

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ROGER SCHNEIDER MACK

**Sistema de recomendação baseado na
localização e perfil utilizando a plataforma
android**

Trabalho de Graduação.

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, dezembro de 2010.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do CIC: Prof. João César Netto

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Gostaria de demonstrar o meu sincero agradecimento a todos aqueles que de alguma forma contribuíram para a realização deste trabalho.

Ao professor Cláudio F. R. Geyer pela orientação no desenvolvimento do trabalho. Principalmente pela inspiração do tema e pelas várias idéias que me ajudaram a definir o curso dele.

À minha família que sempre me deu o apoio necessário. Especialmente aos meus pais que sempre fizeram o possível para realizar o sonho de me formar em um curso superior. Pela compreensão e ajuda nos momentos mais difíceis ao longo desses anos na faculdade.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS.....	7
LISTA DE FIGURAS.....	8
LISTA DE TABELAS	9
RESUMO.....	10
ABSTRACT.....	11
1 INTRODUÇÃO.....	12
1.1 Motivação	12
1.2 Objetivo	13
1.3 Organização do texto	13
2 SERVIÇOS BASEADOS EM LOCALIZAÇÃO.....	15
2.1 Definições de LBS.....	15
2.2 A evolução dos LBS	16
2.2.1 Gartner's Hype Curve.....	16
2.3 Tipos de serviço	17
2.4 Arquitetura de um LBS.....	17
2.5 Tecnologias de Localização	18
2.5.1 GPS.....	18
2.5.2 Cell ID.....	18
2.5.3 Wi-Fi.....	19
2.5.4 Vantagens e desvantagens.....	19
2.6 Aplicações.....	19

2.7	Considerações finais	20
3	SISTEMAS DE RECOMENDAÇÃO.....	21
3.1	Filtragem Baseada no Conteúdo	21
3.1.1	CRESDUP - Exemplo de sistema.....	22
3.1.2	Problemas	24
3.2	Filtragem Colaborativa	24
3.2.1	Algoritmos de filtragem colaborativa.....	25
3.2.2	Problemas	27
3.3	Filtragem Híbrida	28
3.4	Trabalhos relacionados	29
3.4.1	Referral Web	29
3.4.2	GroupLens.....	29
3.4.3	Ringo.....	29
3.4.4	MovieLens.....	30
3.5	Considerações Finais	30
4	PLATAFORMA ANDROID.....	31
4.1	Arquitetura do Android.....	31
4.1.1	Camada Azul	32
4.1.2	Camada Verde	32
4.1.3	Camada Amarela	32
4.1.4	Camada Vermelha	32
4.2	Máquina virtual Dalvik.....	32
4.3	Componentes do Android	33
4.3.1	Activities	33
4.3.2	Services	34
4.3.3	Broadcast Receivers.....	34
4.3.4	Content Providers	34
4.4	Recursos do Android	35
4.4.1	Location Services.....	35
4.4.2	Biblioteca externa do Google Maps	35
4.4.3	SQLiteDatabase	35
4.5	Considerações finais	35
5	MODELAGEM DO SISTEMA	36
5.1	Identificação do problema.....	36
5.2	Requisitos do Sistema.....	36

5.2.1	Requisitos básicos do sistema	36
5.2.2	Requisitos não funcionais do sistema	37
5.3	Características do Sistema	37
5.3.1	Definições.....	37
5.3.2	Questões de Segurança	38
5.4	Modelagem e Arquitetura do Sistema	38
5.4.1	Arquitetura geral do sistema	38
	Módulo Cliente.....	41
5.4.2	Módulo Servidor.....	45
5.5	Casos de uso.....	48
5.5.1	Consultar Recomendação.....	48
5.5.2	Seqüência de eventos	48
5.5.3	Inserir Usuário.....	50
5.5.4	Seqüência de eventos	50
5.5.5	Avaliar Estabelecimento	51
5.5.6	Seqüência de eventos	51
6	CONCLUSÃO	53
6.1	Análise geral do trabalho	53
6.2	Possibilidades futuras	53
	REFERÊNCIAS	55

LISTA DE ABREVIATURAS E SIGLAS

FC	Filtragem Colaborativa
FBC	Filtragem Baseada em Conteúdo
FH	Filtragem Híbrida
ACF	Automated Collaborative Filtering
UFRGS	Universidade Federal do Rio Grande do Sul
CRES DUP	Content Recommendation System based on private Dynamic User Profile
DUP	Dynamic User Profile
LBS	Location-Based Services
SIG	Sistemas de Informação Geográfica
pLSA	Probabilistic latent semantic analysis
SO	Sistema Operacional
VM	Virtual Machine
IMEI	International Mobile Equipment Identity
GPS	Global Positioning System
XML	Extensible Markup Language
API	Application Programming Interface

LISTA DE FIGURAS

<i>Figura 1.1: Crescimento de SO para telefones móveis (GARTNER, 2010)</i>	<i>13</i>
<i>Figura 2.1: Gartner's Hype Curve (FERRARO, 2010)</i>	<i>17</i>
<i>Figura 2.2: Vantagens e desvantagens das tecnologias de localização(FERRARO, 2010).....</i>	<i>19</i>
<i>Figura 3.1: Arquitetura do CRES DUP (CHEN, 2007)</i>	<i>23</i>
<i>Figura 3.2: Correlação de Pearson</i>	<i>26</i>
<i>Figura 3.3: Cálculo da predição.....</i>	<i>26</i>
<i>Figura 3.4: Filtragem híbrida (REATEGUI, 2005)</i>	<i>28</i>
<i>Figura 4.1: Arquitetura do Android (GOOGLE, 2010)</i>	<i>31</i>
<i>Figura 5.1: Arquitetura geral do sistema.....</i>	<i>39</i>
<i>Figura 5.2: Casos de uso do módulo cliente</i>	<i>41</i>
<i>Figura 5.3: Telas da funcionalidade de requisição da recomendação</i>	<i>42</i>
<i>Figura 5.4: Casos de uso do módulo servidor.</i>	<i>45</i>
<i>Figura 5.5: Modelo banco de dados do Servidor.</i>	<i>47</i>

LISTA DE TABELAS

<i>Tabela 5.1: Definição dos atributos do banco de dados do Servidor.....</i>	<i>47</i>
<i>Tabela 5.2: Caso de uso Consultar Recomendação.....</i>	<i>48</i>
<i>Tabela 5.3: Caso de uso Inserir Usuário</i>	<i>50</i>
<i>Tabela 5.4: Caso de uso Avaliar Estabelecimento</i>	<i>51</i>

RESUMO

Com a evolução da tecnologia, os dispositivos vem se tornando menores enquanto que a capacidade de armazenamento de dados e o poder computacional deles estão cada vez maiores. Assim o campo da computação móvel começa a crescer e a criar espaço para outras tecnologias como, por exemplo, novas plataformas de desenvolvimento e serviços baseados na localização utilizados por dispositivos móveis. Outra área que tem crescido bastante é a de sistemas de recomendação que tem o intuito de auxiliar os usuários fazendo a filtragem de informações que não são de seu interesse.

O agrupamento de tecnologias e serviços pode ser utilizado para o desenvolvimento de um sistema mais robusto e gerar um tipo de serviço mais especializado para o usuário. O propósito deste trabalho é de projetar um sistema de recomendação a ser utilizado por usuários de dispositivos móveis com a plataforma Android e que desejam receber a recomendação de estabelecimentos comerciais integrantes de uma categoria (restaurante, cinema, loja, etc) que é solicitada por ele e que utiliza como base para a procura a localização geográfica do indivíduo e suas preferências.

Assim, é realizado um estudo referente às principais funcionalidades envolvidas no sistema proposta. Iniciando pelas tecnologias de localização, passando pelos sistemas de recomendação e mostrando um pouco das características do desenvolvimento com Android. Ao final é apresentado um modelo de aplicação e suas principais funcionalidades e aspectos necessários para a implementação deste sistema.

Palavras-Chave: Dispositivos móveis, sistemas de recomendação, serviço de localização, android.

ABSTRACT

With the evolution of technology, devices are becoming smaller while the capacity of data storage and computational power of them are increasing. And the field of mobile computing begins to grow up and create space for other technologies, such as new development platforms and location-based services used by mobile devices. Another area that has grown is the recommendation systems that aim to help users by filtering information that is not of interest.

The grouping of technologies and services can be used to develop a more robust system and generate a more specialized type of service to the user. The purpose of this study is to design a recommendation system for use by users of mobile devices with the Android platform and they want to receive the recommendation of members of a shopping category (restaurant, cinema, shop, etc.) that is requested by him and uses as a basis to consider the geographic location of the individual and their preferences.

Thus, we conducted a study regarding the main features involved in the proposed system. Started by location technologies, through the recommendation systems and showing some of the features of developing with Android. At the end we present a model of application and its core functionality and features needed to implement this system.

Keywords: Mobile devices, recommendation systems, location services, android.

1 INTRODUÇÃO

1.1 Motivação

Nesta última década aconteceu um grande desenvolvimento de tecnologias para comunicação celular móvel, comunicação via satélite e redes locais sem fio. Com essa evolução os dispositivos móveis vêm se tornando cada vez mais populares e a computação móvel já se transformou em uma realidade no nosso cotidiano. A diversidade de dispositivos criados nos últimos anos, como por exemplo, PDAs (Personal Digital Assistants), Netbooks, smartphones e tablets, agregados ao fato de terem a capacidade de comunicação com redes, outros computadores ou até mesmo com outros dispositivos móveis, demonstram o grande sucesso da computação móvel.

Os dispositivos foram se aperfeiçoando e ganhando cada vez mais recursos. O armazenamento de dados, por exemplo, foi ganhando mais capacidade e já é possível expandí-lo através de cartões de memórias externos que são conectados ao aparelho e permitem salvar uma maior quantidade de dados. A utilização desses dados é uma ferramenta que enriquece os sistemas desenvolvidos para esses aparelhos como o caso de jogos e outras diversas aplicações. Possuem também uma grande capacidade de comunicação, visto que, atualmente podemos utilizar um aparelho para se comunicar com uma pessoa através de uma rede de telefonia em praticamente qualquer cidade do mundo e além disso, utilizar uma rede wi-fi para a navegação na internet e muitos outros serviços através da web.

Uma outra funcionalidade muito comum nos últimos aparelhos é o GPS. E com ele desenvolveram-se vários serviços baseados na localização, os chamados LBS (Location Based Services) que utilizam os dados da localização do usuário como um tipo de serviço relevante a ele. E com isso torna-se possível criar aplicações ricas, que aproveitam os diversos recursos em conjunto para oferecer uma melhor experiência de uso ao indivíduo.

Atualmente uma plataforma de desenvolvimento para smartphones tem atraído a atenção de muitas pessoas e vem ganhando mercado, é o Android. Entre alguns dos fatores que vem despertando o interesse da indústria e dos desenvolvedores podemos citar o fato de ser um sistema aberto, a quantidade rica de bibliotecas disponíveis e principalmente o fato de que existe uma grande empresa dando suporte a tudo isso, a Google. Apesar de ser uma plataforma criada muito recentemente, lançada no fim de 2007, já vem se transformando em um dos principais SO para telefones móveis como é mostrado na previsão do instituto de pesquisa e consultoria Gartner lançada em agosto de 2010 que é demonstrada pela figura 1.1.

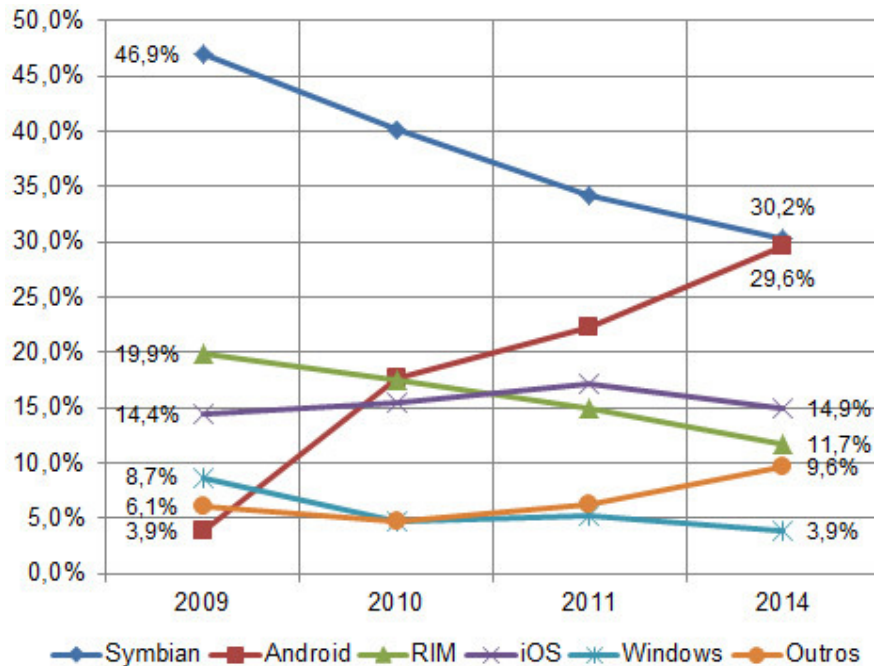


Figura 1.1: Crescimento de SO para telefones móveis (GARTNER, 2010)

1.2 Objetivo

O objetivo deste trabalho consiste em modelar um sistema de recomendação geral, que utilize as informações de localização geográfica e as preferências do usuário a partir de um perfil, para ser utilizado em um aparelho móvel rodando o sistema operacional Android.

A elaboração deste trabalho pretende explorar os recursos e serviços existentes nos dispositivos móveis atuais bem como as mais inovadoras tecnologias que vem sendo utilizadas. Para isso será proposto o modelo de um protótipo demonstrando o funcionamento das ferramentas e APIs que serão utilizadas para fornecer os serviços necessários ao sistema.

1.3 Organização do texto

Após a introdução que demonstrou as principais motivações e objetivos do sistema serão apresentados os próximos capítulos que estão organizados da seguinte forma:

- Capítulo 2 - Serão apresentados os Serviços Baseados em Localização, que é um dos pilares do sistema. Inicialmente serão mostradas algumas definições de LBS para demonstrar do que se tratam esses tipos de serviços. Depois será mostrada a evolução deles para compreendermos o desenvolvimento dele ao longo desses anos e como se tornou uma importante ferramenta para os dispositivos móveis. Então passaremos pela arquitetura de um LBS, algumas das tecnologias usadas bem como suas vantagens e desvantagens. E, por fim, são descritas algumas aplicações desse tipo de serviço.

- Capítulo 3 – Serão apresentados os Sistemas de Recomendação. É realizada uma breve introdução à esses sistemas e então são mostrados os seus tipos. Começando pela descrição da Filtragem Baseada no Conteúdo, passando pela Filtragem Colaborativa e Filtragem Híbrida, demonstrando os seus funcionamentos e problemas envolvidos. Por fim são citados alguns projetos que tiveram sucesso e uma breve descrição de cada um deles.
- Capítulo 4 - Abordagem da plataforma Android. Serão apresentados seus principais fundamentos para o entendimento da construção do modelo da aplicação. Um breve estudo mostrará as principais funcionalidades que serão utilizadas pelo sistema.
- Capítulo 5 - Definição da modelagem da aplicação, demonstrando como será sua arquitetura e funcionalidades, será também definido o modelo de dados utilizado e o projeto das classes do sistema.
- Capítulo 6 – Serão apresentados os detalhes da implementação do sistema demonstrando os detalhes técnicos necessários na constituição dele. As ferramentas, bibliotecas e APIs serão melhores explicadas nesse capítulo.
- Capítulo 7 - Por fim, o capítulo de conclusão mostra os principais aspectos positivos do desenvolvimento do sistema. É feita uma análise sobre os objetivos do trabalho alcançados. Finalizando o texto, são apresentadas sugestões para trabalhos futuros que pretendem explorar os assuntos abordados aqui.

2 SERVIÇOS BASEADOS EM LOCALIZAÇÃO

Neste capítulo será abordado o assunto Serviços Baseados em Localização que servirá como umas das bases para o desenvolvimento do sistema. Serão apresentados algumas definições conhecidas para os serviços baseados em localização, bem como alguns exemplos de tecnologias que são utilizadas. Por fim será mostrado de que maneira os LBS podem ser incorporados ao sistema.

2.1 Definições de LBS

Existem várias definições para o termo *Location-Based Services*, não se pode afirmar que uma definição é mais válida do que outra porque depende do enfoque e terminologias de cada área em que são estudadas e desenvolvidas, como é o das áreas de telecomunicações e da computação que utilizam esse tipo de tecnologia.

Como apresentado em (FERRARO, 2010) serviços baseados em localização são: *“Um serviço de informação, acessível a partir de dispositivos móveis através de uma rede móvel, que utiliza a habilidade de fazer uso da posição geográfica do dispositivo.* Nessa definição podemos notar que o enfoque são explicitamente os aparelhos móveis que são utilizados como instrumentos no consumo de um serviço disponível de localização, como são os casos de aparelhos celulares, aparelhos GPS, entre outros. Além disso o autor descreve que em resumo, um LBS é basicamente um serviço que apresenta três importantes questões que provêm valor ao serviço de localização:

- “Onde eu estou?” – Referente à localização do usuário;
- “O que posso fazer por perto?” – Referente à informações espaciais relacionadas com a localização do usuário;
- “O que eu acho nesse lugar?” – Referente à interação utilizando a localização e o contexto;

Outra definição é apresentada por (KUPPER, 2005) em que a GSM Association define LBS como serviços que usam a localização de um 'alvo' para adicionar valor ao serviço, onde este 'alvo' seria a entidade a ser localizada. Onde são citados como exemplos de valores adicionais que são gerados por esse serviço, a apresentação da localização de um alvo em um mapa ou a ativação automática de um serviço disparada pela aproximação a uma determinada localização.

Ainda, segundo (KUPPER, 2005) uma outra entidade, a 3GPP, apresenta que location services são na verdade uma parte integrante ou um subserviço de um serviço LBS porque o objetivo deles é exclusivamente a obtenção da localização de pessoas ou objetos alvos e com a possibilidade de tornar disponível a atores externos o resultado desta localização. O processamento dessa informação bruta, no sentido de filtrar ou

selecionar uma informação relevante ao uso, como um dado em específico, não é função do serviço de localização, cabe a ele apenas a função de gerar e distribuir os dados referentes à localização em si.

Outro grupo de serviços conhecidos como context-aware services (KUPPER, 2005), onde os LBS são vistos como um subconjunto deles. Os context-aware services são definidos como sendo serviços que adaptam automaticamente seu comportamento, com relação a um ou mais parâmetros relacionados ao contexto em que o dispositivo alvo se encontra.

2.2 A evolução dos LBS

A origem dos sistemas baseados em localização está ligado ao desenvolvimento de Sistema de Informação Geográfica (SIG). Os SIG são sistemas automatizados usados para armazenar, analisar e manipular dados geográficos. Eles são a evolução da cartografia tradicional e topografia. Na década de 80, com o advento da computação pessoal, ocorreu um rápido desenvolvimento desses sistemas bem como sua utilização para fins militares e também planejamento ambiental. Desde então, os LBS têm evoluído apoiados por tecnologias de localização mais eficazes e acessíveis.

2.2.1 Gartner's Hype Curve

Gartner's Hype Curve é um gráfico que representa o grau de maturidade, nível de adoção e aplicação comercial de tecnologias específicas. O termo é da autoria da Gartner, uma empresa consultora de tecnologias de informação que produziu a antevisão da evolução de diversas tecnologias.

Através desse gráfico podemos perceber que os LBS estão entre as tecnologias a terem seu potencial explorado. Porém essa tecnologia já foi a tecnologia chamada de revolucionária para o mundo das comunicações desde 2002. Mas como explicado por (FERRARO, 2010), essa revolução não ocorreu efetivamente para os aparelhos móveis por causa que a proposta de mercado para esse tipo de serviço estava a frente da tecnologia da época, pois os aparelhos móveis ainda não eram tão sofisticados. Uma visão de marketing muito citada era a seguinte: Uma pessoa passando num shopping ao passar em frente à uma cafeteria de repente receberia uma oferta através de uma notificação no aparelho dizendo “Imagine um Cappuccino. Entre em nossa loja e receba um desconto de \$ 0,50”. Porém os usuários de celulares recém estavam se acostumando com celulares com telas coloridas e toques polifônicos.

Pode-se descrever que hoje (FERRARO, 2010), em termos de LBS, adotando a curva 1 mostrada na gráfico abaixo, que foi desenvolvido em 1995 para caracterizar um entusiasmo excessivo e uma decepção posterior associada tipicamente com a introdução de novas tecnologias.

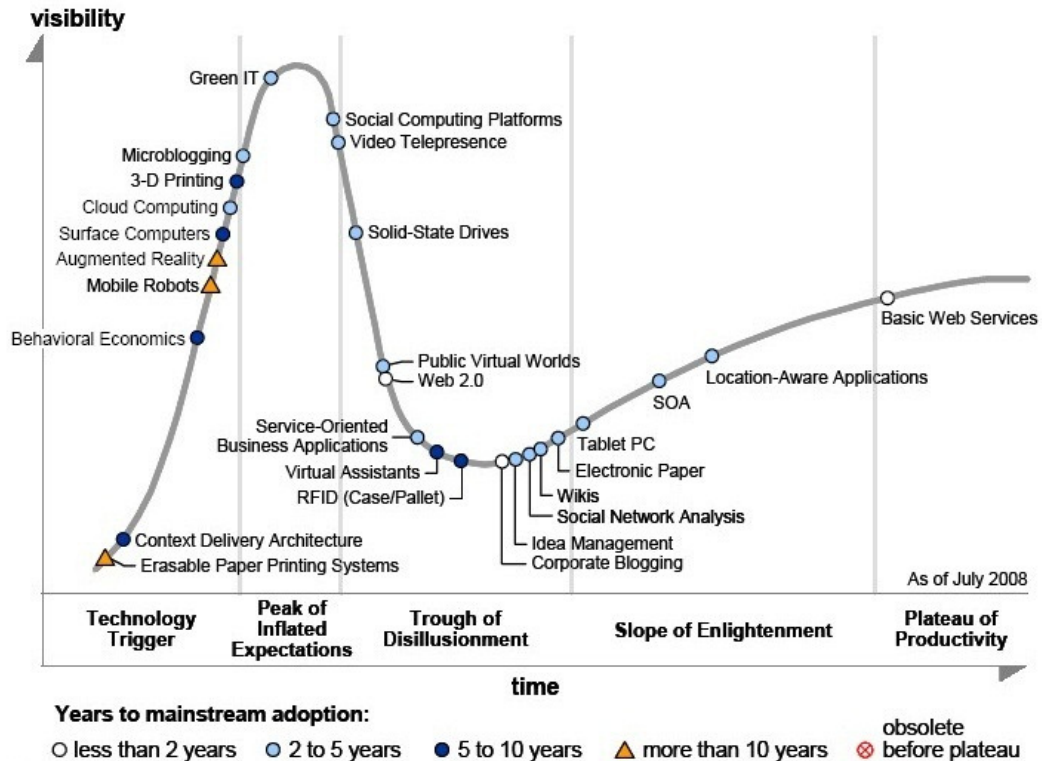


Figura 2.1: Gartner's Hype Curve (FERRARO, 2010)

2.3 Tipos de serviço

De acordo com (CHEN, 2005), serviços baseados em localização podem ser classificados em três tipos de serviços diferentes:

- Push, ou ativo – O próprio serviço percebe a localização de um usuário e oferece uma informação a este. Mas para que este serviço funcione de forma adequada e ofereça uma opção plausível ao usuário é necessário que exista um perfil já previamente cadastrado por ele.
- Pull, ou passivo – De forma inversa ao anterior, este tipo de serviço oferece uma informação ao usuário apenas se for consultado por ele. Funciona como uma forma de servidor de informações.
- Poll, ou periódico - Requisita informações periodicamente através de um protocolo bem definido.

2.4 Arquitetura de um LBS

De acordo com (SNEKKENES, 2001), a arquitetura de um sistema LBS, que considera as políticas de privacidade, possui as seguintes entidades:

- Política de privacidade de localização pessoal – Homologação do que pode ser liberado para quem e quando. Sendo que cada objeto localizado deve possuir uma política associada.

- Custódia da política – Onde as políticas são armazenadas e executadas. O conjunto de operações permitidas podem ser de leitura, escrita, modificação e etc, dependendo da identificação da entidade solicitante.
- Diretório da custódia da política – Um diretório que mostra onde a da custódia da política está armazenada.
- Provedor de localização – Entidade que fornece os dados de localização. Qualquer divulgação de dados deverá ser submetida à política.
- Provedor de serviços – Entidade que é uma combinação de dados de localização com outros dados para produzir algum serviço.
- Consumidor de serviço – Entidade que utiliza o serviço de localização.
- Iniciador do serviço – Entidade que gostaria ou aceita o serviço que é produzido.
- Solicitador do serviço – Entidade que realiza um requisição ao fornecedor do serviço.
- Pedido: Requisição genérica.
- Objeto localizado – A entidade cujos dados de localização serão requisitados para fornecer um serviço. A localização de objeto pode ou não ser conhecida.
- Proprietário do objeto localizado – A entidade que possui o objeto localizado (usuário do celular, por exemplo).

2.5 Tecnologias de Localização

Existem diferentes tecnologias para se tratar dos Serviços Baseados em Localização. Elas diferem no modo como operam e no tamanho do espaço que cobrem, como por exemplo, cobrir apenas o espaço físico de uma loja ou uma cidade ou região inteira.

A seguir será feita uma breve descrição de algumas tecnologias utilizadas em dispositivos móveis.

2.5.1 GPS

O GPS - Global Positioning System é uma tecnologia que foi desenvolvida pelo Departamento de Defesa dos Estados Unidos em 1973, e que foi projetada para fornecer a posição instantânea e a velocidade de deslocamento de um ponto sobre a superfície da Terra. O sistema funciona através de um conjunto de satélites que fornecem via rádio a um aparelho receptor a sua posição com referência às coordenadas geográficas terrestres. A precisão varia da qualidade do chipset do GPS e também pode ser afetada pela quantidade de núvens no céu já que isso interfere no sinal que vem dos satélites.

2.5.2 Cell ID

Utiliza a localização das estações base e a potência do sinal para medir a distância do aparelho até a fonte de origem do sinal. É utilizada em redes GSM, por exemplo, que se baseia que os dispositivos calculem constantemente o nível do sinal das antenas para poderem se comunicar com as estações mais próximas e assim consumir menos energia.

2.5.3 Wi-Fi

A IEEE 802.11, que também é conhecida como rede Wi-Fi (Wireless Fidelity) é um padrão que opera em faixas de frequências que não necessitam de licença para instalação e/ou operação. Pode determinar a localização de um dispositivo pela localização do access point e a força do sinal, podendo utilizar também mais de um access point para aumentar a precisão de localização.

2.5.4 Vantagens e desvantagens

Geralmente, quando existem várias tecnologias para alguma coisa, algumas possuem vantagens e desvantagens em relação às outras e o seu uso está condicionado ao contexto em que ela irá ser aplicada, propiciando que as vantagens da tecnologia escolhida sejam de fato boas e as desvantagens possam ser contornadas, amenizadas ou apenas desconsideradas. Com as tecnologias de localização isso não é diferente, através da figura abaixo podemos observar algumas vantagens e desvantagens das tecnologias citadas anteriormente.

Tecnologia	Vantagens	Desvantagens
GPS	<ul style="list-style-type: none"> • Tecnologia muito bem difundida • Experimentada e testada • Barata • Precisa • De livre acesso 	<ul style="list-style-type: none"> • Geralmente não funciona bem em lugares fechados • Consome muita energia • Requer um aparelho habilitado
Cell ID	<ul style="list-style-type: none"> • Disponível na maioria dos dispositivos • Funciona em lugares fechados • Não consome muita energia 	<ul style="list-style-type: none"> • O acesso à base de dados de Cell ID não é gratuita • É menos precisa que GPS
Wi-Fi	<ul style="list-style-type: none"> • Boa relação entre confiança e precisão • Tecnologia gratuita 	<ul style="list-style-type: none"> • Necessita que o dispositivo possua essa tecnologia habilitada • Pode consumir muita energia da bateria do dispositivo

Figura 2.2: Vantagens e desvantagens das tecnologias de localização(FERRARO, 2010)

2.6 Aplicações

Os Serviços Baseados em Localização podem agregar muitas vantagens a um sistema para um dispositivo móvel como vimos ao longo deste capítulo. A fim de exemplificar sua utilização serão citados algumas aplicações desses serviços de acordo com (QUEIROZ, 2006):

- Serviços de Segurança – Auxiliar nos serviços de atendimento médico de urgência, bombeiros e policias. Utilizando-se dos LBS para fornecer a localização da ocorrência de maneira mais rápida e eficiente permite-se melhorar esse tipo de serviço.

- Serviços de Informação – Fornece vários tipos de informações ao usuário utilizando sua localização. Pode fornecer informações como endereços de restaurantes, bancos 24 horas, postos de gasolina e também podem servir como serviço de navegação oferecendo visualização de mapas e rotas.
- Serviços Comerciais – Fornece um tipo de serviço muito útil à empresas. Pode ser utilizado para publicidade, promoções ou outros tipos de alertas semelhantes que utilizem a proximidade como atributo chave.
- Serviços de Entretenimento – Utilizada para servir como um elemento que acrescente alguma experiência a mais ao usuário em uma ferramenta de entretenimento, como por exemplo, jogos. Também pode ser uma potencial ferramenta de marketing para atrair um grupo específico de pessoas.

2.7 Considerações finais

Neste capítulo foi introduzido o conceito de Serviços Baseados em Localização. Foram apresentados alguns conceitos e características deste tipo de serviço a fim de mostrar a importância de um LBS em um sistema a nível tecnológico e estratégico. Por fim, analisando as tecnologias existentes com suas vantagens e desvantagens e o seu tipo de serviço a ser oferecido no sistema escolheu-se o GPS como melhor a ser desenvolvido. No capítulo sobre Android serão abordados as ferramentas e APIs utilizadas para consumir um serviço de LBS.

3 SISTEMAS DE RECOMENDAÇÃO

Atualmente existe uma grande sobrecarga de informações devido ao grande crescimento da quantidade de dados e à precária organização estrutural da informação, tornando cada vez mais difícil de se encontrar o que se deseja. Existe um grande volume de informações acessíveis hoje em dia, muito mais do que uma pessoa poderia absorver em toda sua vida. A maioria dessas informações apresentam-se pouco estruturadas e possuem pouca relevância ao indivíduo, este problema de sobrecarga de informações (SCHAFER, 2009) resulta em um grande esforço e perda de tempo para o usuário na tentativa de selecionar o que lhe é útil.

Uma solução para este problema é a utilização de ferramentas que auxiliem na filtragem de informações relevantes de forma eficaz como os sistemas de recomendação. Eles têm como objetivo principal sugerir itens de um determinado domínio que melhor se encaixam no perfil de interesse do usuário e assim realizar a recomendação. Esta que se for realizada da forma correta reduzirá a tarefa exaustiva de busca.

Sistemas de recomendação são geralmente classificados em três categorias de acordo com (BALABANOVIC, 1997) que se diferenciam pelo modo de utilização dos padrões de comportamento e relacionamento com os usuários, ou seja, também denominada como filtragem. São os seguintes:

- Filtragem baseada no conteúdo – A recomendação é realizada baseada em itens similares àqueles que o usuário demonstrou preferência no passado.
- Filtragem colaborativa - A recomendação é realizada baseada em itens de pessoas com gostos e preferências similares àquelas que o usuário demonstrou preferência no passado.
- Filtragem híbrida – Este método é baseado na combinação das duas filtragens acima.

A seguir iremos descrever mais claramente os tipos de filtragem:

3.1 Filtragem Baseada no Conteúdo

Atualmente quando um indivíduo quer buscar algum tipo de informação, como por exemplo, procurando um livro para comprar em uma livraria, ele possivelmente não está querendo um título em específico, mas certamente não gostará de ler o título e assunto de todos os livros disponíveis nela. Olhar todos os livros, dependendo do tamanho do acervo da livraria, se tornaria inviável para ele e certamente a maioria nem seria de seu interesse. Então o que normalmente as pessoas tendem a se focar em uma seção para buscar somente assuntos de seu interesse.

É no princípio de que os usuários tendem a se interessar por itens similares aos que demonstraram interesse no passado que funciona a Filtragem Baseada em Conteúdo. É analisada a correlação do conteúdo dos itens com o perfil do usuário a fim de sugerir os itens relevantes e descartar os itens insignificantes. A idéia básica é selecionar os itens preferidos pelos usuários cujas preferências mais se assemelham ao gosto do usuário alvo. Para desenvolver um sistema FBC é preciso resolver dois subproblemas:

- Procurar uma representação – É necessário criar uma forma de representar um documento através de suas características. Como descrito em (PAZZANI, 1999) as abordagens de FBC representam documentos pelas palavras “importantes” contidas nele. Um exemplo de representação para livros, seria por exemplo, descrever seus atributos como autor, tema, número de páginas, etc. Ao encontrar uma representação adequada é possível criar um algoritmo de classificação que possa qualificar o objeto ao usuário.
- Criar um perfil que permita que os documentos sejam recomendados – O perfil do usuário deverá ser criado através de uma coleta de informações que deverá representar as preferências dele. Esta coleta de dados pode ser realizada tanto implicitamente quanto explicitamente. A aquisição de coleta implícita ocorre quando informações de seus hábitos passados são coletados de alguma forma para a criação de seu perfil, o que ocorre geralmente por site de compras, por exemplo. Já a aquisição explícita ocorre quando o usuário relata seus interesses de alguma forma ao próprio sistema, como através de um formulário de seus interesses, e isto se torna necessário em alguns casos em que descobrir o histórico de uma pessoa não é viável.

3.1.1 CRES DUP - Exemplo de sistema

Para demonstrar de forma mais clara como a FBC pode ser utilizada será brevemente mostrado o sistema CRES DUP (Content REcommendation System based on private Dynamic User Profile) (CHEN, 2007), que tem a capacidade de procurar mensagens específicas de acordo com os dados privados de um usuário ao mesmo tempo que esses dados são processados no lado cliente. Através da mineração de dados é construído um DUP (Dynamic User Profile) que é regularmente atualizado de acordo com as operações diárias e feedbacks dos usuários.

3.1.1.1 Arquitetura

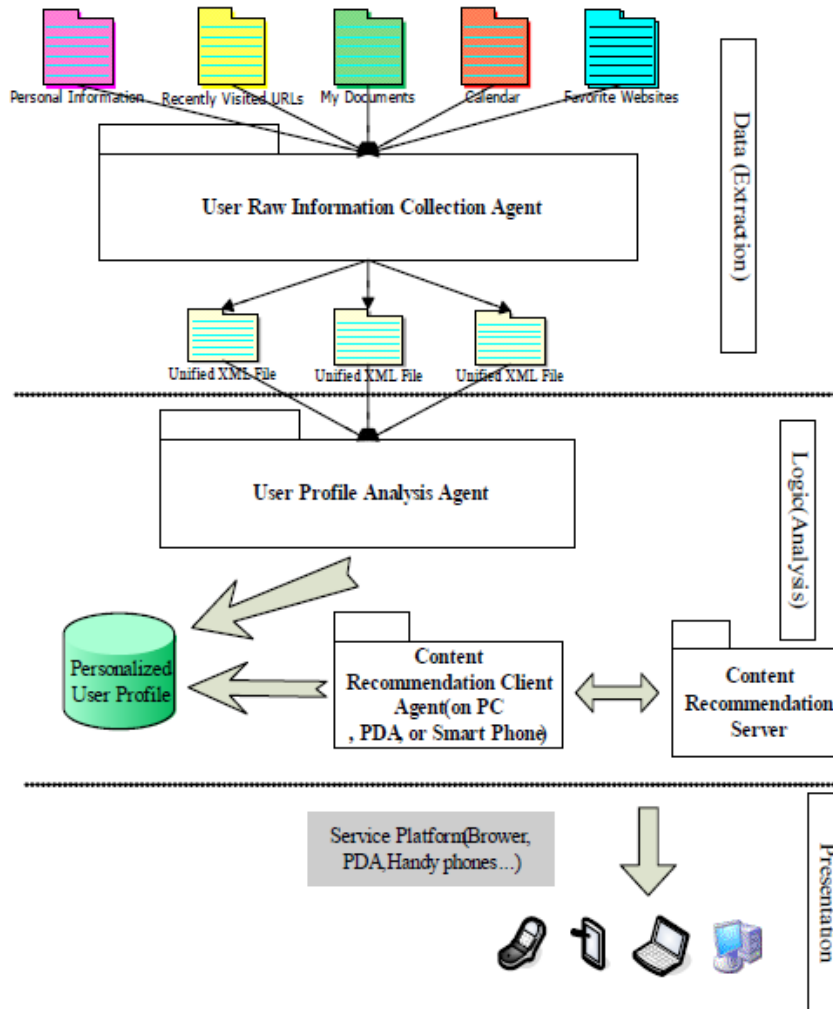


Figura 3.1: Arquitetura do CRES DUP (CHEN, 2007)

Como vemos na figura acima o sistema possui três camadas e é formado por quatro módulos centrais (User Raw Information Collection Agent; User Profile Analysis Agent; Content Recommendation Server; Content Recommendation Client Agent).

A primeira camada é a camada de dados que inclui o módulo “User Raw Information Collection Agent”. A segunda camada é a camada lógica que inclui o módulo “User Profile Analysis Agent” e o processo de construção e uso do DUP. A terceira camada é a de apresentação que inclui a interface personalizada para clientes de PC e dispositivos móveis.

O “User Raw Information Collection Agent” é responsável por realizar a coleta dos dados brutos de outros recursos como: informação pessoal básica, URL’s visitadas recentemente, documentos feitos pelo usuário, calendário e agenda e websites favoritos. Então ele assim define as preferências do usuário.

O “User Profile Analysis Agent” é responsável por definir o perfil do usuário e utilizar o algoritmo de recomendação baseado em conteúdo para recomendar itens semelhantes aos que preferiu no passado.

O “Content Recommendation Server” é responsável por gerenciar o conteúdo e enviá-lo ao cliente de acordo com a solicitação do cliente enquanto que o “Content Recommendation Client Agent” é responsável por receber dados de informações do servidor com base no perfil de usuário, podendo ser tanto um módulo independente do software do lado do cliente ou uma parte integrada do software.

3.1.2 Problemas

A FBC possui algumas limitações que em alguns casos pode tornar a criação de um sistema muito complexa.

Na representação dos dados das preferências do usuário é necessário classificar as características deste objeto, seguindo o exemplo do livro, existem características como título, assunto, autor e número de páginas, porém elas não podem ter a mesma relevância. O número de páginas possivelmente não será um fator muito relevante na escolha enquanto que o autor provavelmente será. Então nos deparamos com o problema de como avaliar corretamente a relevância das características de um objeto para podermos apenas recomendar itens relevantes e descartar os itens não pertinentes.

É difícil de se diferenciar na filtragem, itens com diferentes qualidades, estilos e pontos de vista. Por exemplo, um artigo bem escrito não é diferenciado de um artigo mal escrito se ambos possuem termos semelhantes. E as recomendações ocorrem somente sobre coisas que o usuário já conhece, a sugestão para novos assuntos que possam ser de interesse ao usuário não será realizada porque no seu histórico conhecido nunca o usuário visualizou aquele tipo de conteúdo.

E por último, outro problema existente na FBC é a coleta de informações do usuário para a criação de um perfil que defina corretamente os gostos do usuário. Dando um exemplo mais claro disso podemos citar um site para aluguel de DVD's. Para a criação de um perfil adequado, neste caso, deveríamos coletar informações sobre todos os filmes que o usuário já viu juntamente com a sua avaliação, porém isso se torna inviável de se fazer.

3.2 Filtragem Colaborativa

Diferente da filtragem baseada em conteúdo, a filtragem colaborativa é baseada na análise humana do conteúdo. Para cada item que um usuário experimenta (a leitura de um artigo ou um jantar no restaurante, por exemplo) é associada uma avaliação a este item dada explicitamente pelo usuário, e assim, estabelecendo um perfil de interesses.

Através do perfil de interesses é possível de se fazer uma correlação com outros perfis e criar um grupo de pessoas com interesses similares. Desta forma, as notas dos vizinhos (usuários de perfis similares) são utilizadas para gerar recomendações ao usuário, técnica também chamada de filtragem social (SHARDANAND, 1995).

Os sistemas de recomendação baseados em FC não possuem as limitações que os sistemas de recomendação FBC têm. Com esta técnica os itens da recomendação não necessitam mais serem transformados para uma representação legível ao computador e não há a exigência da compreensão ou reconhecimento do conteúdo dos itens. Também não se faz necessário criar um perfil que necessite agregar tantos dados do histórico do usuário para realizar uma recomendação eficiente porque ele vai ser baseado na opinião de outras pessoas com um perfil similar ao do usuário em vez de se basear no histórico dele. E ainda, agora abre-se a oportunidade do usuário receber a recomendação de um

assunto totalmente novo depois que alguém com um perfil similar avaliar positivamente aquele item.

Segundo (HERLOCKER, 2000) os primeiros sistemas de filtragem colaborativa necessitavam que os usuários indicassem explicitamente os itens de interesse para ocorrer o relacionamento de suas opiniões com a predição. Depois, os sistemas acabaram automatizando esse processo (conhecidos também como ACF - Automated collaborative filtering) através da coleta de pontuação dos itens pelo usuário. O processo de pontuação é a forma com que o usuário irá definir o grau de importância de um item experimentado para sua necessidade de informação. Com essas avaliações é possível apresentar ao usuário uma média de pontuações para cada item com potencial de interesse, assim permitindo a ele visualizar os itens que são considerados de interesse relevante pelo grupo e evitar itens considerados de pouco interesse.

Segundo (HERLOCKER, 1999) a técnica de recomendação pode ser dividido em três passos principais:

- Calcular os pesos – Realizar o cálculo de peso para cada usuário em relação ao usuário alvo utilizando uma métrica de similaridade.
- Buscar os vizinhos – Seleciona-se um subconjunto de usuários com maiores níveis de similaridades (vizinhos) para considerar na predição.
- Realizar a predição – Normaliza-se as avaliações fornecidas pelos usuários para o item analisado na recomendação e calcula-se a predição, ponderando-se as avaliações dos vizinhos com seus respectivos pesos de similaridades.

3.2.1 Algoritmos de filtragem colaborativa

Os algoritmos utilizados em FC funcionam a partir dos valores atribuídos aos itens. Esses valores são atribuídos de acordo com a relevância que o usuário definiu ao escolher um valor de uma escala numérica para o item.

De acordo com (SANDVIG, 2008) os algoritmos de filtragem colaborativa são divididos em duas classes. Um classe é a dos algoritmos baseados em memória e a outra é dos algoritmos baseados em modelos.

3.2.1.1 Algoritmos baseados em memória

A técnica dos algoritmos baseados em memória é a mais popular para aplicações que utilizem filtragem colaborativa. É mantida uma base de dados com todas as preferências de todos usuários e todos itens, e quando deve haver uma predição ocorre um processamento em toda a base de dados.

No processamento é calculada a predição do voto para um item e um determinado usuário com base na avaliação de todos os outros usuários fizeram para aquele item. Os algoritmos mais utilizados são os que usam o modelo do vizinho mais próximo (k-Nearest Neighbor). Esses algoritmos utilizam várias medidas de distância entre os usuários, como por exemplo a distância euclidiana ou coeficiente de correlação de Pearson.

Os algoritmos baseados em memória são geralmente mais simples e os dados são mais facilmente inseridos na base de dados. Porém, se a base se tornar muito grande o seu desempenho pode ficar comprometido já que toda predição é processada com todos os dados da base.

- k-Nearest Neighbor (SANDVIG, 2008) – É o algoritmo padrão mais utilizado e razoavelmente preciso. A similaridade entre o usuário alvo u e o seu vizinho v é calculado utilizando-se a correlação de Pearson.

$$sim_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u) * (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} * \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}}$$

Figura 3.2: Correlação de Pearson

Onde, $r_{u,i}$ e $r_{v,i}$ são as avaliações de algum item i para u e v ,

respectivamente, e \bar{r}_u e \bar{r}_v são as médias das avaliações de u e v sobre I . Após as similaridades serem calculadas, os k usuários mais similares que tenha avaliado o item alvo são selecionados como os vizinhos. Isto implica que um usuário alvo pode possuir diferentes vizinhos para cada item, por isso é comum se definir um limite de similaridade para realizar uma filtragem entre os vizinhos. Após a identificação dos vizinhos deve-se computar a predição para um item alvo e o usuário alvo da seguinte forma:

$$pred_{u,i} = \bar{r}_u + \frac{\sum_{v \in V} sim_{u,v} (r_{v,i} - \bar{r}_v)}{\sum_{v \in V} |sim_{u,v}|}$$

Figura 3.3: Cálculo da predição

Onde V é o conjunto de k vizinhos similares que avaliaram i . $r_{v,i}$ é a avaliação de i para o vizinho v . \bar{r}_u e \bar{r}_v são as avaliações médias de todos os itens avaliados para u e v , respectivamente. E $sim_{u,v}$ é a correlação de Pearson entre u e v . A fórmula calcula o grau de preferência para todos os vizinhos, ponderando pela sua similaridade, e então é acrescentado à avaliação média do usuário alvo.

3.2.1.2 Algoritmos baseados em modelos

Os algoritmos baseados em modelos têm o objetivo de diminuir o tempo e o alto custo computacional dos algoritmos baseados em memória no que refere-se ao processamento na etapa de predição da recomendação.

A princípio é realizado um agrupamento das preferências do usuário com o objetivo de formular um modelo que servirá como base para as predições. A construção deste modelo pode consumir horas ou até dias mas o resultado é um modelo pequeno, rápido e tão preciso quanto o algoritmo baseado em memória.

Este método não é recomendado em ambientes em que as mudanças das preferências dos usuários sejam atualizadas frequentemente porque existe a necessidade de se criar um novo modelo que, como dito anteriormente, pode custar muito processamento. Mas para ambientes em que haja pouca mudança nas preferências esse método é altamente recomendado.

- k-Means Clustering (SANDVIG, 2008) – O algoritmo padrão para a filtragem colaborativa baseada em modelos utiliza k formas para agrupar usuários similares. Dado um conjunto de usuários é realizado um particionamento deles em k grupos com base em uma medida de similaridade. A descoberta desses grupos de usuários é então aplicada à tarefa da criação da vizinhança.

Para realizar a recomendação de um item alvo para um usuário alvo são selecionados os vizinhos de um grupo de usuários que possuem uma avaliação do item alvo e cujo perfil agregado é similar ao do usuário alvo. Para realizar a busca da vizinhança no grupo é utilizada a correlação de Pearson.

- Probabilistic Latent Semantic Analysis (SANDVIG, 2008) – Apresenta uma abordagem probabilística para caracterizar a latência ou ocultação de associações semânticas entre as co-ocorrências de objetos. A pLSA é aplicada para a criação de grupos de usuários no contexto da filtragem colaborativa.

3.2.2 Problemas

Em (SANDVIG, 2008) é citado que um problema que pode ocorrer com os sistema FC é a recomendação de itens super valorizados por causa da injeção de dados falsos no sistema. Isso pode ocorrer quando alguém, com o objetivo de aumentar a média de avaliações de um item, começa a criar vários perfis falsos e cria avaliações muito boas daqueles itens que desejam colocar em destaque. Assim, itens que não fariam parte das recomendações normais porque não existiriam pessoas, com perfis similares, que avaliassem bem aquele item, agora começam a fazer parte de suas preferências porque possuem uma boa média de avaliações.

Segundo (FILHO, 2004) e (BALABANOVIC, 1997), existem algumas limitações para FC:

- Novo item – Quando um novo item aparece no banco de dados do sistema, enquanto nenhum usuário avaliá-lo, não será possível realizar nenhuma predição sobre ele. Os itens só serão recomendados ao usuário se os seus vizinhos avaliarem de forma positiva ele. No caso do cálculo da similaridade de Pearson, um item deve ser avaliado por pelo menos dois usuários, caso contrário, ele não entrará no cálculo da similaridade e será desconsiderado do perfil do usuário. A este problema é dado o nome de “Problema do Novo Item”.

- Novo usuário – Quando um novo usuário é cadastrado no sistema seu perfil de avaliações encontra-se vazio, pois ainda não existem avaliações realizadas por ele. Sem as avaliações também não é possível de se encontrar vizinhos a ele. Este problema é chamado de “Problema do Novo Usuário”, ou também é referenciado pela literatura como “Cold Start User”.
- Escalabilidade – Quando o volume de dados aumenta muito porque o número de usuário, itens e avaliações se tornou muito grande, ocorre o problema da escalabilidade. Os sistemas que devem realizar o cálculo da vizinhança para cada cálculo de previsão podem demorar muito por causa do grande processamento de informações e chegar a um tempo de resposta inaceitável.
- Esparsividade – À medida que o número de itens da base de dados vai aumentando, isso dificulta a probabilidade de usuários possuírem itens em comum, e assim acaba por reduzindo o número de vizinhos para se realizar uma previsão. Diminuindo-se o número de vizinhos as recomendações ficam mais pobres causando um impacto negativo na confiabilidade das recomendações.
- Super-especialização – Este problema ocorre quando um usuário possui muitos itens semelhantes entre si e assim acaba recebendo recomendações de somente de itens semelhantes àqueles. Por exemplo, em um sistema de recomendação de filmes como o MovieLens, o usuário possui muitas avaliações em filmes de guerra no seu perfil, então ele receberá a maioria das recomendações relacionadas com filmes de guerra, mesmo que a preferência dele tenha mudado.

3.3 Filtragem Híbrida

As técnicas de filtragem citadas anteriormente, filtragem baseada no conteúdo e filtragem colaborativa, possuem alguns problemas que são próprios de cada uma. Então resolveu-se criar uma filtragem híbrida com o intuito de unir a FBC e a FC combinando os pontos fortes de cada uma, e assim eliminando suas fraquezas. Através da figura abaixo é possível visualizar algumas dessas vantagens.

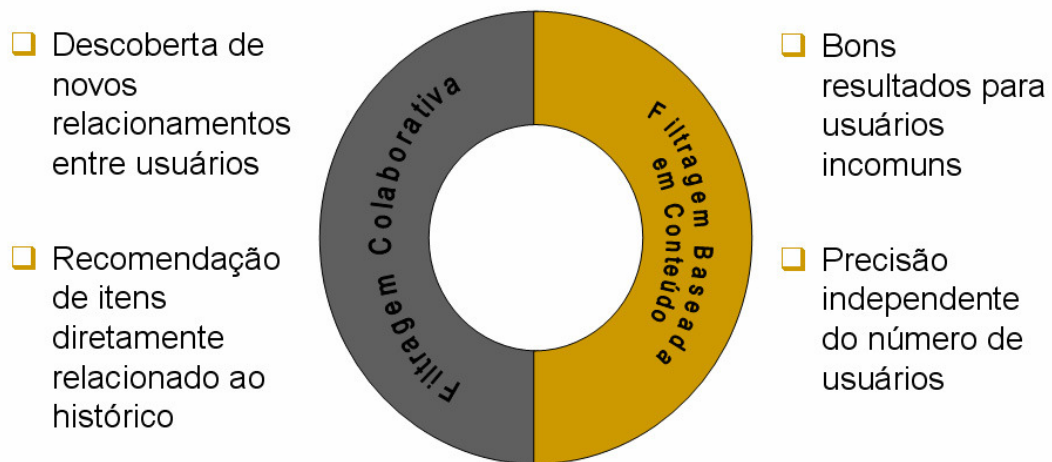


Figura 3.4: Filtragem híbrida (REATEGUI, 2005)

Utilizando os benefícios da FC é possível utilizar a experiência de outros usuários para complementar a FBC que pode ser incompleta e imprecisa se o histórico do usuário alvo não é suficientemente completo. E utilizando os benefícios da FBC não é necessário possuir uma base muito grande de usuários para que as recomendações sejam mesmo relevantes ao usuário alvo.

Essa técnica foi pouco explorada por conter poucas referências apesar de que sabe-se de alguns sistemas que foram desenvolvidos utilizando essa abordagem para eliminar algum ponto negativo de outra técnica, como é o caso do sistema de recomendação de páginas Web chamado de Fab, desenvolvido na Stanford University.

3.4 Trabalhos relacionados

Em (HERLOCKER, 2000) e (REATEGUI, 2005) são citados alguns exemplos de projetos que tiveram sucesso em sua pesquisa na área de sistemas de recomendação e agora são muito utilizados como parâmetros para pesquisas e desenvolvimentos de outros sistemas nessa área.

3.4.1 Referral Web

Este projeto tem o objetivo de identificar e visualizar redes sociais de grupos de pessoas ligadas à atividades profissionais.

Quando um indivíduo tem seus dados adicionados no sistema é realizada uma consulta por textos existentes na web que referenciem seu nome. Através desses documentos buscados são extraídos nomes de outros indivíduos que são citados. Com isso cria-se uma rede global de citações que é representada na forma de um grafo. A rede está baseada somente na relação entre usuário, documento e citações, a importância e relevâncias dos trabalhos ou indivíduos não é considerada.

3.4.2 GroupLens

O GroupLens é um projeto de pesquisa desenvolvido pela Universidade de Minnesota pelo qual foi criado um sistema de recomendação de artigos que foram postados em fóruns de discussão, e tem como objetivo diminuir o tempo gasto das pessoas para ler as notícias.

O sistema coleta as avaliações dos usuários em relação aos artigos lidos por eles em uma escala de 1 a 5. Ele utiliza essa avaliação para identificar os vizinhos mais próximos do usuário alvo e assim prever novos artigos que possam ser de interesse daquele usuário.

3.4.3 Ringo

Ringo é um sistema desenvolvido para recomendação personalizada de música e foi desenvolvido no Massachusetts Institute of Technology.

O sistema explora similaridades entre os gostos musicais de diferentes usuários para recomendar os itens, baseando-se na idéia de que as pessoas possuem a tendência de terem padrões de gostos similares a certos grupos. As preferências pessoais dos indivíduos são descobertas através da avaliação de canções que são conhecidas por

eles,. Para a pontuação é aplicada uma escala Likert de 7 pontos, sendo o 1 (não gosta), 4 (indiferente) e 7 (adora).

Ele permite a requisição de 3 tipos de recomendação: itens de interesse, itens rejeitados e nota atribuída a um determinado item. O retorno dado pelo Ringo aos usuário não inclui nenhuma informação em particular sobre a identidade dos outros usuários que contribuíram com as recomendações com o objetivo de manter a identidade dos outros usuários em segredo.

3.4.4 MovieLens

O MovieLens é um sistema de recomendação para filmes disponível na internet sendo necessário apenas cadastro de alguns dados pessoais para ter acesso ao sistema.

De início é pedido ao usuário a avaliação de no mínimo 15 filmes, atribuindo-lhes uma nota na escala de 1 a 5. Este requisito é necessário devido ao problema da partida a frio. A partir daí começam a serem apresentados ao usuário uma lista de cinco melhores filmes recomendados para ele, ao lado de uma média. A base do MovieLens tem 3000 filmes e é atualizada semanalmente com os novos lançamentos nos Estados Unidos com a colaboração do IMDB.

3.5 Considerações Finais

Neste capítulo demonstramos os principais conceitos dos sistemas de recomendação com o objetivo de elucidar o leitor ao significado deles e o valor que pode ser agregado com seu uso. Foram descritas as três categorias em que os sistemas de recomendações são divididos bem como os problemas envolvidos na utilização de cada uma delas. Também foram mostrados alguns algoritmos geralmente utilizados nesses tipos de filtragem. Por fim são descritos alguns projetos de sucesso para demonstrar os benefícios que as recomendações podem trazer aos usuários que a utilizam.

4 PLATAFORMA ANDROID

Este capítulo introduz uma visão geral sobre a plataforma de desenvolvimento Android. O objetivo é apresentar algumas características, como os componentes que formam essa plataforma juntamente com suas bibliotecas e APIs. Não será realizado um estudo muito aprofundado da plataforma porque não é esse o escopo do trabalho. Inicialmente serão demonstradas algumas das razões para a escolha dessa plataforma.

4.1 Arquitetura do Android

A Arquitetura da plataforma Android é formada por várias camadas (GOOGLE, 2010), que também é chamada de *pilha de softwares*. E cada camada possui programas que suportam funções específicas no sistema operacional.

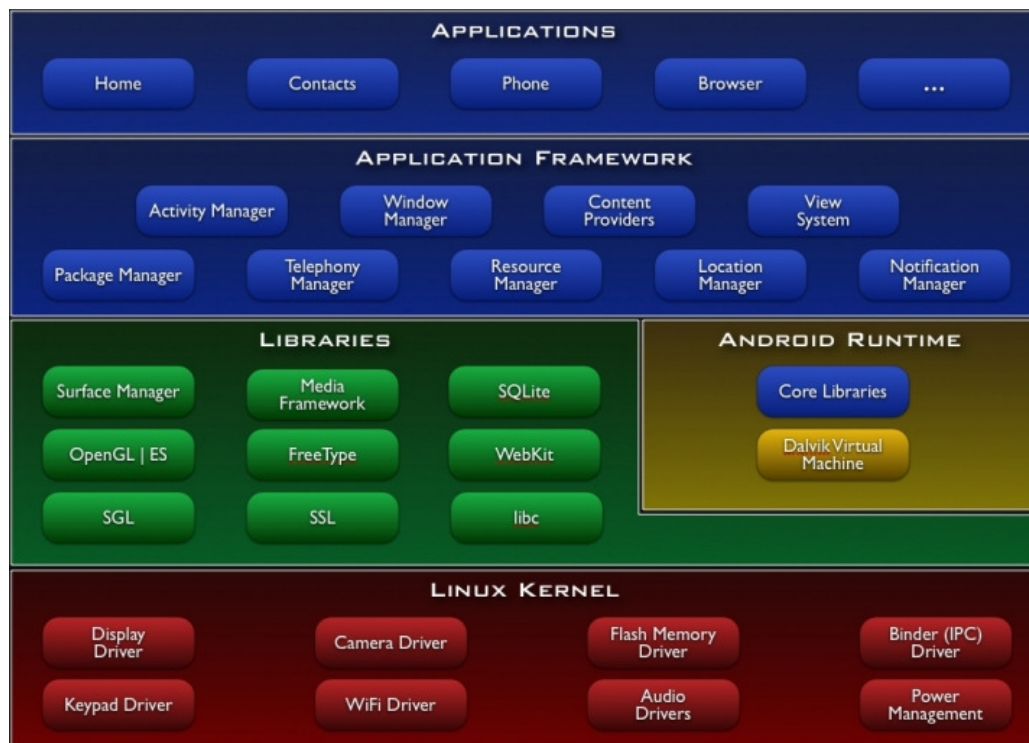


Figura 4.1: Arquitetura do Android (GOOGLE, 2010)

4.1.1 Camada Azul

4.1.1.1 Applications

No topo da pilha encontra-se um conjunto de aplicações como cliente de email, calendário, mapas, navegador e também as funções mais básicas como fazer chamadas telefônicas. É nesta camada que o usuário comum interage com as interfaces dos aplicativos. As camadas mais baixas da pilha são acessadas apenas por desenvolvedores e fabricantes de hardware.

4.1.1.2 Application Framework

A próxima camada é o framework de aplicação que inclui os programas que gerenciam as funções básicas do telefone, como alocação de recursos, aplicações de telefone, mudança entre processos ou programas. Possibilita o total acesso ao desenvolvedores para que tirem vantagens das capacidades do hardware do dispositivo, execução de serviços em background, definição de alarmes e muitos outros. É projetada para simplificar a reutilização de componentes e assim criar ferramentas mais sofisticadas a partir de ferramentas mais básicas.

4.1.2 Camada Verde

4.1.2.1 Libraries

Camada em que várias bibliotecas são disponibilizadas para que os desenvolvedores tenham acesso à componentes do sistema como banco de dados, navegadores web, aceleradores gráficos e outros.

4.1.3 Camada Amarela

Esta camada fica no mesmo nível da camada verde.

4.1.3.1 Android Runtime

Nesta camada é incluído um conjunto de bibliotecas do núcleo Java. Também é nessa camada que se encontra a máquina virtual Dalvik onde são executados os processos da aplicação. A Dalvik VM executa os arquivos em um formato otimizado para o baixo consumo de memória, são os arquivos *Dalvik Executable (.dex)* que são resultantes da compilação.

4.1.4 Camada Vermelha

4.1.4.1 Linux Kernel

A base da pilha é o kernel do Android que foi baseada na versão 2.6 do Linux. Inclui os programas de gerenciamento de memória, configurações de segurança, gerenciamento de energia e vários drivers de hardware.

4.2 Máquina virtual Dalvik

As aplicações escritas em Java são compiladas em bytecodes Dalvik e executadas usando a Máquina virtual Dalvik, que é uma máquina virtual especializada desenvolvida para uso em dispositivos móveis, o que permite que programas sejam distribuídos em formato binário (bytecode) e possam ser executados em qualquer

dispositivo Android, independentemente do processador utilizado. Apesar das aplicações Android serem escritas na linguagem Java, ela não é uma máquina virtual Java, já que não executa bytecode JVM.

4.3 Componentes do Android

No Android os elementos básicos de aplicações podem ser compartilhados. Se as permissões são concedidas a estes elementos, é possível usar os recursos já desenvolvidos de outras aplicações. Com o Android ainda é permitido que processos de aplicação possam ser iniciados somente quando forem necessários, bem como permitir a instanciação de objetos Java. Em outras palavras, enquanto a maioria dos sistemas e aplicações têm um único ponto de entrada, ou função principal, aplicações Android invocam componentes centrais que são instanciados e executados quando eles forem necessários. Existem quatro tipos principais de componentes Android: *activities*, *services*, *broadcast receivers*, e *content providers*.

Os *content providers* são ativados quando forem requisitados por um *ContentResolver*. Já os outros três componentes – *activities*, *services* e *broadcast receivers* – são ativados através de mensagens assíncronas chamadas de *Intents*. Essas mensagens são utilizadas para a requisição de algum recurso entre estes componentes, como, por exemplo, transmitir o pedido de uma *activity* para mostrar ao usuário uma imagem ou permitir que ele edite um texto.

4.3.1 Activities

Uma atividade é uma representação de uma interface visual que permite aos usuários escolher uma determinada tarefa a desempenhar. É através da composição dessas atividades, que agem independentes, podemos criar uma interface de usuário consistente. Em outras palavras, cada atividade é aplicada como uma subclasse da classe base de atividade. Dentro de aplicações podem existir uma grande diversidade de atividades. Por exemplo, uma aplicação pode ser feita de uma única atividade ou pode ser uma coleção de atividades. Podemos passar de uma atividade para outra sendo que uma atividade atual tenha como ponto de partida uma outra atividade. Além disso, cada atividade em um aplicativo é atribuída uma janela padrão que a desenha. Esses janelas podem encher a tela, flutuar em cima de outras janelas, ou podem interagir com outras janelas.

4.3.1.1 Ciclo de vida

Cada componente tem seus ciclos de vida. Eles são os diferentes estados em que um componente assume desde que sua instância é criada até a sua destruição. Podem representar que uma atividade está ativa ou inativa, se está visível ao usuário ou não. Uma atividade possui essencialmente três estados:

- *Active or running* – A atividade está ativa quando está em primeiro plano, pronta para receber as ações dos usuários. Ou seja, ela está no topo da pilha de execuções da tarefa.
- *Paused* – A atividade está pausada quando perdeu o foco mas ainda visível ao usuário. Ela está em segundo plano porque uma outra atividade está cobrindo parcialmente ela ou é transparente, mas ainda continua completamente viva com todas as suas informações.

- *Stopped* – A atividade está parada quando uma outra atividade cobriu completamente ela, ou seja, não está visível ao usuário. Mas ainda continua mantendo seus estados e informações enquanto não for morta pelo sistema.

Cada transição desses estados exibe uma notificação ao sistema e pode ser capturada pela aplicação. Assim um desenvolvedor pode estabelecer um comportamento ou função para cada uma das ações da atividade, como, por exemplo, capturar o método *onStart()* e definir uma função para quando a atividade for iniciada mas antes dela se tornar visível ao usuário.

4.3.2 Services

Os serviços não têm uma interface visual para o usuário porque eles executam em segundo plano por tempo indeterminado. Esta poderia ser a música de fundo que toca quando os usuários lidam com outros problemas ou poderia ser cálculos de fundo que são fornecidas ao usuário. O player de mídia é o exemplo perfeito de um serviço. Como as músicas são tocadas a partir de uma lista, a aplicação é projetada em torno de múltiplas atividades que permitem ao usuário escolher músicas e tocá-las. Por outro lado, a reprodução em si não é uma atividade, mas um serviço. Atividades são ações como iniciar e parar o aplicativo que está sendo utilizado, mas queremos que a música continue tocando até quando passamos para algo diferente. A thread principal do processo da aplicação é onde os serviços são executados.

4.3.2.1 Ciclo de vida

Como as atividades, os serviços também possuem ciclos de vida. Mas são diferentes porque neste caso não é visualizado pelos usuário, então possui apenas os estados de *running* ou *stopped*. Pode ser iniciado chamando-se o *Context.startService()* ou parado chamando-se o *Context.stopService()*.

4.3.3 Broadcast Receivers

Este tipo de componente é bastante simples e funciona apenas recebendo e reagindo à transmissão de anúncios. Muitos broadcast receivers iniciam através do sistema, como por exemplo, anúncios sobre a bateria fraca ou alteração do idioma. Muitos aplicativos também podem iniciar transmissões. Isso permite que aplicativos possam se comunicar com outras aplicações.

4.3.3.1 Ciclo de vida

Quando uma mensagem de broadcast chega ao receptor, o Android chama o seu método *onReceive()* e passa o objeto *Intent* contido na mensagem. O broadcast receiver é considerado como ativo somente enquanto ele está executando este método. Quando *onReceive()* retorna, ele se torna inativo.

4.3.4 Content Providers

Eles são responsáveis pela persistência dos dados, ou seja, gravação e acesso de informações, em uma base de dados como é o caso do SQLite por exemplo, e permite a disponibilização destes dados para outras atividades ou aplicativos.

4.4 Recursos do Android

Aqui serão brevemente mostrados alguns recursos que são oferecidos por APIs e bibliotecas no Android (GOOGLE, 2010).

4.4.1 Location Services

Para os serviços de localização que serão usados para obter as coordenadas geográficas do usuário será utilizada a API *android.location*. Pode-se registrar um *listener* que receberá as atualizações de posições geográficas periodicamente ou quando entrar em alguma área específica (alerta de proximidade).

A classe *LocationManager* proverá o acesso ao serviço de localização. Já a classe *LocationListener* pode ser registrada com o *LocationManager* para receber atualizações periódicas. E a classe *LocationProvider* é a responsável por determinar a tecnologia de localização que será utilizada pelo *LocationManager* para fornecer a posição, que no caso do sistema proposto neste trabalho utilizará a tecnologia Sistema de Posicionamento Global - GPS.

Outro serviço interessante é o de localização de um local pelo endereço urbano. Com o GPS é possível somente conseguir as coordenadas geográficas, mas o Android disponibiliza uma classe chamada *Geocoder* que possui a função de traduzir essa coordenada em um endereço real para que possamos utilizar no sistema.

4.4.2 Biblioteca externa do Google Maps

Uma outra funcionalidade que será utilizada pelo sistema proposto é a visualização de mapas com a localização de um estabelecimento. Assim, o usuário poderá encontrar com mais facilidade a localização de um determinado local recomendado através do recurso de um mapa.

O Google oferece esse serviço através da biblioteca dentro do pacote "com.google.android.maps". A classe *MapActivity* é uma atividade que pode ser estendida para se obter uma *MapView*, que é usada para exibir o mapa e cuidar das informações requisitadas para a internet (buscar imagens do mapa por exemplo). A classe *MapController* é utilizada para se interagir com o mapa. Como o Google Maps não faz parte do padrão Open Source é necessário se obter uma chave que é disponibilizada pelo próprio Google para a utilização desse tipo de serviço.

4.4.3 SQLiteDatabase

Um outro recurso importante para uma aplicação no Android é a persistência de dados. Mais especificamente, um banco de dados, que servirá como repositório para o armazenamento e manipulação de informações para o sistema proposto. O *SQLiteDatabase* é o recurso oferecido pelo Android para este propósito. Nele existem métodos para criar, excluir, executar comandos SQL e também executar algumas outras tarefas comuns de banco de dados.

4.5 Considerações finais

Neste capítulos foram demonstrados alguns aspectos fundamentais da plataforma Android para uma melhor compreensão dessa nova tecnologia. Foram também mostrados alguns recursos que serão fundamentais para o desenvolvimento do sistema e o entendimento do modelo proposto.

5 MODELAGEM DO SISTEMA

Este capítulo apresenta a modelagem proposta para o sistema de recomendação que se chamará Recomendo. Apresentaremos qual o problema que deseja-se solucionar e como isso será realizado demonstrando as funcionalidades do sistema.

5.1 Identificação do problema

Atualmente existe uma grande disponibilidade de serviços para diferentes fins. A falta de informação já não é mais o problema, agora a questão é como acessar essa informação e como filtrar os dados para conseguirmos aquilo que desejamos.

O problema que estamos desejando solucionar neste trabalho é descrito pela seguinte situação:

- Uma pessoa está andando na rua, então ela começa a sentir fome. Consequentemente se ela deseja comer alguma coisa vai querer buscar um estabelecimento próximo, mas não qualquer um, ela quer um que satisfaça sua preferência. Então qual é a melhor forma de fazer isto?

Pensando nisso, e aproveitando de tecnologias e serviços disponíveis para se criar uma aplicação que possa dar suporte à um usuário de um aparelho móvel definimos o seguinte problema: Como uma pessoa pode utilizar seu smartphone para descobrir a localização de um estabelecimento próximo que deseja ir?

Então a seguir será definida uma modelagem e arquitetura necessárias para essa aplicação solucionar o problema proposto.

5.2 Requisitos do Sistema

Os requisitos do sistema podem ser definidos como condições ou capacidades necessárias que o software deve possuir para que o usuário possa resolver um problema ou atingir um objetivo e para atender as necessidades ou restrições dos outros componentes do sistema.

5.2.1 Requisitos básicos do sistema

Aqui constam os requisitos mínimos de aceitação que o sistema deve apresentar.

O sistema deve:

- Permitir a vinculação do usuário com seu smartphone no sistema;
- Gerar uma lista de estabelecimentos como resposta à recomendação requisitada pelo usuário, com base em sua localização e preferências;

- Possibilitar a visualização do mapa com a identificação do estabelecimento selecionado;
- Possibilitar a avaliação de um estabelecimento;
- Realizar a busca pela posição geográfica de forma invisível ao usuário;

5.2.2 Requisitos não funcionais do sistema

Aqui serão listados alguns requisitos não funcionais que apesar de não serem essenciais para o funcionamento do sistema, devem ser considerados por corresponderem a um fator de qualidade de software.

- Usabilidade: É importante que uma aplicação que rode em um aparelho móvel seja bem simples, mas que possua todas as funções essenciais para o sistema. A interface deverá ter uma tela principal que a partir dela possam ser tomada todas as ações principais do sistema.
- Desempenho: É um requisito muito importante no sistema. As respostas às ações dos usuários deverão ser rápidas e objetivas e para isso é importante manter um modelo simples. A função principal do sistema que é gerar um lista de estabelecimentos como recomendação também não deve ser demorada, para que a utilização não se torne cansativa ao usuário.
- Hardware: Não deverá ser um fator que impacte na performance do sistema. A exigência não deverá ser de que o aparelho do usuário tenha a mais alta configuração de processador e memória do mercado. A aplicação deverá rodar bem em qualquer aparelho com Android na medida do possível.
- Software: Para permitir a exploração de todas as funcionalidades e o perfeito funcionamento da aplicação no aparelho deverá ser exigido no mínimo a versão 2.1 do sistema operacional Android.
- Rede: Também deve ser exigência que o aparelho esteja liberado para acessar a internet, pois a execução das funções do sistema requisitam chamadas a um módulo externo que enviará respostas através da rede.

5.3 Características do Sistema

5.3.1 Definições

A escolha das características do sistema tiveram como objetivo tornar ele mais simples e de fácil utilização pelo usuário.

Em relação aos tipos de estabelecimentos que deveriam ser recomendados pelo sistema optou-se por colocar algo mais genérico para que não deixasse o sistema centrado em apenas algum tipo de estabelecimento, como restaurantes por exemplo. Mas sim, deixar o sistema simples possível e ao mesmo tempo genérico o suficiente para que possamos ter mais tipos para realizar a recomendação. Então definiu-se “categoria” e “especialidade”, a primeira é relativa ao tipo de estabelecimento, como bares e restaurantes, boates, lojas, etc. Já a segunda refere-se a uma especialização da primeira como forma de definir um atributo para que mais tarde isso também seja usado como preferência do usuário para buscar uma recomendação. Exemplificando, uma categoria “bar ou restaurante” tem as especialidades “Pizzaria”, “Churrascaria”,

“Lanchonete”, e outros, enquanto que a categoria boates teria como gênero as especialidades “Baile Sertanejo”, “Pub”, “Danceteria” e assim por diante.

Já no caso do tipo de filtragem escolhida para a recomendação, resolveu-se utilizar a filtragem colaborativa. A filtragem baseada em conteúdo não foi escolhida por exigir que cada tipo de item tenha que ter várias características diferentes para poderem formar parte das preferências dos usuário. Como a filtragem colaborativa é mais simples nesta questão e também não necessita que se tenha um histórico do usuário para poder realizar a recomendação, foi feita a escolha dela.

Depois de escolher a filtragem colaborativa, foi necessário modificar o sistema a fim de não centralizar todas operações no dispositivo, já que a recomendação deverá ser realizada com base na escolha de outras pessoas, esses dados devem ficar em uma base comum para todos. Assim, essa base deverá ficar disponível para todos os dispositivos consultarem ela, foi proposto então uma arquitetura cliente-servidor, onde o cliente tem a aplicação e o servidor os dados.

Tendo tudo isso em mente, também é preciso definir onde será executado o algoritmo de recomendação. Já que o volume de dados pode se tornar muito grande devido serem adicionados vários estabelecimentos e vários usuários, então decidiu-se delegar a tarefa da implementação do algoritmo de recomendação ao servidor. Assim o volume de dados a serem transmitidos na rede será baixo e a aplicação no cliente exigirá menos esforço computacional, tendo isso grande vantagem para um dispositivo móvel.

5.3.2 Questões de Segurança

O sistema em questão não trata aspectos de segurança como uma prioridade porque não existem dados sigilosos a serem guardados, considerando que apenas o nome e o login serão as informações do usuário a serem salvas no banco de dados.

Porém, como os dados estarão disponibilizados pelo servidor através de um web service serão tomados alguns cuidados. Para poder solicitar um serviço ao servidor será exigida uma identificação única do dispositivo, o IMEI (International Mobile Equipment Identity) que servirá como uma espécie de chave para verificação. Essa chave vinculará o aparelho ao usuário, não podendo mais ninguém acessar os dados de preferências do usuário sem conhecer essa chave. Isso será invisível aos olhos do usuário porque a aplicação do aparelho tratará disso sozinha.

5.4 Modelagem e Arquitetura do Sistema

A modelagem e arquitetura têm o objetivo de apresentar uma abstração dos componentes do sistema, o conjunto de funcionalidades que são encapsuladas e que assumem algum tipo de responsabilidade dentro do sistema, e seus conectores. Esses conectores que são responsáveis pela coordenação e comunicação entre os componentes.

5.4.1 Arquitetura geral do sistema

O sistema será definido por uma arquitetura cliente-servidor. Onde o cliente será o dispositivo móvel com Android que rodará a aplicação em que o usuário irá requisitar e visualizar a recomendação do estabelecimento. O servidor será o responsável por gerar as recomendações e manter o banco de dados com todas as informações pertinentes ao sistema, que serão descritos detalhadamente mais tarde. Na figura abaixo é mostrada a

representação da arquitetura cliente-servidor do sistema juntamente com seus componentes.

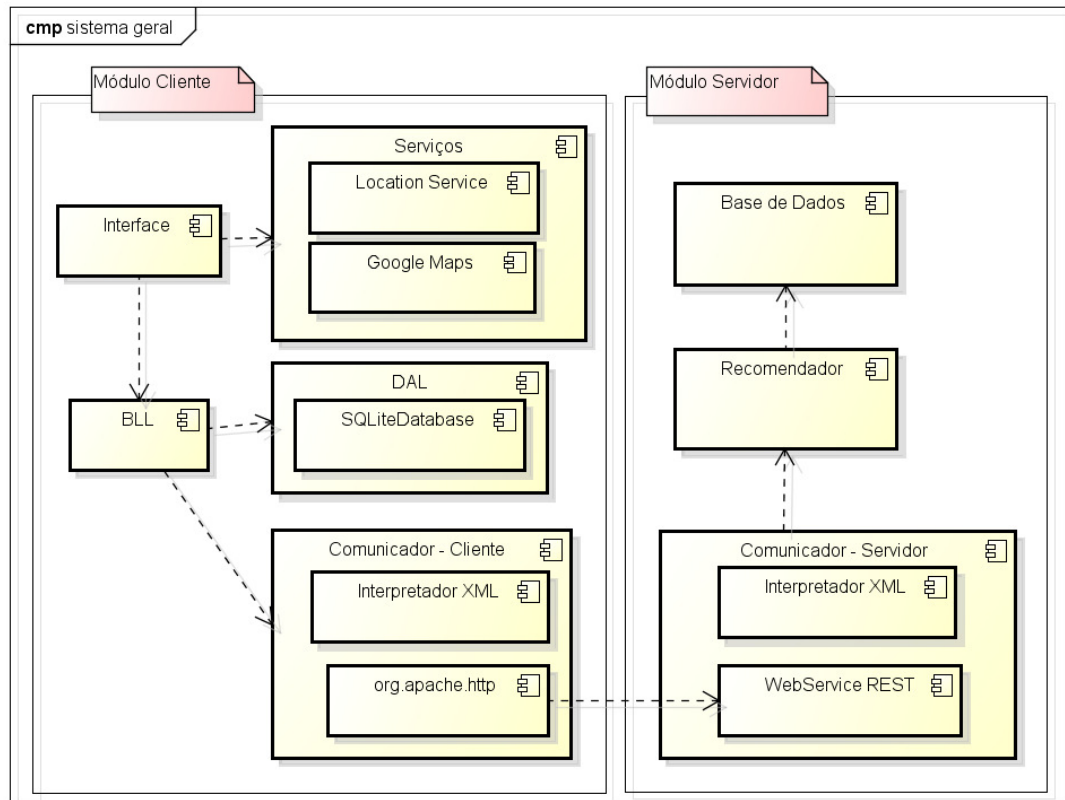


Figura 5.1: Arquitetura geral do sistema

Visando a questão de mobilidade, a parte Cliente será responsável por toda a interação e coleta de dados do usuário. Através dela que o usuário poderá acessar o serviço para receber a recomendação de localidades com o dispositivo móvel rodando Android. A aplicação contida no aparelho também será responsável por se comunicar com o serviço de localização e fornecer o local onde o indivíduo se encontra. Mais adiante serão explicados mais a fundo a parte Cliente.

O controle de segurança dos dados é garantido pelo Servidor que ficará responsável por todas consultas e modificações na base de dados. Nenhum dado sigiloso de um usuário será fornecido a outro porque isso não faz parte das funcionalidade do sistema, ele apenas mandará os dados de estabelecimentos recomendados a um dispositivo. A base de dados guardará os dados de todos os usuários, estabelecimentos e mais algumas informações que serão descritas mais tarde quando mostraremos todo o modelo de dados e explicaremos cada um de seus dados.

Para um entendimento mais claro do sistema ele será dividido em dois módulos. O Módulo Cliente que será a aplicação que rodará no Android e o Módulo Servidor que será a aplicação que rodará no servidor e englobará a base de dados.

5.4.1.1 Componentes

Módulo Cliente - Através da figura 5.1 é possível visualizar uma visão geral do sistema com o módulo cliente e o módulo servidor juntamente com seus componentes. A seguir serão descritos os componentes do Cliente.

- Interface – Responsável por agregar as classes responsáveis por controlar as telas da aplicação, ou seja, as atividades que fazem a interação com o usuário.
- BLL – O Business Logic Layer possui as classes que controlarão o acesso da interface aos componentes do banco de dados e da comunicação.
- Serviços – Este componente é responsável por agregar as classes responsáveis por utilizar a API de Location Service e a biblioteca do Google Maps.
- DAL – O Data Access Layer contém as classes que realizam as operações no banco de dados SQLite que fica dentro do dispositivo.
- Comunicador – Cliente – É o responsável por realizar as funções de comunicação com o módulo servidor. Possui a classe que manipula os arquivos xml, tanto para gerar a requisição quanto para interpretar as respostas para uma model. E também agrega a classe que realiza as chamadas e respostas ao webService no Servidor.

Módulo Servidor - A seguir serão descritos os componentes do Servidor.

- Base de Dados – Responsável por controlar o acesso à base de dados, realizar consultas e inserções.
- Recomendador – Contém as classes responsáveis por realizar a manipulação dos dados, rodar o algoritmo de recomendação e gerar a resposta.
- Comunicador – Servidor - É o responsável por realizar as funções de comunicação com o módulo cliente. Possui a classe que manipula os arquivos xml, tanto para gerar o resultado das recomendações, quanto para interpretar as requisições feitas pelo Cliente.

5.4.1.2 Comunicação

Para realizar a comunicação entre os módulos cliente e servidor será utilizada uma arquitetura REST que deverá ser implementada pelos componentes chamados de “Comunicadores”. Com ela é possível criar uma interface web simples que utilize HTTP e XML para a comunicação de dados.

Através do protocolo HTTP é possível simplificar a comunicação na medida que trata disso como requisições de serviços que disponibilizam recursos. A complexidade de regras de negócio são deslocadas para o serviço e assim a interface fica mais simplificada. Possui um conjunto de operações bem definidas - POST, GET, PUT e DELETE. Essas operações serão utilizadas para requisitar um recurso (documento xml no caso deste sistema) ou também enviar.

Os detalhes da arquitetura REST não serão abordados neste trabalho. A implementação dela será abstraída pelos Comunicadores do Cliente e do Servidor que serão responsáveis pelas comunicações entre eles.

Módulo Cliente

Será o módulo responsável por apresentar e controlar as telas do sistema e se comunicar com o módulo servidor para realizar a recomendação. Também será função deste módulo utilizar as bibliotecas de GPS e mapas que serão responsáveis por buscar a localização do usuário e apresentar o ponto do local recomendado pelo sistema como resposta, respectivamente.

5.4.1.3 Funcionalidades

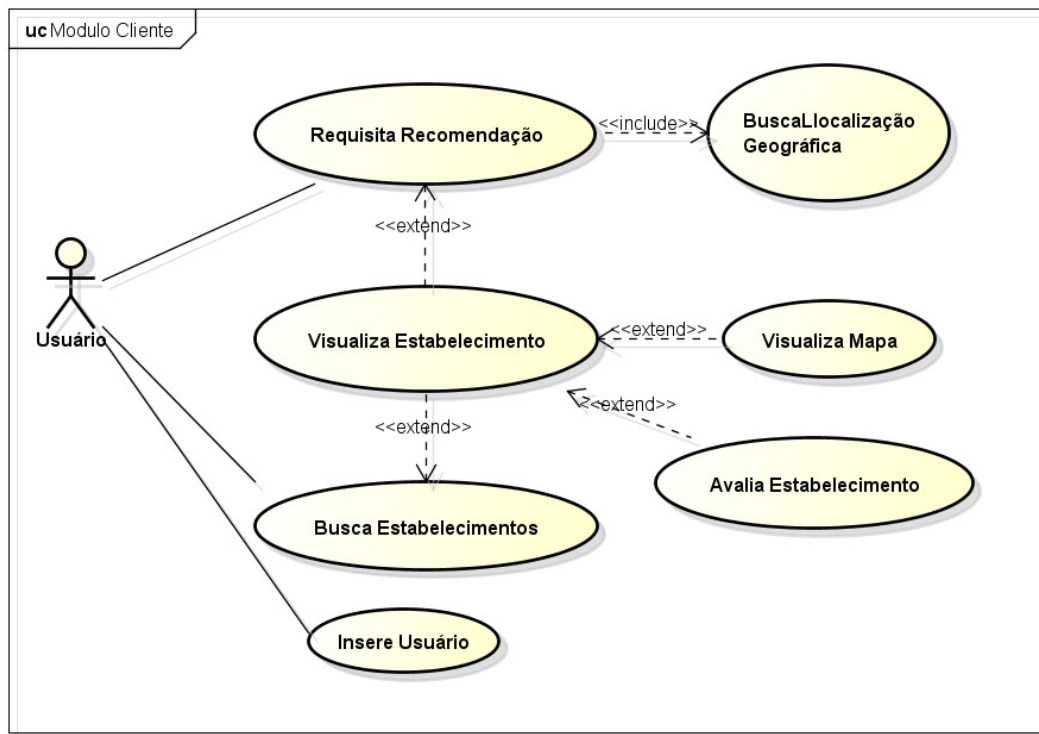


Figura 5.2: Casos de uso do módulo cliente

Conforme demonstrado pela figura acima, no diagrama de casos de uso do módulo cliente, existem três funcionalidades inicializadas diretamente por ações do usuário através da interface do dispositivo. A seguir essas funcionalidades serão descritas:

Requisita Recomendação

É a principal funcionalidade do sistema e nela estão envolvidos os principais processos da recomendação. Abaixo é mostrada na figura as telas e logo uma breve descrição de cada uma delas.



Figura 5.3: Telas da funcionalidade de requisição da recomendação

Na primeira tela, que será a Activity que iniciará no sistema, existirão duas escolhas a serem definidas pelo usuário. O campo “Perímetro de procura” (número 1) deverá conter as opções: Cidade, Bairro e Rua, como podemos ver na segunda tela, que serão usadas como base para delimitar a área de busca dos locais a partir da própria localização do usuário.

No campo “Categoria” (número 2) serão carregadas categorias de lugares que serão fornecidas como opção de recomendação no sistema de acordo com a base de dados (as opções serão por exemplo: Bases e Restaurante, Boates, Cinemas, etc – seguindo o exemplo temos a terceira tela demonstrando a seleção de uma dessas categorias). A principal funcionalidade disso é criar um sistema com uma maior abrangência de categorias.

Quando o usuário acionar o botão “Seguinte” a aplicação requisita uma ação para a funcionalidade “Busca Localização Geográfica” e quando receber o retorno da localização do usuário enviará uma requisição ao módulo servidor juntamente com os dados coletados e então ficará esperando pela resposta do servidor.

A requisição para o Servidor será um arquivo xml que será serializado e deverá seguir o modelo abaixo:

```

<REQUISICAO>
  <USUARIO>ROGER</USUARIO>
  <IMEI>000000--00-000000-0</IMEI>
  <LOCALIZACAO>
    <CIDADE SET="FALSE">PORTO ALEGRE</CIDADE>
    <BAIRRO SET="FALSE">CENTRO</BAIRRO>
    <RUA SET="TRUE">RUA DOS ANDRADAS</RUA>
  </LOCALIZACAO>
</REQUISICAO>

```

Serão enviadas as informações de login do usuário, o IMEI do dispositivo e as informações de sua localização. Cada elemento como cidade, bairro e rua possuirá um atributo “SET” que identificará qual é o perímetro de busca selecionado pelo usuário (somente um deles deve estar setado). Como resposta, o Cliente receberá um xml com a lista de estabelecimentos recomendados com todos os dados necessário para sua localização e a sua média geral de avaliação, seguindo o modelo abaixo:

```

<RECOMENDACAO>
  <LISTA_ESTABELECIMENTOS>
    <ESTABELECIMENTO>
      <NOME>PASTEL DO ZÉ</NOME>
      <ESPECIALIDADE>LANCHERIA</ESPECIALIDADE>
      <NOTA_MEDIA>3,5</NOTA_MEDIA>
      <LOCALIZACAO>
        <CIDADE>PORTO ALEGRE</CIDADE>
        <BAIRRO>CENTRO</BAIRRO>
        <RUA>RUA DOS ANDRADAS</RUA>
        <NUMERO>500</NUMERO>
        <COMPLEMENTO></COMPLEMENTO>
      </LOCALIZACAO>
    </ESTABELECIMENTO>
    ....
    <ESTABELECIMENTO>
      <NOME>HOT-DOG CACHORRÃO</NOME>
      <ESPECIALIDADE>LANCHERIA</ESPECIALIDADE>
      <NOTA_MEDIA>2,9</NOTA_MEDIA>
      <LOCALIZACAO>
        <CIDADE>PORTO ALEGRE</CIDADE>
        <BAIRRO>CENTRO</BAIRRO>
        <RUA>RUA DOS ANDRADAS</RUA>
        <NUMERO>600</NUMERO>
        <COMPLEMENTO></COMPLEMENTO>
      </LOCALIZACAO>
    </ESTABELECIMENTO>
  </LISTA_ESTABELECIMENTOS>
</RECOMENDACAO>

```

Avalia Estabelecimento

Esta funcionalidade tem por objetivo associar uma nota de uma escala de 1 a 5 ao estabelecimento que o usuário visitou, para que na próxima requisição de recomendação do usuário já faça parte de seu perfil. Esta avaliação será transformada em um arquivo xml que será enviado ao módulo servidor para ser integrado ao histórico do usuário. Estará disponível apenas quando o usuário estiver visualizando o estabelecimento através da funcionalidade *Vizualiza Estabelecimento*.

Visualiza Estabelecimento

Esta funcionalidade estará disponível quando o módulo cliente receber a resposta do módulo servidor à requisição de recomendação ou quando o usuário solicitar a funcionalidade de buscar estabelecimentos. Após isso, é apresentado ao usuário uma lista com todos os estabelecimentos. Quando o usuário selecionar um item dessa lista a funcionalidade então será acionada mostrando detalhes daquele estabelecimento.

Visualiza Mapa

Esta funcionalidade mostrará na tela do aparelho a visualização de um mapa que irá apontar a localização do estabelecimento que for selecionado. Estará disponível apenas quando o usuário estiver visualizando o estabelecimento através da funcionalidade *Vizualiza Estabelecimento*.

Utilizará a api do google disponível para android para fazer a visualização no mapa. Com isso o usuário poderá ter uma melhor referência para achar o local que deseja através da visualização das ruas e assim definir uma rota até o estabelecimento a partir do ponto em que ele se encontra.

Busca Localização Geográfica

Esta funcionalidade somente é acionada pela aplicação. Ela acionará a API disponibilizada no android que utiliza o GPS para buscar as coordenadas geográficas. Essa função da API tem a vantagem de poder buscar a localização no formato de endereço urbano e assim ficará mais fácil usar essa informação no sistema. O usuário não visualizará a ocorrência dessa funcionalidade pelo fato de não fazer parte do processo natural de uso do sistema por parte do usuário, porém só depois que esses dados forem recuperados que será possível realizar a requisição de recomendação ao módulo servidor.

Busca Estabelecimentos

Esta funcionalidade tem por objetivo buscar a lista de estabelecimentos que já foram consultados pelo usuário. Esta lista deverá conter apenas os dados disponíveis no banco de dados do módulo cliente e não será necessário realizar nenhuma consulta ao módulo servidor. A função disso é apenas visualizar os dados locais sobre os estabelecimentos com o objetivo de consulta ou de avaliação de um estabelecimento.

Deve ser observado que esta funcionalidade será muito útil para o usuário avaliar um estabelecimento visto que ele não deverá fazer isso logo depois da consulta pela

recomendação, mas somente depois que ele conhecer melhor o estabelecimento. Então será necessário ele realizar uma consulta posterior ao sistema para executar uma avaliação de acordo com seus critérios e preferências.

Inserir Usuário

Esta funcionalidade só existirá quando o usuário iniciar a aplicação pela primeira vez no aparelho. Ela tem o objetivo de realizar a identificação do usuário e a vinculação do aparelho ao seu perfil.

Na primeira vez que o usuário executar a aplicação deverá informar ao sistema alguns dados básicos como o seu nome e o nome para o login. Depois disso o sistema se encarregará de enviar uma mensagem ao servidor informando os dados do usuário e alguns dados do aparelho, como o IMEI (identificador internacional do aparelho móvel), e assim o módulo servidor cadastrará o novo usuário na sua base de dados.

5.4.2 Módulo Servidor

5.4.2.1 Funcionalidades

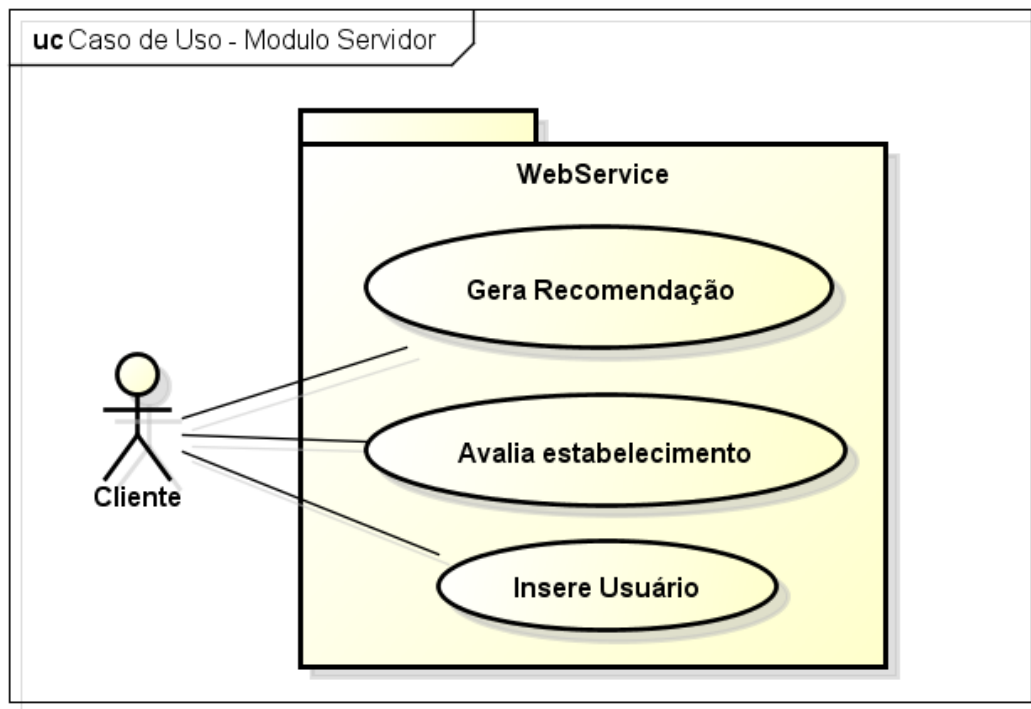


Figura 5.4: Casos de uso do módulo servidor.

Gera Recomendação

É a principal tarefa do sistema. Ela é baseada em um algoritmo de filtragem colaborativa que realiza o processamento dos dados contidos na base de dados do servidor. Esse banco de dados tem o objetivo de centralizar as informações do sistema e fornecer uma estrutura mais simples de pesquisa.

O módulo cliente realiza uma requisição para o webService do módulo servidor com a informação do usuário e de sua localização. Então o servidor utiliza o algoritmo para a recomendação através dos seguintes passos:

- Cálculo dos pesos – O sistema realiza um cálculo de pesos para cada usuário em relação ao usuário alvo utilizando uma métrica de similaridade. Para isso é utilizada a Correlação de Pearson descrita no outro capítulo.
- Procura dos vizinhos – Seleciona-se um subconjunto de usuários com maiores níveis de similaridades (vizinhos) para considerar na predição.
- Realização da predição – Normaliza-se as avaliações fornecidas pelos usuários para o item analisado na recomendação e calcula-se a predição, ponderando-se as avaliações dos vizinhos com seus respectivos pesos de similaridades.

Ao final disso, é criada uma lista com todos os estabelecimentos selecionados pelo algoritmo, bem como a média de avaliações dele pelos usuários, e então é repassado à função que transformará essa lista em um arquivo xml e o enviará ao cliente como resposta.

Avalia Estabelecimento

A avaliação de um estabelecimento é uma função bem simples mas de extrema importância para revelar as preferências de um usuário depois. O Cliente enviará um arquivo xml contendo as identificações do usuário e do estabelecimento, bem como a sua avaliação. Esse dados serão inseridos na base de dados onde possuem as informações de avaliações de estabelecimentos para cada usuário. Se já existir uma avaliação feita anteriormente pelo usuário àquele estabelecimento ela será ignorada e o dado sobrescrito, pois será considerado que o usuário reavaliou aquele estabelecimento.

Inserir usuário

Esta funcionalidade, como foi dita anteriormente no módulo cliente, tem o objetivo de inserir novos usuários ao sistema e associá-los ao seus respectivos dispositivos. Cada usuário poderá estar relacionado com mais de um dispositivo, porém, para uma simplificação, será considerado que um dispositivo pertença a apenas um usuário.

O Cliente então mandará uma requisição com a sua identificação e também o IMEI do dispositivo para realizar a vinculação. O servidor verificará se o usuário já está cadastrado no sistema. Caso já conste no banco de dados só haverá a inserção do dispositivo e sua vinculação com o usuário. Senão será realizada a inserção deste novo usuário também.

5.4.2.2 Modelagem do banco de dados

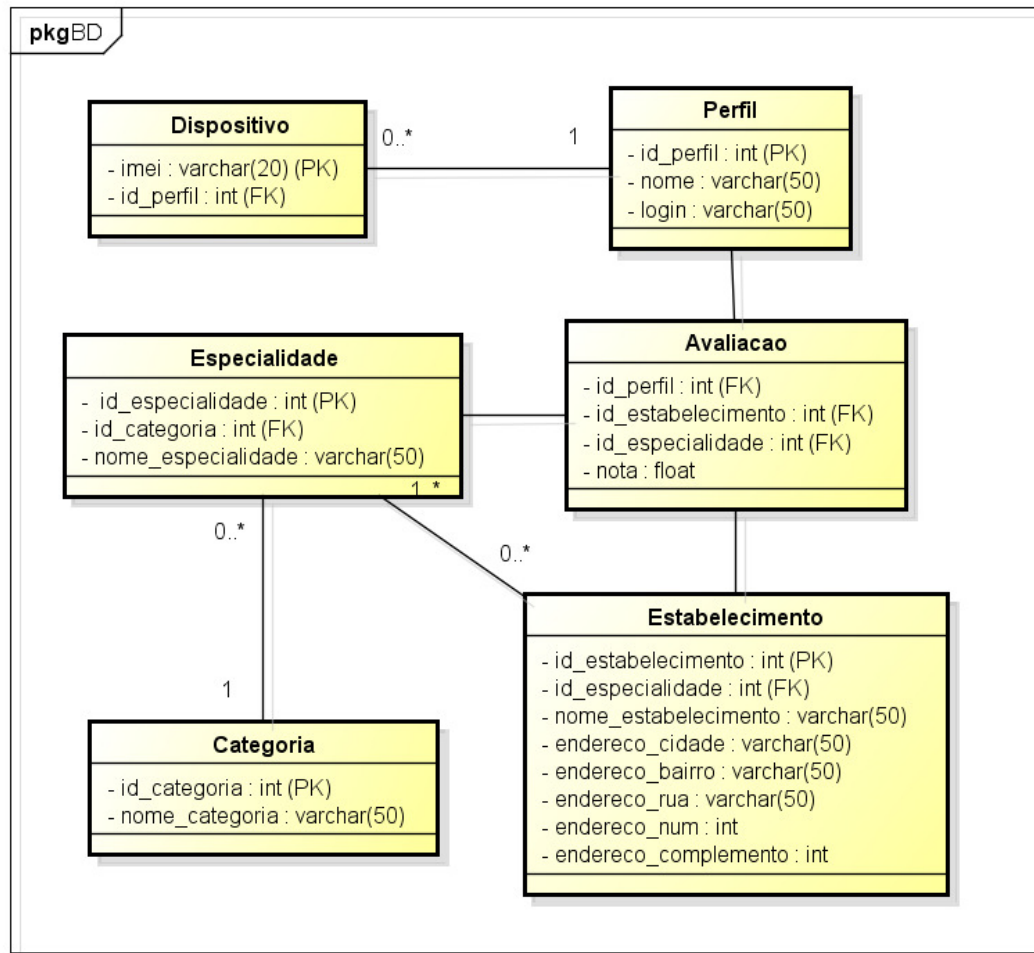


Figura 5.5: Modelo banco de dados do Servidor.

Abaixo segue a relação dos campos e tabelas com as suas respectivas descrições.

Tabela 5.1: Definição dos atributos do banco de dados do Servidor

Tabela	Atributo	Tipo	Tam	Descrição
Dispositivo	imei(PK)	varchar	20	Código de identificação única do dispositivo.
Dispositivo	id_perfil(FK)	int		Código do perfil que o dispositivo está relacionado.
Perfil	Id_perfil(PK)	int		Código de identificação do perfil.
Perfil	nome	varchar	50	Nome do usuário. Não necessariamente o nome completo do indivíduo.
Perfil	login	varchar	50	Login do usuário. Necessário ser único.

<i>Categoria</i>	<i>id_categoria(PK)</i>	<i>int</i>		Código de identificação da categoria.
<i>Categoria</i>	<i>nome_categoria</i>	<i>varchar</i>	50	Nome da categoria.
<i>Especialidade</i>	<i>id_especialidade(PK)</i>	<i>int</i>		Identificação da especialidade.
<i>Especialidade</i>	<i>id_categoria(FK)</i>	<i>int</i>		Identificação da categoria associada à especialidade.
<i>Especialidade</i>	<i>nome_especialidade</i>	<i>varchar</i>	50	Nome da especialidade.
<i>Estabelecimento</i>	<i>id_estabelecimento(PK)</i>	<i>int</i>		Identificação do estabelecimento.
<i>Estabelecimento</i>	<i>id_especialidade(FK)</i>	<i>int</i>		Identificação da especialidade associada ao estabelecimento.
<i>Estabelecimento</i>	<i>nome_estabelecimento</i>	<i>varchar</i>	50	Nome do estabelecimento.
<i>Estabelecimento</i>	<i>endereco_cidade</i>	<i>varchar</i>	50	Nome da cidade referente ao endereço.
<i>Estabelecimento</i>	<i>endereco_bairro</i>	<i>varchar</i>	50	Bairro referente ao endereço.
<i>Estabelecimento</i>	<i>endereco_ rua</i>	<i>varchar</i>	50	Rua referente ao endereço.
<i>Estabelecimento</i>	<i>endereco_num</i>	<i>int</i>		Número do endereço.
<i>Estabelecimento</i>	<i>endereco_complemento</i>	<i>int</i>		Complemento do número do endereço.
<i>Avaliação</i>	<i>id_perfil(FK)</i>	<i>Int</i>		Identificador do perfil associado à avaliação.
<i>Avaliação</i>	<i>id_estabelecimento</i>	<i>Int</i>		Identificador do estabelecimento associado à avaliação.
<i>Avaliação</i>	<i>id_especialidade</i>	<i>int</i>		Identificador da especialidade associado à avaliação.
<i>Avaliação</i>	<i>nota</i>	<i>float</i>		Nota associada à avaliação realizada pelo usuário ao estabelecimento.

5.5 Casos de uso

5.5.1 Consultar Recomendação

Nome: Consultar Recomendação

Atores: Usuário.

Finalidade: Realizar a consulta de recomendação para uma categoria

Visão geral: Este caso de uso se inicia quando o usuário acessa a tela principal da aplicação e deseja realizar uma consulta.

Tipo: Essencial

Tabela 5.2: Caso de uso Consultar Recomendação

5.5.2 Seqüência de eventos	
Ação do ator	Resposta do sistema
1. O usuário inicia por selecionar o perímetro de procura de um estabelecimento. Ele deve decidir se	

quer procurar um estabelecimento nos perímetros da Cidade, Bairro ou Rua.

2. O usuário então seleciona uma categoria que deseja. Como, por exemplo, a categoria “Bares e Restaurantes”.

3. O usuário pressiona o botão consultar para realizar a consulta.

4. O Sistema recolhe os dados selecionados pelo usuário na interface. E enquanto isso é mostrada uma mensagem amigável ao usuário dizendo para que espere um momento até o processo terminar.

5. O Sistema acessa o componente responsável por identificar a localização através do GPS e então espera pela resposta.

6. Ao receber a resposta da localização o Sistema acessa o componente responsável pela comunicação de dados com o Servidor e requisita uma consulta enviando os dados coletados.

7. O módulo de comunicação encapsula essa informação em um arquivo xml e realiza a requisição.

8. Do outro lado, o servidor recebe a requisição e desencapsula os dados.

9. Logo após, o Sistema realiza o processamento da recomendação com base na informações enviadas pelo Cliente.

10. A resposta que contém uma lista de estabelecimentos é ordenada de acordo com a média geral e encapsulada dentro de outro arquivo xml.

11. A parte Cliente do Sistema recebe o arquivo com a resposta e desencapsula os dados para uma lista.

12. A lista então é apresentada na interface e a mensagem de espera é desativada.

13. O usuário visualiza a lista de estabelecimentos recomendados.

14. O usuário seleciona um dos estabelecimentos da lista.

15. O Sistema exhibe todas as informações disponíveis daquele estabelecimento e um botão com a opção de visualização de mapas.

<p>16. O usuário seleciona o botão de visualização de mapas com o objetivo de conseguir tomar a direção para a localização do estabelecimento.</p> <p>18. O usuário visualiza o mapa e então decide se retorna à lista(13) ou se dirigir ao estabelecimento.</p>	<p>17. O Sistema então ativa a biblioteca do Google maps e mostra ao usuário um mapa com a marcação do local do estabelecimento.</p>
--	---

5.5.3 Inserir Usuário

Nome: Inserir Usuário

Atores: Usuário.

Finalidade: Realizar a inserção do usuário e a vinculação do mesmo ao aparelho

Visão geral: Este caso de uso se inicia quando o usuário acessa pela primeira vez o sistema depois de instalá-lo no dispositivo.

Tipo: Essencial

Tabela 5.3: Caso de uso Inserir Usuário

5.5.4 Seqüência de eventos	
Ação do ator	Resposta do sistema
<p>1. O usuário inicia pela primeira vez o sistema</p> <p>3. O usuário preenche o campo de nome e login e confirma os dados.</p>	<p>2. O Sistema ao verificar que é a primeira execução da aplicação mostra uma tela de preenchimento de dados do usuário com os campos de nome e login.</p> <p>4. O Sistema acessa o componente de comunicação e encapsula os dados do usuário em um arquivo xml.</p> <p>5. O Servidor recebe o arquivo e desencapsula os dados.</p> <p>6. Ao verificar que a requisição é de inserção de usuário, acessa o banco de dados e realiza uma consulta pelo login do usuário.</p> <p>a) Caso não exista o sistema realiza a inserção dos dados do usuário juntamente com o imei do dispositivo que será associado.</p> <p>b) Caso já exista um usuário com aquele login, apenas associa o novo imei ao usuário</p> <p>8. O Servidor prepara uma mensagem com a descrição de sucesso ou erro da inserção.</p>

<p>10. O usuário verifica a mensagem e então aperta no botão para confirmar que leu ela.</p>	<p>9. É enviado ao Cliente e este desencapsula a mensagem e a exibe na interface para o usuário.</p> <p>11. O sistema cadastra os dados do usuário na aplicação do dispositivo para não ser mais necessária essa etapa e apresenta a tela principal do Sistema de recomendação ao usuário.</p>
---	--

5.5.5 Avaliar Estabelecimento

Nome: Avaliar Estabelecimento

Atores: Usuário.

Finalidade: Realizar a avaliação de um estabelecimento.

Visão geral: Este caso de uso se inicia quando o usuário acessa a aplicação para avaliar o estabelecimento que ele visitou.

Tipo: Essencial

Tabela 5.4: Caso de uso Avaliar Estabelecimento

5.5.6 Seqüência de eventos	
Ação do ator	Resposta do sistema
<p>1. O usuário inicia a aplicação.</p> <p>3. O usuário acessa o menu de opções e seleciona a visualização da lista de estabelecimentos.</p> <p>6. Ao visualizar a lista o usuário procura por aquela que ele deseja.</p> <p>7. Quando o usuário encontrar ele seleciona o estabelecimento.</p> <p>9. O usuário seleciona a avaliação na nota.</p>	<p>2. O Sistema exibe a tela principal.</p> <p>4. O Sistema acessa o banco de dados local no dispositivo e realiza uma consulta por estabelecimentos.</p> <p>5. O Sistema apresenta então uma lista de estabelecimento na tela do dispositivo.</p> <p>8. O Sistema exibe os detalhes daquele estabelecimento e, se existir, exibe a nota do usuário associada àquele local.</p> <p>10. O Sistema exibe um campo com as notas de</p>

<p>11. O usuário seleciona a nota e confirma.</p>	<p>1 a 5 para o usuário escolher.</p> <p>12. O Sistema então se encarrega de enviar essa avaliação ao servidor através do componente de comunicação.</p> <p>13. Os dados são encapsulados e enviados ao servidor que ao receber os dados envia uma mensagem de sucesso.</p> <p>14. A aplicação confirma a operação com uma mensagem de sucesso ao usuário. Se ocorrer um erro durante a operação será exibida uma mensagem de erro na tela.</p>
--	--

6 CONCLUSÃO

6.1 Análise geral do trabalho

Relembrando a motivação do trabalho, que mostrou como as tecnologias na área de mobilidade têm se desenvolvido e vem se tornando cada vez mais populares no cotidiano das pessoas. Novas tecnologias vem surgindo e outras começam a ser melhores aproveitadas. Este trabalho tinha o objetivo de abordar um tema atual e mostrar como ele poderia se tornar uma ferramenta útil para as pessoas.

Foi realizado um estudo sobre os serviços baseados em localização, que demonstram claramente como um serviço de mobilidade pode oferecer informações que possam agregar algum valor ao usuário. Foram demonstrados como funcionam os sistemas de recomendação, sua importância na filtragem de informação que hoje é necessária pela grande quantidade de dados disponibilizados. A plataforma Android também foi foco de estudo neste trabalho, pela grande oportunidade que deu aos desenvolvedores, que agora podem utilizar uma plataforma Open Source para criar aplicações mais ricas para dispositivos móveis.

O objetivo do trabalho, a partir dessa motivação, foi o de desenvolver o conhecimento através dessas tecnologias inovadoras e propor um modelo de sistema que as utilize e possa ser implementado nos dispositivos móveis atuais. Através desse estudo, foi criado um modelo para utilizar da melhor forma possível as ferramentas disponíveis e descobriu-se que a melhor forma de fazer isso era a de dividir o sistema em duas partes, uma parte cliente que rodaria no dispositivo móvel, e outra parte servidor que seria responsável por centralizar todos os dados do sistema com a finalidade de poder criar as recomendações que é a principal funcionalidade do sistema.

6.2 Possibilidades futuras

Através de todo o estudo e elaboração do modelo do sistema neste trabalho, foram imaginadas outras possibilidades que essas tecnologias podem gerar para o estudo e desenvolvimento de outros trabalhos nesta área.

Analisando características do sistema proposto, pode-se perceber que existem alguns problemas relacionados com as decisões que foram tomadas. Um deles é o de utilizar a filtragem colaborativa para a recomendação de estabelecimentos, pois com isso surge o problema já mencionado neste trabalho da “partida a frio”, que nada mais é do que a ocorrência de problemas no algoritmo de recomendação quando o usuário ainda não possui nenhum estabelecimento avaliado ou não possui avaliações suficientes para criar o seu perfil de preferências. Pode-se então estender o modelo proposto para criar uma

filtragem híbrida que utilize características da filtragem baseada em conteúdo para construir as recomendações.

Uma outra oportunidade que surge é a criação de uma aplicação mais inteligente para recomendar estabelecimentos sem que o usuário precise acioná-la. Ela utilizaria aprendizagem para identificar certos padrões de comportamento do usuário. Como, por exemplo, que ele sempre almoça nas segundas-feiras num mesmo tipo de lugar, então o sistema lhe alertaria sobre lugares parecidos quando ele se aproximasse de um deles. Outra função interessante a ser incorporada num sistema desses é a de sugerir promoções que existirem em um dia, que esteja de acordo com suas preferências.

Outra idéia seria a de utilizar redes sociais e outros sites para realizar a coleta de informações de usuários para que a base de dados fosse mais completa. Sabendo que é preciso possuir uma base com muitas informações para que as recomendações se tornem realmente relevantes para o usuário, poderia ser criado um sistema que permitiria a comunicação com vários outros sites que realizam a interação de usuários, como são os casos do Facebook, Twitter e etc.

REFERÊNCIAS

PAZZANI, MICHAEL J. **A Framework for collaborative, content-based and demographic filtering.** Artificial Intelligence Review, Vol. 13, Issue 5. Kluwer Academic Publishers, Hingham, MA, USA, (1999), pp. 393-408.

SANDVIG , JEFF J. et al **A Survey of Collaborative Recommendation and the Robustness of Model-Based Algorithms.** IEEE Data Eng. Bull. 31(2): 3-13 (2008).

HERLOCKER, J., KONSTAN, J., BORCHERS, A., RIEDL, J. **An algorithmic framework for performing collaborative filtering.** In Conference on Research and Development in Information Retrieval, 1999.

HERLOCKER, J., KONSTAN, J. e RIEDL, J. **Explaining Collaborative Filtering Recommendations.** In Proceedings of ACM 2000 Conference on Computer Supported Cooperative Work, 2000.

FILHO, WAGNER DANDA da SILVA. CAZELLA, SÍLVIO CÉSAR. **STAR: Um Framework para recomendação de artigos científicos baseado na relevância da opinião dos usuários e em filtragem colaborativa.** SBC 2005 – XXV Congresso da Sociedade Brasileira de Computação.

FILHO, WAGNER DANDA da SILVA. **Um Framework para recomendação de artigos científicos baseado na relevância da opinião dos usuários e em filtragem colaborativa.** 2004. 56 f. Projeto de Diplomaciação (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Tecnológicas, Unisinos, São Leopoldo.

CAZELLA, SÍLVIO CÉSAR. **Um Modelo para Recomendação de Artigos Acadêmicos Baseado em Filtragem Colaborativa Aplicado à Ambientes Móveis.** Dezembro (2008)..

CHEN, TING. et al. **Content Recommendation System Based on Private Dynamic User Profile.** Agosto (2007).

QUEIROZ, RODRIGO BARBOSA de. **Serviços Baseados em Localização (LBS) Proposta de um Framework para Implantação de LBS em Aparelhos Celulares.** 2006. 49 f. Projeto de Diplomaciação (Bacharelado em Sistemas de Informação) – Unimontes, Montes Claros.

REATEGUI, ELISEO BERNI. CAZELLA, SÍLVIO CÉSAR. **Sistemas de Recomendação.** XXV Congresso da Sociedade Brasileira de Computação, 2005. São Leopoldo.

FERRARO, RICHARD; AKTIHANONOG, MURAT. **Location Based Services Past, Present, and Future.** Green paper from Location Based Services Navigating through the Mobile Jungle, janeiro, 2010.

BALABANOVIC, M., SHOHAM, Y.: Fab: **Content-Based, Collaborative Recommendation**. Comm. ACM, vol. 40, no. 3, (1997) 66-72.

SNEKKENES, EINAR. Concepts **for personalk location privacy policies**. In ACM Conference on Electronic Commerce (EC'01), 14-17 Outubro 2001 Tampa, Florida, USA., pages 48-57. acm press, 2001.

CHEN, YING. et al. **e-Business Engineering**, 2005. ICEBE 2005. IEEE International Conference. (2005) pp. 419-426.

SHARDANAND, U., MAES, P. **Social information filtering: Algorithms for automating "word of mouth"**. In **Proceedings of Human Factors in Computing Systems (CHI '95)**, Denver, Colorado, USA, (1995), pp. 210-217.

SOBOROFF, IAN. NICHOLAS, Charles. **Collaborative filtering and generalized vector space model**. In: Annual Conference on Research and Development in Information Retrieval, 2000. Atenas, Grécia.

KUPPER, AXEL. **Loaion-based Services: Fundamentals and Operation**. John Wiley & Sons Ltd, 2005.

SCHAFER, J. B., KONSTAN, J., RIEDL, J. **Recommender systems in ECommerce**. In Proceedings of the ACM Conference on Electronic Commerce (EC'99), N. Y., USA, (1999), pp. 158-166

BALABANOVIC, M., SHOHAM, Y.: Fab: **Content-Based, Collaborative Recommendation**. Comm. ACM, vol. 40, no. 3, (1997) 66-72.

GOOGLE, **Android Developers**. Disponível em <<http://developer.android.com/>> Acesso em dezembro de 2010.

GARTNER, **Gartner Research**. Disponível em <<http://www.gartner.com/it/page.jsp?id=1434613/>> Acesso em dezembro de 2010.