

## 4

### Trabalhos Relacionados

Os trabalhos apresentados nesta seção são os que buscam de alguma forma resolver as questões levantadas nos capítulos 1 e 2 e possuem alguma semelhança entre si. Eles serão comparados segundo quatro critérios: o primeiro diz respeito aos tipos de contexto pervasivos que são utilizadas e compartilhados pela solução apresentada. Um exemplo de contexto pervasivo que pode ser obtido é a posição geográfica do usuário. O contexto social de um usuário será melhor entendido à medida que mais informações de contexto forem obtidas.

O segundo critério leva em consideração a quantidade de redes sociais, se houver, com as quais o trabalho analisado se conecta ou mesmo se ele define sua própria rede social. O terceiro considera quais são as inferências sociais realizadas a partir dos contextos pervasivos extraídos do dispositivo móvel. Por exemplo, a partir da posição geográfica do usuário é possível deduzir a velocidade de deslocamento e, por inferência, se o usuário está em um congestionamento. Por fim, o quarto e último critério diz respeito à escalabilidade e performance da aplicação e de que forma essa questão é tratada por cada um dos trabalhos analisados.

#### 4.1

##### CenseMe

O CenseMe é uma aplicação social pervasiva desenvolvida por um grupo de pesquisadores de Dartmouth College, Hanover, NH, EUA (24). A motivação do grupo foi desenvolver uma aplicação sensível ao contexto para registrar e compartilhar automaticamente informações do usuário através de aplicações de redes sociais como Facebook e MySpace. O CenseMe infere informações de presença humana através dos sensores do smartphone e pode, por exemplo, inferir pelo nível de ruído, se o usuário está dançando em uma festa com amigos ou conversando em uma reunião. Dessa forma, a aplicação explora três tipos de dados de contextos: sons, imagens e informações do acelerômetro.

A aplicação também permite que o usuário personalize como sua informação de presença (*status*) será exibida editando alguns ícones ou textos. Além disso, o CenseMe leva em consideração as limitações impostas pelo sis-

tema operacional. Essas limitações vão desde a negação de pedidos de utilização de recursos (a prioridade são as ligações) até a falta de suporte da API a certas funcionalidades como acessar o acelerômetro.

Um característica importante de sua arquitetura é descrita pelos autores como *Split-Level Classification*: o aparelho realiza alguma classificação dos dados inferidos e produz primitivas, que são enviadas para um servidor remoto (*backend*), que realiza uma classificação adicional e produz fatos. Ou seja, ocorre uma divisão do processamento das informações de contexto entre o cliente móvel e o *backend*.

### 4.1.1

#### Limitações

O CenseMe atualmente se conecta às redes sociais para publicar informações de contexto do usuário. No entanto, não há nenhuma leitura das informações já existentes nestas redes e publicadas por algum outro meio. A localização dos amigos no Facebook Places, por exemplo, poderia ser lida e utilizada mesmo que o amigo em questão não utilize o CenseMe.

O texto não deixa claro questões referentes à performance na distribuição do conteúdo e apenas menciona os testes preliminares realizados com 8 usuários e estudos de caso envolvendo 22 usuários.

## 4.2

### Common Ground Application

O Common Ground Application é uma aplicação desenvolvida em J2ME por um grupo do Departamento de Computação da University of Bath, no Reino Unido. Seu objetivo é facilitar o processo de compartilhamento e estabelecimento de relações sociais entre dois usuários fisicamente próximos (25) ao permitir que eles compartilhem entre si o conteúdo de sua agenda de contatos e, a partir desse contexto, identificar as amizades em comum.

Essa troca de informação pode ser feita via Bluetooth ou NFC (Near Field Communication) e cada uma dessas duas tecnologias tem um uso particular que influencia na forma de utilização, na usabilidade e na aceitabilidade da aplicação. O Bluetooth, por exemplo, por não requerer tanta proximidade (normalmente 10 metros), pode ser usado como primeiro contato entre pessoas estranhas. Já o NFC requer uma proximidade de poucos centímetros (cerca de 2 cm) e, portanto, um contato quase físico entre usuários. Conseqüentemente, o NFC dificilmente será utilizado entre pessoas estranhas ou como primeira forma de contato. De ambas as formas, não há necessidade de um servidor remoto para processamento de informações como no caso do CenseMe.

### 4.2.1

#### Limitações

A maior limitação do Common Ground é não possuir integração com as redes sociais, que poderiam oferecer informações adicionais para o estabelecimento de amizades ou até mesmo servir de plataforma para distribuição da agenda do usuário. Como não há distribuição dessas informações, questões de performance e escalabilidade não são discutidas.

### 4.3

#### Context Watcher

O Context Watcher (26) é uma aplicação desenvolvida em Python para a plataforma Série 60 da Nokia cujo objetivo é tornar mais fácil para o usuário a gravação, o armazenamento e a utilização de informações de contexto. Essas informações podem ser utilizadas, por exemplo, como entrada para outros serviços, para o compartilhamento com família, amigos e colegas ou simplesmente para o armazenamento e utilização futura.

A aplicação foi desenvolvida a partir do *framework* MobiLife (27), que tem como objetivo descobrir, compartilhar e processar as informações de contexto de uma forma simples, facilitando o fluxo de informações entre diversos provedores e um usuário específico de forma a produzir informações de alto nível. As principais tarefas do Mobile Life são permitir a descoberta de provedores de contexto, suportar a associação entre diferentes provedores para prover informações situacionais de mais alto nível, padronizar a troca de contexto entre provedores e consumidores e facilitar o processamento dos dados através de componentes de processamento.

O Context Watcher utiliza diversos provedores de contexto, locais (no próprio dispositivo) e remotos, para gravar informações sobre a localização (baseada em GPS e ou GSM), o humor (baseado em entrada de dados pelo usuário), as atividades e encontros, dados do corpo (baseado em sensores que medem os batimentos cardíacos e outros localizados no pé), clima (baseado em provedores de contexto remotos) e também sobre dados visuais (fotos com dados contextuais). Os usuários da aplicação podem convidar outros usuários e, assim, formar uma rede social a partir da qual essas informações de contexto podem fluir de um dispositivo para outro.

### 4.3.1

#### Limitações

Assim como o Common Ground, o Context Watcher não interage com as redes sociais. No critério performance, os autores apenas mencionam que

a aplicação faz integração com um número considerável de provedores de contexto remotos e locais e que possui uma base de mais de 100 usuários sem, no entanto, revelar se algum tipo de teste de carga foi realizado durante o desenvolvimento.

#### 4.4 SAMOA

O SAMOA (*Socially-Aware MOBILE Architecture*) (28) é um *framework* para a criação de redes sociais semânticas baseadas em afinidades e interesses sociais. Ele provê uma série de ferramentas que facilitam a criação, o gerenciamento e a propagação da visibilidade de redes sociais independentes baseadas em localização.

Essas redes são centralizadas em um usuário (*ego user*) e baseiam-se em dois tipos de contexto: lugar (físico) e perfil (requisitos ou características do usuário). O primeiro limita a rede social aos usuários fisicamente próximos ao *ego user*, diferentemente da segunda.

Existem três papéis gerenciais (*roles*) nesta arquitetura:

- *managers*: são os *ego users* interessados em criar uma rede social e responsáveis por definir os critérios de formação da rede social;
- *clients*: são os usuários dentro do escopo de descoberta da rede social;
- *members*: são os usuários afiliados à rede.

Cada um desses atores é definido por um *profile* baseado no formato OWL (*Web Ontology Language*) contendo, por exemplo, a descrição e a atividade exercida no local ou o perfil do usuário (preferências, idade, nome etc).

Alguns testes de performance foram realizados e apresentaram bons resultados. Em um grupo de estudo com 65 usuários, o tempo de escolha do gerente da rede social levou cerca de 9ms.

##### 4.4.1 Limitações

O *middleware* poderia utilizar-se das redes sociais populares (como Facebook) para colher dados dinâmicos dos usuários. Isso é bastante útil para contornar a demora na identificação de novos clientes potenciais, que poderia ser agilizada através, por exemplo, da extração da localização de um determinado usuário baseando-se em seu último comentário em uma rede social. Além disso, as informações encontradas nos perfis dos usuários nas redes sociais poderiam ser utilizadas na criação das redes semânticas.

## 4.5

### MobiSoc

O MobiSoc (29) é um middleware para computação social móvel que fornece suporte para o desenvolvimento e *deployment* de aplicações MSCAs (*Mobile Social Computing Applications*). Ele oferece uma plataforma para captura, gerenciamento e compartilhamento de estados sociais de comunidades físicas e uma API para desenvolvimento de aplicações. O estado social é inferido a partir de diversas informações de contexto: perfis de pessoas ou lugares, afinidades entre pessoas e afinidades entre pessoas e lugares.

Assim como o SAMOA, a rede social é modelada através de perfis de usuários e de locais mas sua arquitetura é centralizada, o que possibilita o gerenciamento do “estado social” de toda uma comunidade. Ela é composta por camadas, é extensível e inclui um módulo para o gerenciamento de privacidade.

O MobiSoc possui um componente que executa no cliente móvel e outro que executa em um servidor web. Este último é composto por diversos módulos que podem ser distribuídos em diversos servidores para ganhar escalabilidade nas operações realizadas.

A primeira categoria de testes de performance simulou 1000 usuários, 100 lugares, 500 grupos e 500 eventos. Cada usuário tinha aproximadamente 150 amigos aleatórios e realizava o envio de sua localização de 10 em 10 segundos. O tempo de resposta ficou entre 300 e 500ms, impactado principalmente pelo acesso ao banco de dados.

#### 4.5.1

##### Limitações

O MobiSoc permite a implantação apenas em modo cliente-servidor, e assume sempre uma conexão permanente entre os clientes e o servidor. Essa conexão permanente é bastante crítica uma vez que exige bastante consumo de bateria, recursos computacionais do dispositivo e banda de Internet. Outra limitação é utilizar o perfil do usuário apenas do Facebook.

## 4.6

### MobiClique

MobiClique (30) é uma plataforma com arquitetura descentralizada que pode ser usado em redes móveis ad hoc usando Bluetooth ou WLAN. Assim, quando usuários móveis se conectam em modo ad hoc, ocorre uma troca dos seus perfis, e a depender da similaridade dos mesmos, as suas listas de contatos são atualizadas em ambos os dispositivos. A plataforma provê, ainda, uma API para encorajar o desenvolvimento de aplicações de terceiros.

Atualmente, o MobiClique obtém informações sobre os contatos apenas do Facebook e assume que o dispositivo se conectará à Internet periodicamente para sincronizar essas informações.

Em relação à performance, os testes da plataforma MobiClique foram feitos durante duas conferências e envolveram, respectivamente, 28 e 22 participantes.

#### 4.6.1 Limitações

A principal limitação da arquitetura é possuir integração apenas com o Facebook. O dinamismo de outras redes sociais, como o Twitter, pode ser de grande valia para determinar contatos e interesses em comum. Assim como os trabalhos anteriores, ele não obtém informações em tempo real sobre a localização dos usuários.

#### 4.7 SocialAware

SocialAware (31) é um *framework* para a criação de serviços de redes sociais móveis sensíveis ao contexto e que utiliza a arquitetura do WhozThat (32) para a troca e comparação de UserIDs de redes sociais como o Facebook. Possui uma arquitetura cliente-servidor, e usa o Bluetooth para a descoberta de dispositivos próximos, para realizar um *matching* de perfis e fazer a atualização das listas de contatos.

##### 4.7.1 Limitações

O SocialAware não leva em consideração o contexto atual do usuário. Essa característica fica evidente nas aplicações protótipo SocialAwareTunes e SocialAwareFlicks que, respectivamente, tocam uma lista de música ou exibe uma lista de *traillers*. Em ambos os casos, as listas são montadas a partir do perfil do usuário, assumindo que esse perfil contém seus filmes favoritos, para criar um *playlist*. No entanto, se o usuário gosta de Star Wars mas tuitar dizendo que gostaria de assistir Harry Potter e esse filme não estiver no seu perfil, ele não fará parte do *playlist* gerado.

#### 4.8 Análise comparativa

A tabela 4.1 compara os trabalhos segundo os critérios já estabelecidos.

Tabela 4.1: Quadro Comparativo dos Trabalhos Relacionados

Aplicação	Contexto	Uso das Informações de Contexto	Redes Sociais	Escalável
CenseMe	Sensores de Som	Informações sobre Presença	Facebook e MySpace	-
	Bateria	Regulagem dos Ciclos de Trabalho		
	Câmera Acelerômetro			
Common Ground	Agenda de Contatos	Troca de Contatos para propor amizades	-	-
Context Watcher	GPS ou GSM	Localização	Própria	-
	Entrada de Dados	Humor		
	Sensores Cardíacos	Dados do organismo		
SAMOA	Provedores Remotos	Clima	Própria	Sim
	Fotos	Dados Visuais		
	Localização	Formação de Redes Sociais Semânticas		
MobiSoc	Perfil do Usuário	Estado Social de uma Comunidade	Facebook	Sim
	perfis de pessoas			
	perfis de lugares			
MobiClique	afinidades entre pessoas	Troca de Contatos	Facebook	-
	afinidades entre pessoas e lugares			
	Perfil de Usuário no Facebook			
SocialAware	Perfil de Usuário no Facebook	Troca de Contatos	Facebook	-