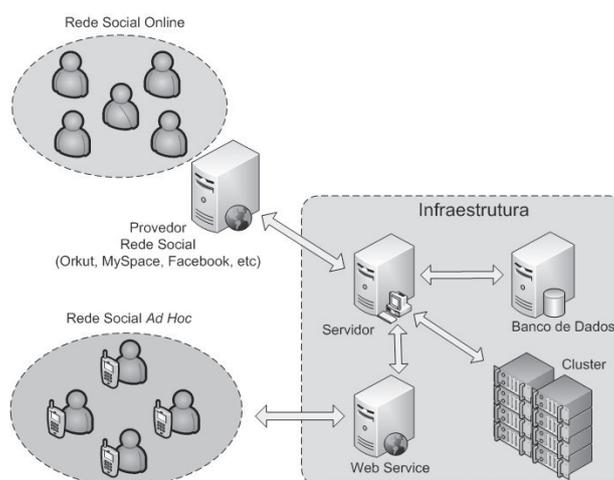
Por fim, iniciativas como OpenSocial<sup>1</sup> e Social Graph API<sup>2</sup> disponibilizam um conjunto de APIs, para que aplicações sociais acessem as informações dos perfis das redes sociais online e as conexões sociais presentes nos diferentes serviços sociais existentes na web, respectivamente.

Apesar de existir uma grande variedade de soluções de *middlewares* para o desenvolvimento de aplicações pervasivas para Redes Sociais Móveis, elas ainda não fornecem todas as funcionalidades necessárias para o desenvolvimento de aplicações nesse domínio. Especificamente, não há uma abordagem isolada que forneça as seguintes funcionalidades: (i) suporte à execução de serviços e acesso às informações de contexto social e de localização dos usuários; (ii) integração das abordagens *Ad Hoc* e de redes sociais online; (iii) conhecimento prévio sobre quem são os amigos diretos e indiretos dos usuários, e; (iv) conhecimento em tempo real da localização dos usuários. Portanto, este trabalho apresenta uma infraestrutura para o desenvolvimento de aplicações para Redes Sociais Móveis, que disponibiliza o conjunto de funcionalidades enumeradas anteriormente.

### 3. Projeto da Infraestrutura

A infraestrutura foi desenvolvida com base em uma Arquitetura Orientada a Serviços (SOA), fornecendo facilidade de acesso aos serviços disponibilizados, heterogeneidade e interoperabilidade entre as diversas plataformas de desenvolvimento disponíveis.

Especificamente em relação à construção de Redes Sociais Móveis, conforme ilustrado na Figura 1, a infraestrutura possibilita a integração com as redes sociais online e com as redes sociais *Ad Hoc* (oriundas da interação oportunista entre os dispositivos móveis dos usuários), através dos modos de funcionamento online e offline<sup>3</sup>, respectivamente.



**Figura 1. Visão Geral da Infraestrutura**

<sup>1</sup><http://code.google.com/apis/opensocial/>

<sup>2</sup><http://code.google.com/intl/pt-BR/apis/socialgraph/>

<sup>3</sup>Embora o modo offline faça parte da especificação arquitetural da infraestrutura, a sua apresentação não faz parte do escopo deste trabalho.

As próximas subseções descrevem detalhadamente cada um dos componentes envolvidos na arquitetura da infraestrutura.

### **3.1. Modo online com suporte à construção de Redes Sociais Móveis**

O modo online disponibilizado pela infraestrutura baseia-se na premissa de que as tecnologias de acesso sem fio são pervasivas, ou seja, estão presentes em qualquer lugar, a qualquer momento e estão embarcadas nos dispositivos móveis disponíveis no mercado consumidor. Dessa forma, os usuários possuem acesso constante à Internet através de seus dispositivos móveis. Este modo de funcionamento tem como objetivo principal realizar a integração com as redes sociais online, a fim de utilizar as informações existentes nessas redes para enriquecer as informações locais, armazenadas na infraestrutura, sobre os membros das Redes Sociais Móveis.

O modo online funciona a partir de um servidor central, que utiliza uma conexão com a Internet para recuperar as informações dos perfis dos usuários das redes sociais online (e.g., preferências pessoais, lista de amigos, dentre outras informações). As informações extraídas das redes sociais online são persistidas pela infraestrutura e auxiliam na criação do contexto social dos usuários e formação de Redes Sociais Móveis.

A fim de obter acesso às informações dos perfis dos usuários das redes sociais online a partir da infraestrutura, utilizou-se uma abordagem onde o usuário, mediante instalação em seu perfil de uma aplicação disponibilizada pelo desenvolvedor da infraestrutura, autoriza explicitamente o acesso às suas informações.

Neste trabalho, utilizou-se a API OpenSocial para acesso às informações de perfil das redes sociais online. OpenSocial fornece um conjunto de APIs para construção de aplicações sociais para redes sociais online. Uma vez que essas aplicações estão instaladas no perfil dos usuários nas redes sociais online, elas fornecem um mecanismo de acesso, às informações de perfil dos usuários, para o desenvolvedor da aplicação. Ainda, são fornecidas APIs do lado do servidor para interação com a aplicação instalada no perfil do usuário. Para isso, o servidor deve fornecer ao provedor da rede social online as credenciais da aplicação instalada no perfil do cliente.

### **3.2. Módulo responsável pela Expansão das Conexões Sociais**

Alguns resultados, como aqueles obtidos em [Quercia et al. 2010], demonstram que há uma maior probabilidade de um usuário vir a formar um novo vínculo de amizade com alguém próximo ao seu ciclo de amizades (e.g., com um amigo de um amigo em comum), do que com um usuário completamente desconhecido (e.g., usuário recomendado a partir de um algoritmo de casamento de perfis ou recomendação).

Motivado pelo resultado anterior, foi desenvolvido um *Módulo de Expansão das Conexões Sociais*, que possui como objetivo calcular os níveis de relacionamento diretos e indiretos entre todos os membros de uma Rede Social Móvel e armazenar essas informações em um mecanismo de armazenamento persistente, para que elas sejam utilizadas posteriormente por outros módulos da infraestrutura.

O módulo proposto modela os relacionamentos sociais dos usuários através de uma representação de grafos. Nesse modelo, os vértices representam os usuários e as arestas os relacionamentos existentes entre eles. Ainda, as arestas são não-direcionadas,

uma vez que o relacionamento de amizade entre dois usuários é considerado mútuo ou bidirecional pela infraestrutura. Além disso, as arestas do grafo podem ser ponderadas com peso igual a 1, ou ainda, podem ser representadas sem a utilização de pesos.

Especificamente, o *Módulo de Expansão das Conexões Sociais* utiliza um algoritmo que soluciona um problema clássico da Teoria dos Grafos, denominado *problema de caminhos mais curtos de todos os pares*. De acordo com [Cormen et al. 2001], dentre as possíveis soluções para esse problema destacam-se os algoritmos de *Johnson*, *Floyd-Warshall* e da *elevação ao quadrado repetida em matrizes*, com complexidades  $O(V^2lgV + VE)$ ,  $\Theta(V^3)$  e  $\Theta(V^3lgV)$ , respectivamente. Ainda segundo [Cormen et al. 2001], outra alternativa é aplicar um algoritmo que soluciona o *problema de caminhos mais curtos de única origem* para todos os vértices do grafo. Algumas soluções para esse problema são os algoritmos de *Bellman-Ford*, *Dijkstra* e *Breadth-First Search (BFS)*, com complexidades  $O(V^2E)$ ,  $O(V^2lgV + VE)$  e  $O(V^2 + VE)$  (se o grafo estiver conectado a complexidade será  $O(VE)$ ), respectivamente.

Entretanto, uma vez que na representação escolhida as arestas não são ponderadas, o algoritmo BFS mostra-se uma boa alternativa para solucionar o problema anterior. Essa solução apresenta uma complexidade assintoticamente melhor do que os algoritmos de *Dijkstra* e *Bellman-Ford* e diferentemente das demais soluções, pode ser facilmente adaptado para que a travessia de cada vértice no grafo seja limitada pela quantidade de níveis a serem descobertos. Dessa forma, o *Módulo de Expansão das Conexões Sociais* processa no máximo seis níveis de amizade para todos os vértices do grafo.

A escolha de processar no máximo seis níveis foi realizada com base na *Teoria dos Seis Graus de Separação* [Kleinberg 2000], que afirma que são necessários em média seis laços de amizade para que duas pessoas quaisquer no mundo estejam conectadas entre si. Ainda, alguns estudos como aqueles realizados em [Kumar et al. 2006] demonstraram que essa teoria pode ser aplicada para as Redes Sociais Online, ao encontrarem um valor médio para o caminho mínimo entre 5 e 7, para quaisquer pares de vértices distintos pertencentes às amostras dos grafos das redes sociais analisadas.

O resultado obtido como saída da execução do algoritmo, pode ser utilizado para indicar a distância no grafo entre dois pares de vértices. Ou ainda, na modelagem utilizada, o nível de relacionamento existente entre dois usuários. Dessa forma, o nível 1 (i.e. distância igual a 1 entre dois pares de vértices) significa que os usuários são amigos diretos e os demais níveis (i.e. distâncias maiores ou iguais a 2 entre dois pares de vértices), indicam que os usuários estão relacionados indiretamente (e.g., através de amigos em comum na passagem de um nível para outro).

Por fim, com base nos resultados obtidos após execução do algoritmo, a infraestrutura possibilita a formação de novos vínculos de amizade entre usuários que compartilham interesses em comum e que estão localizados próximos uns dos outros. Com isso, evoluindo as informações de contexto social dos usuários.

### 3.3. API para acesso às informações da Infraestrutura

Na Tabela 1 estão sumarizados os serviços fornecidos pela API da infraestrutura, disponibilizada através do *Web Services* contido na Figura 1. Detalhes sobre os parâmetros de entrada e o retorno de cada serviço estão disponíveis em: <http://embedded.ufcg.edu.br/~danielbruno/SBRC2011/>.

**Tabela 1. API da Infraestrutura.**

<b>Serviço</b>	<b>Descrição</b>
addFriendship()	Adiciona um novo contato à lista de amigos do usuário
registerUser()	Registra um novo usuário
registerLocation()	Registra a localização do usuário
getMembersFrom()	Lista os membros da Rede Social Móvel
getUserInfo()	Recupera as informações sobre o usuário
getUserPreferences()	Lista os interesses e/ou preferências do usuário
getFriendsFrom()	Lista os amigos do usuário
getFriendsAtLevel()	Lista os amigos do usuário para o nível especificado
getSocialNetworksFrom()	Lista as redes sociais que o usuário faz parte
getCurrentLocation()	Recupera a informação sobre a última localização do usuário
getNearByUsers()	Lista os usuários que estão próximos

#### 4. Estudo de Caso: Aplicação MySocial

Nessa seção descrevemos o suporte da infraestrutura ao desenvolvimento de uma aplicação social móvel baseada em localização, denominada MySocial. Essa aplicação utiliza as informações de contexto social e de localização dos usuários, disponíveis através da API da infraestrutura, para: (i) construir listas de contatos, contendo os amigos diretos e indiretos das redes sociais online que os usuários participam e (ii) informar aos usuários quando esses contatos estiverem próximos.

##### 4.1. Simulação do Grafo das Conexões Sociais

Com o objetivo de simular as conexões sociais dos usuários das Redes Sociais Online e manipular estruturas de dados representadas por grafos, foi utilizada a biblioteca em Java JGraphT<sup>4</sup>. Essa biblioteca apresenta classes que constroem grafos de acordo com a topologia escolhida pelo usuário, a quantidade de vértices e de arestas passadas como parâmetros. Para o exemplo utilizado nesta seção foi construído um grafo representando a topologia *Scale-Free* [Mislove et al. 2007], que caracteriza as Redes Sociais. O grafo construído constitui-se de 100 mil vértices e um conjunto de arestas representando os usuários do sistema e os relacionamentos existentes entre eles, respectivamente.

##### 4.2. Processamento do Grafo das Conexões Sociais

O grafo gerado, na subseção anterior, é processado pelo *Módulo de Expansão das Conexões Sociais* presente na Infraestrutura. De acordo com as informações descritas previamente na seção 3.2, esse módulo tem como objetivo executar um algoritmo que soluciona o *problema de caminhos mais curtos de todos os pares*. Dessa forma, após execução do algoritmo, espera-se obter como resultado, para cada usuário presente no grafo, uma tabela *hash* [Cormen et al. 2001]. Onde, as chaves (valores inteiros entre 1 e 6) representam a menor distância entre o vértice inicial e os demais vértices no grafo.

No contexto deste trabalho, o resultado obtido pode ser interpretado da seguinte maneira: para cada tabela *hash* gerada, suas chaves armazenam os níveis de relacionamento existente entre o usuário inicial e os demais usuários do sistema. Onde, o nível

<sup>4</sup>Disponível em: <http://www.jgrapht.org/>

1 contém os amigos diretos do usuário (i.e. os vizinhos do vértice inicial) e os demais níveis contém os amigos indiretos do usuário (i.e. os vértices alcançáveis a partir do vértice inicial cuja distância seja maior ou igual a dois).

### 4.3. Obtenção dos dados para Simulação

O objetivo desta etapa foi o de obter um conjunto de coordenadas geográficas para simular a mobilidade dos usuários no sistema. Para atingir este objetivo foram escolhidos dois valores iniciais para latitude e longitude (valores referentes à localização do Laboratório Embedded), para compor as coordenadas geográficas que servem de base para obtenção das demais. Com base nessas coordenadas, aplicou-se uma variação equivalente a cinco metros no valor da latitude, fixando-se o valor da longitude. Esse procedimento foi realizado novamente, dessa vez fixando-se o valor da latitude e variando-se o valor da longitude. Como resultado foram obtidas duas retas, contendo as coordenadas de latitude e longitude iniciais como ponto de intersecção entre elas, cujos pontos são equidistantes entre si, com uma distância de  $\pm 5$  metros entre o ponto anterior e o posterior.

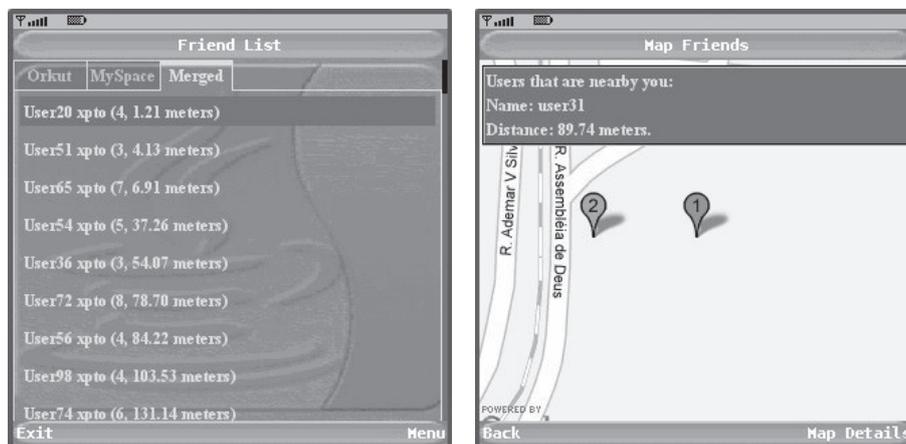
### 4.4. Execução do Estudo de Caso

Com base nas coordenadas geográficas obtidas anteriormente e de uma lista contendo os usuários cadastrados no sistema, aplicou-se o seguinte algoritmo para simular a movimentação dos usuários no sistema. Inicialmente, foram selecionados de forma aleatória, um usuário da lista e um par de coordenadas geográficas de latitude e longitude a cada segundo. Durante o processo de obtenção das novas coordenadas geográficas, o algoritmo verifica a última posição em que o usuário se encontrava, a fim de não produzir uma variação de movimento não permitida (e.g., o usuário em um instante de tempo estar localizado a uma distância de 200 metros das coordenadas iniciais e 1 segundo após estar a 1000 metros da distância anterior). Caso as coordenadas selecionadas sejam inválidas, o algoritmo seleciona uma nova localização para o usuário. Por fim, de posse das informações selecionadas, o algoritmo registra através da API da infraestrutura o novo par de coordenadas geográficas para o usuário selecionado. Esse procedimento é repetido para os demais usuários do sistema indefinidamente, enquanto o algoritmo não tiver sua execução finalizada.

No lado da infraestrutura, as informações de localização dos usuários enviadas pelo simulador são recebidas através do *Web Services* e armazenadas na base de dados, sendo persistidas na tabela que armazena a localização dos usuários do sistema. Quando a aplicação MySocial requisita o serviço *getNearByUsers()*, através da API da infraestrutura, esta recupera as informações sobre os amigos diretos e indiretos do usuário (amigos de nível 1 e entre os níveis 2 e 6, respectivamente), que foram processadas previamente pelo *Módulo de Expansão das Conexões Sociais*, em conjunto com as informações sobre a localização desses usuários armazenadas na base de dados.

Do ponto de vista do usuário que está observando a aplicação MySocial sendo executada em seu *smartphone*, a cada minuto as informações da sua lista de contatos são atualizadas, com base nas informações obtidas a partir da API da infraestrutura. A *interface* gráfica do dispositivo móvel do usuário apresentará as seguintes informações: nome do contato, o nível de amizade e a distância em metros entre os dois usuários. Caso o usuário deseje visualizar a localização do contato, a aplicação apresentará através de um mapa a localização dos dois usuários, indicando a distância entre eles.

Os resultados obtidos são apresentados na Figura 2. Na Figura 2(a), o usuário pode visualizar os amigos diretos e indiretos que estão próximos, através de listas de contatos separadas pelas redes sociais online que o usuário faz parte. Já na Figura 2(b), o usuário pode visualizar, a partir de um mapa gerado pela aplicação, a sua distância em metros em relação a outro usuário presente na sua lista de contatos.



(a) Lista de contatos

(b) Visualização da localização dos usuários através de um mapa

**Figura 2. Telas da aplicação MySocial**

## 5. Análise Experimental

Nessa seção, realizamos uma avaliação da escalabilidade do *Módulo de Expansão das Conexões Sociais* disponibilizado pela infraestrutura.

### 5.1. Configuração do ambiente de execução

Para execução dos experimentos, montou-se um ambiente contendo 9 máquinas configuradas em *cluster*. O *cluster* disponível no Laboratório de Sistemas Embarcados e Computação Pervasiva (Embedded), da Universidade Federal de Campina Grande, apresenta a seguinte configuração: são 9 máquinas, cada uma contendo 2 processadores Intel (R) Xeon (R) x64 de 2,33 GHz, com 4 GB de memória DDR2 800 MHz, 2 HDs SATA 7500 RPM de 250 GB e 2 placas de rede Gigabit Ethernet.

Na configuração utilizada, um dos nós do *cluster* desempenha o papel do servidor e os oito restantes, denominados de clientes, realizam o processamento distribuído do grafo. O servidor possui as seguintes atribuições: (1) construir o grafo de acordo com a topologia e a quantidade de vértices requeridos, (2) gerenciar as conexões com os clientes, dividindo a carga de trabalho de forma proporcional à quantidade de clientes disponíveis (e.g., considerando oito clientes disponíveis, cada um receberá 1/8 do conjunto de vértices a serem processados), (3) gerenciar as estatísticas dos experimentos enviadas pelos clientes, após o término da execução de sua carga de trabalho e (4) persistir os dados processados na base de dados.

Cada um dos clientes recebe o grafo criado pelo servidor, em conjunto com uma lista de trabalho (i.e. um conjunto de vértices distintos a serem processados por cada cliente). De posse do grafo gerado e da lista de vértices a serem processados, os clientes

iniciam em paralelo o processamento do grafo para o conjunto de vértices sob sua responsabilidade. Durante o processamento, os clientes geram arquivos contendo o resultado do seu processamento. Esses arquivos serão persistidos na base de dados pelo servidor. Após o término do processamento de todos os vértices de sua lista de trabalho, os clientes enviam para o servidor as estatísticas referentes à execução de cada experimento.

## 5.2. Execução dos Experimentos

Inicialmente, definimos duas configurações de grafos para execução dos experimentos. Na primeira configuração, os grafos possuíam as seguintes características: 10 mil vértices, aproximadamente 16 mil arestas e topologia *Scale-Free*. Já na segunda configuração, os grafos possuíam as seguintes características: 100 mil vértices, aproximadamente 159 mil arestas e topologia *Scale-Free*. Para cada uma das configurações, foram realizadas 7 repetições nos experimentos, utilizando o *Módulo de Expansão das Conexões Sociais* para determinar o tempo necessário para processar apenas o primeiro nível e os seis primeiros níveis, para todos os vértices dos grafos.

Ainda, com o objetivo de avaliarmos o ganho de processamento ao distribuímos a carga de trabalho entre os nós do *cluster*, verificamos para cada uma das configurações definidas o tempo necessário para processar os grafos, utilizando apenas um dos nós do *cluster* e todos os nós disponíveis.

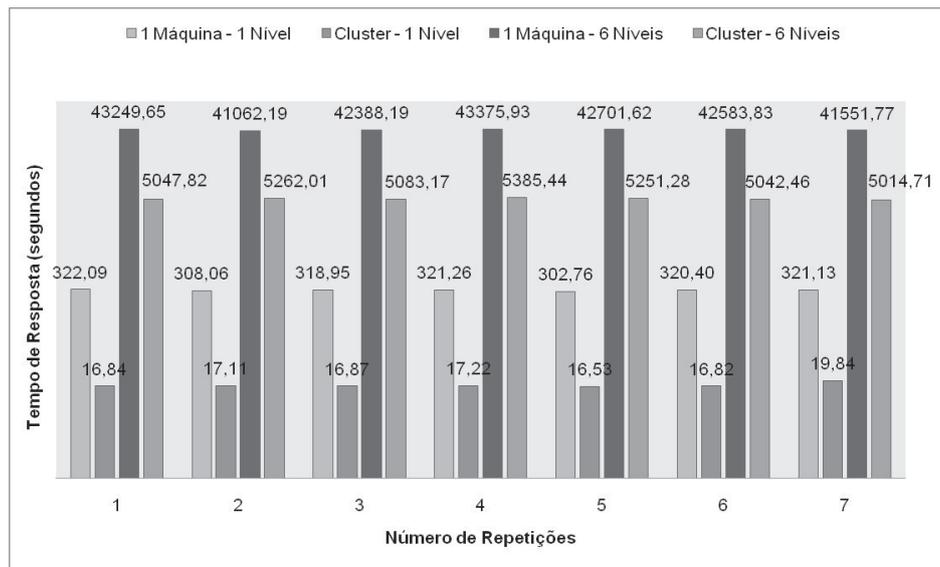
Por fim, com o objetivo de avaliarmos a escalabilidade do *Módulo de Expansão das Conexões Sociais*, adicionamos 10% e 50% de novas arestas aos grafos nas duas configurações, após o término da execução dos grafos originais. Dessa forma, pudemos aferir o impacto dessas atualizações no processamento do grafo pelo módulo.

Os resultados dos experimentos estão disponíveis em: <http://embedded.ufcg.edu.br/~danielbruno/SBRC2011/>

## 5.3. Apresentação e Análise dos Resultados

Analisando o gráfico da Figura 3, observa-se um crescimento exponencial no tempo de processamento dos grafos ao aumentarmos a quantidade de níveis a serem processados de 1 para 6, nas configurações contendo apenas 1 nó e os 8 nós do *cluster*. Esse resultado está fortemente relacionado à topologia *Scale-Free*, uma vez que grafos contendo essa topologia são caracterizados como esparsos, contendo uma pequena quantidade de arestas em relação a quantidade de vértices total. Portanto, quanto maior a quantidade de níveis a serem processados, mais tempo será necessário para processar grafos com essa topologia.

Ainda, com a utilização dos oito nós do *cluster* para o processamento do grafo contendo 100 mil usuários, obteve-se uma redução do tempo necessário para processar o primeiro nível do grafo (i.e. determinar apenas os vizinhos de todos os vértices) de aproximadamente vinte vezes, se comparado com o processamento resultante de apenas 1 nó do *cluster*. Quando levamos em consideração o tempo necessário para processar os seis primeiros níveis desse mesmo grafo, observa-se uma redução no tempo de processamento de aproximadamente oito vezes, em relação ao processamento resultante de apenas 1 nó. Esse resultado pode ser traduzido como uma diminuição do tempo de processamento de 12 horas para apenas 1 hora e 30 minutos, utilizando-se 1 nó e 8 nós do *cluster*, respectivamente.



**Figura 3. Análise comparativa do tempo de processamento de grafos contendo 100 mil usuários, utilizando o *cluster* com 1 nó e 8 nós, respectivamente.**

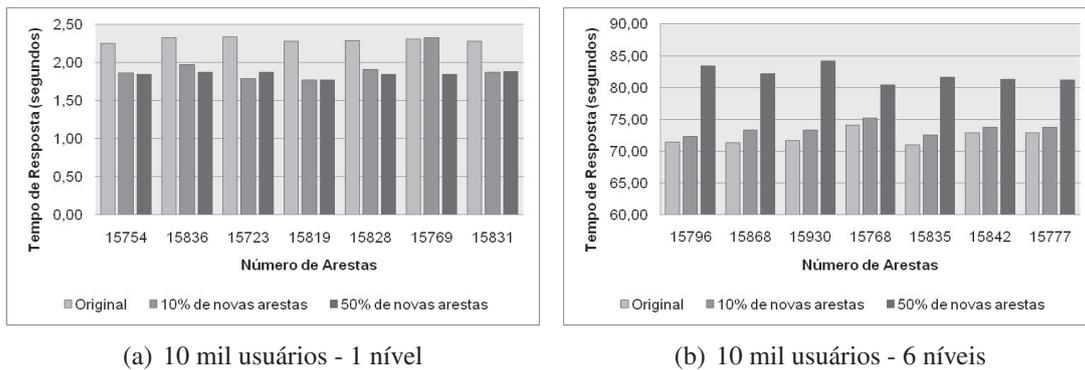
As Figuras 4 e 5 apresentam os resultados da nossa avaliação sobre a escalabilidade do *Módulo de Expansão das Conexões Sociais*. Especificamente, os resultados apresentados nos gráficos das Figuras 4(a) e 5(a) mostram que ao adicionarmos 10% e 50% de novas arestas aos grafos contendo 10 mil e 100 mil usuários, o tempo necessário para processar o nível 1 não aumenta. Ao contrário, o tempo de processamento torna-se menor em relação ao do grafo original. Este resultado deve-se a uma menor influência da topologia no processamento dos primeiros níveis do grafo.

Por outro lado, os resultados apresentados nos gráficos das Figuras 4(b) e 5(b) mostram que o módulo, após a adição das novas arestas ao grafo original, tende a demorar mais tempo para processar os seis primeiros níveis do grafo. Este último resultado pode ser observado pelo aumento do tempo necessário para processar os grafos contendo 10 mil e 100 mil usuários, respectivamente. Especificamente, os testes realizados nos grafos de 100 mil usuários, contendo 10% de novas arestas apresentaram um aumento no tempo de processamento de aproximadamente 30 minutos (33,3% do tempo de processamento do grafo original). Já os testes em grafos de 100 mil usuários, contendo 50% de novas arestas, mostram que o tempo de processamento torna-se quase duas vezes maior.

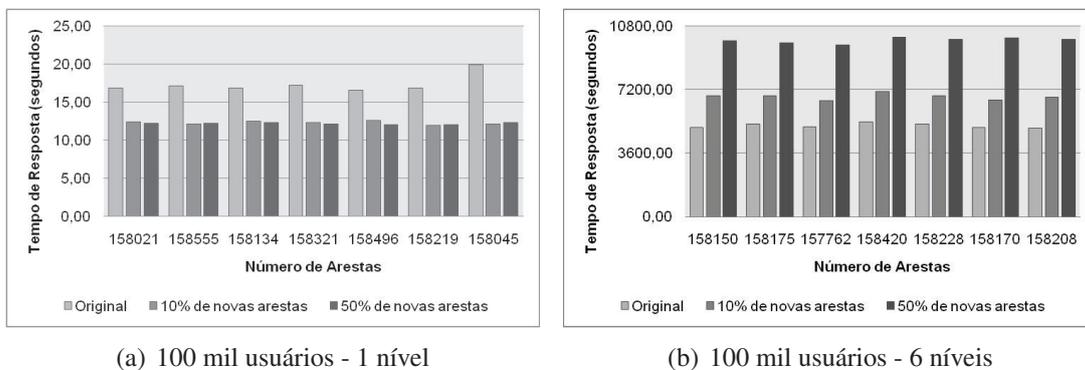
## 6. Conclusões e Trabalhos Futuros

Neste trabalho apresentamos uma infraestrutura que suporta o desenvolvimento de aplicações pervasivas cientes de Redes Sociais. Especificamente, através da utilização da API e dos serviços disponibilizados pela infraestrutura, desenvolvemos a aplicação MySocial, que se integra, através de *plugins*, às Redes Sociais Online (e.g., Orkut e MySpace) e com as Redes Sociais *Ad Hoc*. Com isso, mostramos o suporte da infraestrutura para o desenvolvimento de aplicações para Redes Sociais Móveis.

Além disso, avaliamos a escalabilidade do *Módulo de Expansão das Conexões Sociais* disponibilizado pela infraestrutura. Para isso, foram realizados experimentos em grafos com topologia análoga àquela presente nas Redes Sociais, com o objetivo de medir



**Figura 4. Tempo de processamento para um grafo com 10 mil usuários, após adição de 10% e 50% de novas arestas**



**Figura 5. Tempo de processamento para um grafo com 100 mil usuários, após adição de 10% e 50% de novas arestas**

o tempo de processamento necessário para determinar os seis primeiros níveis do grafo para todos os vértices. Os resultados demonstraram que, para as configurações avaliadas, o algoritmo utilizado pelo *Módulo de Expansão das Conexões Sociais* soluciona de forma satisfatória o *problema de caminhos mais curtos de todos os pares*.

Por fim, observou-se que ao aumentarmos a quantidade de vértices e arestas do grafo, sem variação da topologia, o tempo de processamento sofre um crescimento exponencial. Todavia, esse crescimento pode ser minimizado ao aumentarmos a quantidade de máquinas para o processamento do grafo. Por exemplo, nos nossos experimentos, o processamento de seis níveis em grafos contendo 10 mil e 100 mil usuários, respectivamente, utilizando um *cluster* com oito nós, foi cerca de oito vezes mais rápido do que o tempo de processamento de apenas um nó para as mesmas configurações. Esse último resultado fornece indícios de haver uma relação inversamente proporcional entre a quantidade de computadores e o tempo de processamento.

Como trabalhos futuros, nós iremos realizar uma análise estatística sobre os resultados obtidos, a fim de avaliarmos o impacto do conjunto de variáveis e fatores envolvidos no problema. Dessa forma, esperamos poder generalizar os resultados obtidos para grafos contendo uma quantidade de vértices semelhante àquelas encontradas nas Redes Sociais Online. Além disso, iremos avaliar a utilização do algoritmo de *Johnson* no *Módulo de Expansão das Conexões Sociais*. Uma vez que, de acordo com [Cormen et al. 2001], esse

algoritmo se mostra mais eficiente para o processamento de grafos esparsos.

## Referências

- Beach, A., Gartrell, M., Akkala, S., Elston, J., Kelley, J., Nishimoto, K., Ray, B., Razgulin, S., Sundaresan, K., Surendar, B., Terada, M., and Han, R. (2008). Whozthat? evolving an ecosystem for context-aware mobile social networks. *IEEE Network*, 22(4):50–55.
- Ben Mokhtar, S. and Capra, L. (2009). From pervasive to social computing: algorithms and deployments. In *ICPS '09: Proceedings of the 2009 international conference on Pervasive services*, pages 169–178, New York, NY, USA. ACM.
- Boldrini, C., Conti, M., Delmastro, F., and Passarella, A. (2010). Context- and social-aware middleware for opportunistic networks. *Journal of Network and Computer Applications*, 33(5):525–541.
- Bottazzi, D., Montanari, R., and Toninelli, A. (2007). Context-Aware Middleware for Anytime, Anywhere Social Networks. *IEEE Intelligent Systems*, 22(5):23–32.
- Churchill, E. F. and Halverson, C. A. (2005). Guest Editors' Introduction: Social Networks and Social Networking. *IEEE Internet Computing*, 9(5):14–19.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press, second edition.
- da Costa, C. A., Yamin, A. C., and Geyer, C. F. R. (2008). Toward a general software infrastructure for ubiquitous computing. *IEEE Pervasive Computing*, 7(1):64–73.
- Dey, A. K. (2001). Understanding and Using Context. *Personal Ubiquitous Computing*, 5(1):4–7.
- Eagle, N. and Pentland, A. (2005). Social Serendipity: Mobilizing Social Software. *IEEE Pervasive Computing*, 4(2):28–34.
- Eagle, N., Pentland, A. S., and Lazer, D. (2009). Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278.
- Eagle, N. and (Sandy) Pentland, A. (2006). Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268.
- Google (2009). Google Latitude. <http://www.google.com/latitude>. Último acesso em Janeiro 2010.
- Gupta, A., Kalra, A., Boston, D., and Borcea, C. (2009). Mobisoc: a middleware for mobile social computing applications. *Mobile Networks and Applications*, 14(1):35–52.
- Jones, Q. and Grandhi, S. (2005). P3 systems: putting the place back into social networks. *Internet Computing, IEEE*, 9(5):38 – 46.
- Kleinberg, J. (2000). The small-world phenomenon: an algorithm perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, STOC '00, pages 163–170, New York, NY, USA. ACM.

- Küpper, A. (2005). *Location-based Services: Fundamentals and Operation*. John Wiley & Sons.
- Kumar, R., Novak, J., and Tomkins, A. (2006). Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 611–617, New York, NY, USA. ACM.
- Li, N. and Chen, G. (2009). Multi-layered friendship modeling for location-based mobile social networks. In *Mobile and Ubiquitous Systems: Networking Services, MobiQuitous, 2009. MobiQuitous '09. 6th Annual International*, pages 1–10.
- Loureiro, E., Ferreira, G., Almeida, H., and Perkusich, A. (2007). Pervasive Computing: What is it Anyway? In Lytras, M. and Naeve, A., editors, *Ubiquitous and Pervasive Knowledge and Learning Management: Semantics, Social Networking and New Media to their full potential*, chapter 1, pages 9–36. Idea Group Inc, Hershey, PA, USA.
- Low, D. M.-k., Huang, R. K., Mishra, P., Jain, G., Gosnell, J., and Bush, J. (2010). Group Formation Using Anonymous Broadcast Information. <http://www.freepatentsonline.com/y2010/0070758.html>. United States Patent Application.
- Miluzzo, E., Lane, N. D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S. B., Zheng, X., and Campbell, A. T. (2008). Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350, New York, NY, USA. ACM.
- Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B. (2007). Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 29–42, New York, NY, USA. ACM.
- Pietiläinen, A.-K., Oliver, E., LeBrun, J., Varghese, G., and Diot, C. (2009). Mobiclique: middleware for mobile social networking. In *WOSN '09: Proceedings of the 2nd ACM workshop on Online social networks*, pages 49–54, New York, NY, USA. ACM.
- Quercia, D., Ellis, J., and Capra, L. (2010). Using mobile phones to nurture social networks. *IEEE Pervasive Computing*, 9(3):12–20.
- Saha, D. and Mukherjee, A. (2003). Pervasive Computing: A Paradigm for the 21st Century. *Computer*, 36(3):25–31.
- Satyanarayanan, M. (2001). Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, 8:10–17.
- Toninelli, A., Pathak, A., Seyedi, A., Speicys Cardoso, R., and Issarny, V. (2010). Middleware Support for Mobile Social Ecosystems. In Computer Society Press, I., editor, *Second IEEE International Workshop on Middleware Engineering (COMPSAC Workshops)*, Seoul Korea, Republic Of.
- Weiser, M. (1991). The Computer for the 21st century. *Scientific American*, 265(3):66–75.