

SISTEMA DE RECOMENDAÇÃO INTELIGENTE MÓVEL PARA COMPRAS

INTELLIGENT RECOMMENDATION SYSTEM FOR SHOPPING APPS

Jéssica Fernanda Galvani*

Maurício Acconcia Dias**

RESUMO

O acesso à informação é uma realidade e trouxe vários benefícios e também desvantagens a quem busca estas informações. Considerando o grande volume de informações de praticamente qualquer assunto buscado na internet atualmente tornou-se necessário que exista algum auxílio com relação a qual informação buscar, onde buscar esta informação e principalmente a busca por informações diretamente relacionadas à informação buscada. Atualmente existem sistemas que previamente relacionam as informações e quando determinado assunto é buscado por um determinado usuário o retorno da busca apresenta não só a informação buscada como também resultados para informações relacionadas. Estes sistemas são chamados de sistemas de recomendação. Considerando a situação de uma lista de compras onde muitas vezes são relacionados produtos para serem adquiridos, a probabilidade de um produto que sempre é comprado juntamente com os demais não ser colocado na lista e ser esquecido é alta. Este trabalho de pesquisa propõe um sistema de recomendação que ao se criar uma lista de produtos, outros produtos são sugeridos ao usuário com base em compras anteriores. O sistema é embarcado, pois foi desenvolvido para ser executado em plataformas móveis. Os resultados iniciais demonstram que o sistema tem potencial e que o grau de acerto das sugestões depende da quantidade de informações que o software possui e também que cada conjunto de informações deve ser associado a apenas um usuário para que suas características sejam mais facilmente identificadas pelo software.

Palavras-chave: Sistema de Recomendação. Android. Embarcados. Compras.

ABSTRACT

Access to information is a reality and brought many benefits and also disadvantages for those who use the information. Considering the large volume of information from almost any topic searched on the Internet today, having some assistance on what information to seek, where to get this information and especially the search for information directly related to the information sought. has become necessary. Systems that previously relate the information and when a particular issue is searched by a given user, search result has not only information directly related to the searching topic but also results for related information previously considered. These systems are called recommender systems. Considering the situation of a shopping list where listed products are related to other products not listed but also needed, the probability of a product that is always bought together with other is not listed and needed

* Faculdade de Tecnologia, Ciências e Educação (FATECE). jfgalvani1@hotmail.com

** Faculdade de Tecnologia, Ciências e Educação (FATECE). macdias@icmc.usp.br

is high. Based on this scenario, this research work proposes a recommendation system that when list of products is created, other related products are suggested to the user based on previous purchases. The idea of an embedded system designed for mobile operating systems is a consequence of the high number of virtual list apps. Initial results show that the system has the potential and the degree of accuracy of the suggestions depends on the amount of information available for the software, and also that each set of information must be associated with only one user to enhance system capabilities of learning the user preferences.

Keywords: Recommendation System. Android. Embedded System. Shopping

Introdução

Atualmente existe uma quantidade considerável de informações disponíveis na internet e existe uma crescente demanda por aplicativos e sites que sejam capazes não só de filtrar esta quantidade de informação, mas também implementar algum tipo de sistema que faça sugestões de conteúdo independentemente do tipo deste conteúdo. Sistemas que possuem esta habilidade são chamados de sistemas de recomendação. Como é uma área de pesquisa relativamente nova datando do início dos anos 90 (RICCI; ROKACH; SHAPIRA, 2011), apesar destes sistemas se desenvolverem praticamente junto com a internet, existem diversos algoritmos em desenvolvimento oferecendo diferentes resultados.

Os sistemas de recomendação auxiliam no aumento da capacidade e eficácia do processo de indicação já bastante conhecida na relação social entre as pessoas (RESNICK; VARIAN, 1997). Em sistemas de recomendação básicos, o sistema coleta informações e redireciona para potenciais usuários. Um dos principais desafios é relacionar as recomendações certas para cada pessoa. Resnick e Varian (1997) propõem que o termo sistema de recomendação é o mais indicado (em comparação com outro termo para a mesma finalidade, filtro colaborativo) por dois motivos, o primeiro é que a pessoa que recebe a recomendação e a pessoa que gerou o banco de dados que auxiliou na recomendação podem nem se encontrar, e segundo que as recomendações são itens particulares e filtram o que deve e o que não deve ser recomendado. Segundo Manber (2000), a primeira grande utilização de sistemas de recomendação foi feita pelo site *My Yahoo* em julho de 1996. Atualmente vários sites, sistemas, aplicativos de celular utilizam sistemas de recomendação para as mais diversas finalidades.

O aumento considerável na venda de smartphones nos últimos anos¹ criou uma demanda por desenvolvimento de aplicativos que facilitem o dia a dia de seus usuários. Estes aplicativos incluem aplicativos de agenda, aplicativos de compra, de serviços e também de organização.

Baseado neste cenário os aplicativos estão incluindo em suas funcionalidades os serviços de recomendação. Estes serviços são basicamente serviços que recomendam itens ao usuário baseado em informações obtidas pela utilização do software e de outros softwares relacionados. Algumas vezes estes aplicativos funcionam de forma indesejada obtendo informações de softwares que deveriam manter seu conteúdo privado. Os serviços de recomendação são destinados aos usuários que possuem experiência pessoal insuficiente em determinados assuntos ou serviços oferecidos (RICCI; ROKACH; SHAPIRA, 2011).

A lista de aplicativos mais baixados para o sistema operacional Android², por exemplo, demonstra que a maioria envolve redes sociais, jogos e o primeiro da lista que se enquadra na categoria de escritório é o pacote Office. Apesar da preferência ser pelos tipos de software citados, a maioria destes aplicativos possui sistemas de recomendação integrados, seja de notícias, produtos, propagandas. E os usuários utilizam o serviço muitas vezes sem ao menos perceber. Exemplos de software de recomendação de produtos podem ser vistos no trabalho de Lua et al. (2012).

Anteriormente não existiam estudos que apresentassem um detalhamento das técnicas e resultados dos diferentes tipos de sistemas de recomendação existentes (LUA et al., 2012). Devido a popularização e aumento da demanda por este tipo de sistemas vários autores disponibilizaram estudos sobre as técnicas utilizadas e o resultado como mostra o trabalho de Beel et al. (2013).

Esta nova área de pesquisa se desenvolveu muito nos últimos anos devido ao grande interesse de empresas como Youtube, Netflix, Yahoo, dentre outras, devido ao surgimento de conferências dedicadas a este assunto, cursos de computação por todo o mundo que priorizam os sistemas de recomendação em seus cursos e também a edições especiais de revistas importantes da área de computação (RICCI; ROKACH; SHAPIRA, 2011).

¹ <http://www.wow5stars.com/best-current-smartphone/>

² https://en.wikipedia.org/wiki/List_of_most_downloaded_Android_applications

Considerando a necessidade atual de desenvolvimento de aplicativos utilitários que incorporem os conceitos de recomendação de informações, juntamente com a crescente demanda por softwares em um mercado crescente de venda de dispositivos móveis, este trabalho propõe o desenvolvimento de um sistema de recomendação para compras onde o usuário faz uma lista de compras e o sistema propõe itens relacionados. Estes itens relacionados podem ser obtidos de várias maneiras possíveis como internet, histórico de listas, sugestões, dentre outros. O software será desenvolvido de tal maneira a suportar uma interface cliente-servidor que futuramente pode se conectar ao estoque de supermercados e outras lojas e já permitir que o usuário apenas faça listas de produtos que estejam disponíveis para compra. O objetivo principal do sistema é proporcionar um ambiente de criação de listas, onde durante a criação o usuário receba recomendações de qual item comprar.

Este artigo está dividido da seguinte maneira: a seção 1 apresenta a teoria sobre sistemas de recomendação e sua classificação, a seção 2 apresenta os conceitos relacionados ao desenvolvimento de aplicativos móveis, seguido pela apresentação do sistema desenvolvido nos resultados e análises da seção 3 e considerações finais.

1 Sistemas de Recomendação

Segundo Ricci, Rokach e Shapira (2011), um sistema de recomendação é uma ferramenta de software que possui técnicas que permitem ao sistema sugerir itens de interesse a um determinado usuário. Item é um termo geral utilizado para descrever o que é recomendado pelo sistema. Estes sistemas são inicialmente direcionados a usuários que não possuem experiência suficiente para avaliar um grande número de alternativas em determinada circunstância. Alguns aspectos definem as características dos sistemas de recomendação como, por exemplo, a forma como as informações são coletadas, as estratégias utilizadas para recomendar novos itens, as técnicas de filtragem de informação utilizadas para gerar as recomendações e também a arquitetura e os aspectos de implementação do sistema. Este capítulo irá apresentar detalhadamente cada uma destas características.

1.1 Coleta de Informações

Um sistema de recomendação possui seu princípio básico na extração de conhecimento a partir de informações previamente apresentadas ao sistema. Portanto é necessário coletar estas informações de alguma forma.

Inicialmente o usuário do sistema é identificado. Esta identificação pode ser feita de forma local onde o usuário entra com seu *login* e senha, ou então de forma remota onde o *login* e senha do usuário são enviados a um servidor que faz a autenticação. Após o *login* as informações do usuário podem ser obtidas de forma explícita ou implícita. Na forma explícita o usuário fornece as informações do que interessa espontaneamente. No caso da obtenção de informações de forma implícita, o sistema de recomendação obtém informações do usuário utilizando suas preferências e suas necessidades. Através desta técnica os usuários podem ser identificados sem que tenham fornecido diretamente uma informação sequer. Após ser coletado um conjunto grande de informações, estas informações precisam ser filtradas.

1.3 Técnicas de Filtragem

A quantidade de informação gerada e disponível sugere que seja utilizado um filtro para realmente encontrarmos o que é de interesse. Segundo Blekin e Croft (1992), filtragem de informação é um nome utilizado para descrever uma grande variedade de processos envolvidos no processo de se entregar informações a pessoas que precisam dela.

Geralmente a filtragem ocorre em dados pouco ou não estruturados, e lida com informações textuais. A quantidade de dados geralmente é grande e envolve trabalhar com *streaming* de informações. A ideia principal é selecionar informação para ser removida e não o controle das informações adicionadas. A seguir serão descritas algumas técnicas de filtragem de informações.

1.3.1 Filtragem baseada em conteúdo

A ideia da filtragem baseada em conteúdo é gerar informações sobre itens e comparar com informações obtidas pela análise de escolhas feitas pelo usuário. Os interesses do usuário são obtidos por informações cedidas pelo próprio, ou então por uma análise de suas ações como seleção e aquisição de itens. Por exemplo, caso o usuário necessite de uma busca de

palavras em documentos, a frequência que cada palavra aparece pode ser armazenada em um vetor de indexação de termos. Os vetores que estão mais próximos do usuário são os mais relevantes para ele. Outras técnicas são a busca booleana baseada em uma árvore construída com os termos da busca, filtragem probabilística quando para cada item uma probabilidade de ser relevante para a pesquisa é associada a ele, dentre outras. Um sistema especialista pode ser utilizado nestes casos, onde o usuário classifica alguns itens e o sistema busca itens semelhantes aos que o usuário classificou como de interesse.

1.3.2 Filtragem colaborativa

O sistema de filtragem colaborativa é baseado na avaliação de usuários. Quando é feita a busca por um item, os melhores avaliados diretamente relacionados ao item são considerados as respostas para a busca. O sistema de pontuação, que coleta a avaliação dos usuários, evoluiu para de forma automática retornar os resultados da avaliação itens fazendo a média das avaliações de todos os usuários. Os sistemas de filtragem colaborativa mais elaborados descobrem relações entre usuários de acordo com as notas atribuídas aos itens e encontra relações entre itens pelos gostos similares de usuários. Nestes sistemas alguns problemas são encontrados:

- Quando um item é adicionado ao banco de dados, ele precisa ser avaliado por uma pessoa para que possa fazer parte do sistema. Se ninguém avaliar o item ele ficará de fora das recomendações
- Quando existe um número muito grande de dados no sistema e um pequeno número de usuários as pontuações podem ficar esparsas e o sistema pode não conseguir criar as relações
- Caso os usuários tenham gostos que variam do normal o sistema pode ter problemas em recomendar itens para estes usuários

1.3.3 Filtragem híbrida

A filtragem híbrida utiliza as duas técnicas anteriores para encontrar melhores resultados. Para atingir este objetivo no caso de usuários comuns a filtragem colaborativa é

utilizada e no caso dos problemas da filtragem colaborativa a filtragem baseada em conteúdo é utilizada. Os problemas de não se ter uma classificação, poucas classificações ou usuários muito diferentes da média são resolvidos pois os itens são classificados por suas relações independente dos usuários.

1.3.4 Descoberta de conhecimento em base de dados

Quando não se pode utilizar informações de usuários ou não se tem as mesmas disponíveis, a ideia é utilizar técnicas de mineração de dados, dentre as mais utilizadas podemos citar: regras de associação, classificação e agrupamento (HAN, 2005). Após os dados serem filtrados, são utilizadas estratégias para recomendação de itens.

1.4 Estratégias de Recomendação

Diferentes estratégias podem ser utilizadas para recomendar itens para usuários, cada um exigindo um grau de complexidade diferente. Dentre algumas estratégias podemos citar:

- Listas de recomendação: são mantidas listas de itens organizados por tipos de interesses. O nível de complexidade é baixo, pois apenas é necessário organizar pelos mais populares. A principal vantagem é a facilidade de implementação e a principal desvantagem é que não existe especialização da lista para cada usuário.
- Avaliações de usuários: a opinião do usuário sobre o item é mostrada de acordo com algum gráfico ou indicador ao lado do item. Este sistema é igualmente simples, porém necessita de informações verdadeiras por parte de usuários.
- Recomendações específicas: este tipo permite que sejam feitas recomendações para um usuário específico. São basicamente de dois tipos: implícitas que são feitas baseadas em escolhas anteriores do usuário, e as explícitas que são escolhas feitas pelo próprio usuário.
- Recomendação por associação: são recomendações feitas nos casos onde um item foi escolhido em conjunto com outro, como no exemplo “usuários que

compraram X também compraram Y”. Esta forma é a mais complexa, pois exige uma análise profunda dos hábitos do usuário para a identificação de padrões e recomendação de itens com base nos padrões.

Um determinado sistema de recomendação pode apresentar mais de uma opção de estratégia de recomendação baseando-se nas necessidades dos usuários. Após definidas as estratégias o sistema então pode ser implementado.

1.5 Arquiteturas e Implementação

Considerando os diferentes tipos de implementação de filtros e de estratégias de recomendação é possível dividir os sistemas de recomendações em duas arquiteturas básicas (como ilustrado na Figura 1): os baseados em técnicas de filtragem de informação que trabalham diretamente sobre as informações dos usuários, sem a construção de um modelo de recomendações generalizado, e o segundo tipo onde há um módulo de mineração de dados que é responsável por encontrar padrões baseados nos perfis dos usuários, que são armazenados em uma base de conhecimento.

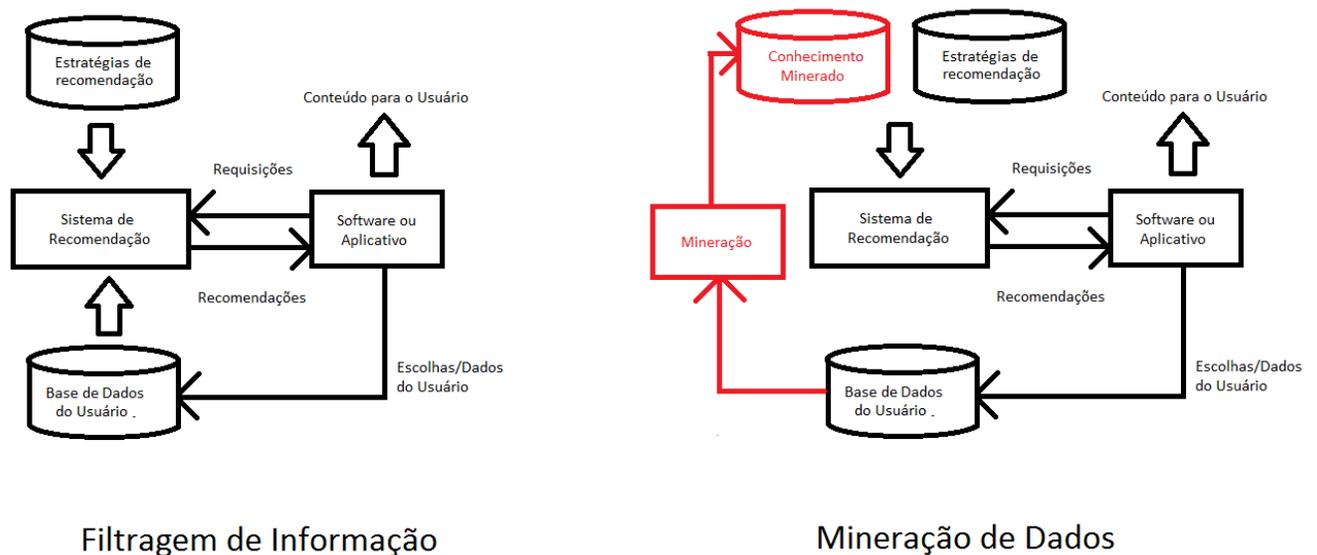


Figura 1 – Arquitetura de sistemas de recomendação. A primeira arquitetura demonstra os sistemas que não possuem mineração de dados, e a segunda os sistemas que são baseados em mineração de dados.

O sistema de recomendação desenvolvido neste trabalho de pesquisa terá foco em sistemas operacionais de dispositivos móveis. A ideia é recomendar produtos em uma lista criada no dispositivo pelo usuário. O desenvolvimento para dispositivos móveis possui algumas particularidades que serão discutidas na próxima seção.

2 Desenvolvimento Móvel

Atualmente um grande número de aplicativos está sendo desenvolvido para plataformas móveis. A questão do código aberto de sistemas operacionais móveis como o Android³ e a utilização de linguagem JAVA para seu desenvolvimento têm atraído a atenção de muitos desenvolvedores, porém o grande problema é que apesar de toda a tecnologia envolvida em um dispositivo móvel ele ainda possui limitações e alguns cuidados precisam ser tomados para que o programa funcione de forma correta sem prejudicar o sistema como um todo.

2.1 Características de sistemas e dispositivos móveis

Os dispositivos móveis podem ser considerados, não totalmente, como sistemas embarcados. Sistemas embarcados são sistemas desenvolvidos para propósito específico combinando interfaces de hardware e software. Sendo assim, possuem hardware limitado e são projetados para realizar tarefas específicas. Atualmente os *smartphones* possuem diversos tipos de recursos que agregam vários dispositivos em um só, porém ainda existem sérios problemas como qualidade de sinal, tempo de duração de bateria enquanto executamos atividades, durabilidade do hardware, temperatura, dentre outros.

A funcionalidade principal do celular de ser um dispositivo móvel que permite chamadas de qualquer lugar com uma autonomia aceitável está sendo prejudicada pela presença de todos os outros recursos. A bateria do celular dura menos tempo pois os aplicativos e a câmera consomem muita energia, a qualidade da chamada fica prejudicada, pois o celular está executando milhares de operações em plano de fundo enquanto tenta

³ <https://www.android.com/>

processar a chamada. Neste caso, o desenvolvimento de sistemas para dispositivos móveis deve levar em conta fatores como:

- Pouca quantidade de memória principal disponível que é dividida entre um número considerável de aplicações
- Consumo de energia deve ser controlado pela quantidade de processamento executado pelo software
- A quantidade de dados utilizada e armazenada pelo software não pode exceder uma determinada quantidade que pode prejudicar o desempenho do sistema
- Os sistemas operacionais de *smartphones* evoluem constantemente, portanto o desenvolvimento deve priorizar as versões do sistema que dão suporte a versões mais antigas
- A interface é normalmente constituída de uma tela *touchscreen* e poucos botões, sendo que deve ser desenvolvida para agregar todas as funcionalidades necessárias sem perder características de acessibilidade e adaptabilidade
- O software desenvolvido para celulares é normalmente distribuído em repositórios de aplicativos. Estes repositórios possuem regras a serem seguidas pelos aplicativos para que possam ficar disponíveis e, muitas vezes, um custo associado

O sistema operacional que está presente na maioria dos dispositivos móveis atuais é o sistema Android. A evolução do sistema ao longo de suas versões pode ser utilizada para entender melhor como se comporta o sistema e quais as suas principais funcionalidades.

2.2 O sistema operacional Android

O sistema operacional Android está presente em mais de 80% dos dispositivos móveis em média desde 2013⁴. Este fato faz com que seja escolhido como a principal plataforma-alvo no desenvolvimento de software para dispositivos móveis. A importância desta questão

⁴ <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

fica evidente com a análise de vendas de dispositivos, já que os usuários preferem aparelhos que possuam uma grande quantidade de aplicativos disponíveis, fenômeno conhecido como *software GAP*⁵.

Porém, a ideia inicial do sistema Android não foi exatamente ser um sistema operacional de dispositivos móveis. A empresa Android INC. criada por Rubin, Miner, Sears e White em Palo Alto na Califórnia tinha como objetivo o lançamento de um novo sistema para câmeras digitais⁶. O mercado, entretanto, mostrou que a opção por dispositivos móveis seria mais rentável. Todo este processo, juntamente com a aquisição da empresa pela Google⁷ em 17 de agosto de 2005⁸, acelerou o desenvolvimento do sistema que foi lançado em sua primeira versão comercial em 2007 em sua versão *Alpha*.

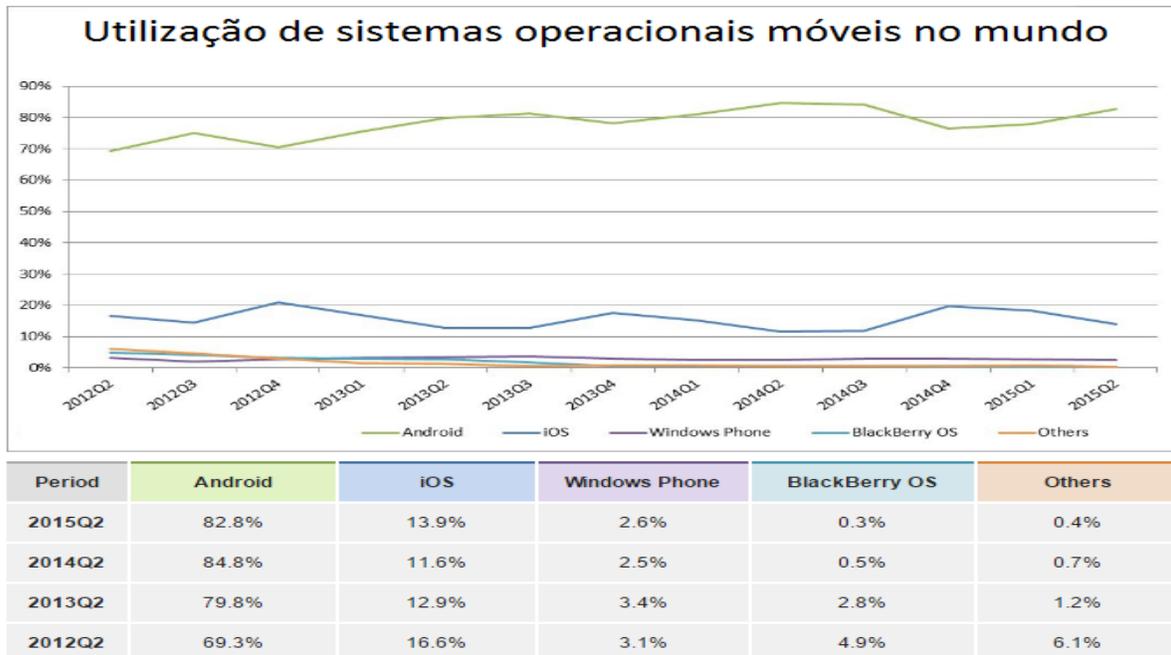


Figura 2 – Comparação entre a utilização de sistemas operacionais móveis entre os anos de 2012 e 2015

Fonte: Adaptado de IDC4

Foram lançadas a partir de então diversas versões com soluções para problemas das versões anteriores e também adaptações ao desenvolvimento do hardware de sistemas móveis. Uma das principais vantagens do sistema Android é ser baseado no Kernel do Linux

⁵ <http://www.windowscentral.com/untold-app-gap-story-part-ii-app-gap-what-app-gap>

⁶ <https://www.android.com/history/#/marshmallow>

⁷ <https://www.cnet.com/news/google-buys-android/>

⁸ https://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions_by_Alphabet

e ser aberto permitindo que seja modificado e adaptado pelos usuários, como fazem empresas como as fabricantes Samsung e Xiaomi que adaptaram o sistema para seus dispositivos.

Assim como o desenvolvimento do sistema é aberto, o desenvolvimento de *software* para o sistema também é, sendo que existem diversas ferramentas que podem ser utilizadas para o desenvolvimento.

2.3 Ferramentas de desenvolvimento para Android

O desenvolvimento de *software* para dispositivos móveis, também conhecidos como aplicativos móveis ou ainda *mobile apps*, cresceu significativamente por diversos fatores, dentre eles o alto número de pessoas que utilizam *smartphones* e a previsão de que o número de usuários de sistemas *desktop* vai cair 30% em relação ao número de usuários de dispositivos móveis em 2016⁹.

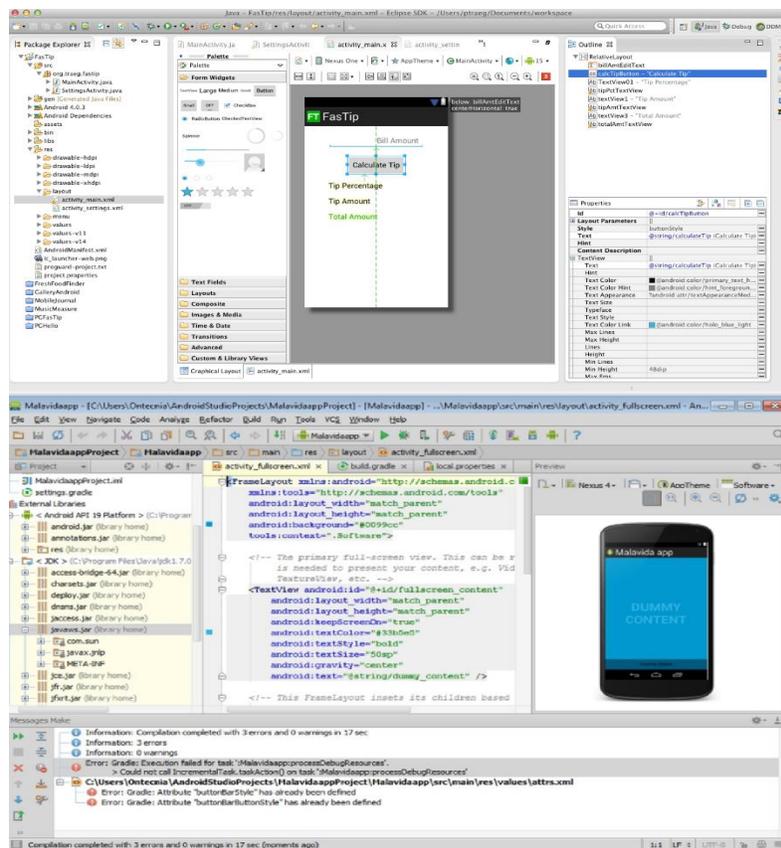


Figura 3 – Comparação entre a interface do *plugin* ADT para eclipse e da IDE Android SDK.

⁹ <http://www.cmdagency.com/13-search-statistics-convince-design-mobile-first/>

A demanda por aplicativos fez crescer o interesse de desenvolvedores independentes e empresas de desenvolvimento em suprir esta demanda. O desenvolvimento para aplicativos móveis necessita de compiladores específicos e da possibilidade de se visualizar o desenvolvimento já que a interface é diferente e limitada em comparação aos sistemas para *desktops*. Diversas ferramentas foram desenvolvidas para facilitar o desenvolvimento de aplicativos móveis e algumas serão apresentadas a seguir.

A IDE (*Integrated Development Enviroment*) mais conhecida para o desenvolvimento de aplicativos é o *Android Studio*¹⁰. A ferramenta apresenta um ambiente integrado onde é possível desenvolver o aplicativo, compilar, simular sua utilização e ainda gerar o aplicativo que será utilizado no celular. Esta é a opção oficial de desenvolvimento segundo o Google. Outra opção para os desenvolvedores que já utilizam a IDE Eclipse é configurar em seu ambiente o SDK do Android e então instalar o ADT *plugin*¹¹. Desta forma é possível desenvolver para Android diretamente da IDE Eclipse. A Figura 3 apresenta uma comparação entre os ambientes de desenvolvimento que são semelhantes sendo, portanto, uma questão de escolha.

Uma alternativa ao desenvolvimento utilizando IDEs é o desenvolvimento online. Atualmente estão disponíveis sites para a criação de aplicativos onde o usuário cria uma conta e desenvolve sem a necessidade de instalar *softwares* de desenvolvimento. A questão principal nestes casos é que todos possuem uma contrapartida como, por exemplo, a inclusão de propagandas no aplicativo desenvolvido e também a quantidade de sub menus desenvolvidos sendo necessário, portanto, adquirir uma licença de desenvolvimento. Dentre os sites podemos citar como referência a Fábrica de Aplicativos¹² e também o Easy Easy Apps¹³. A Figura 4 apresenta uma comparação entre os ambientes de desenvolvimentos dos dois sites que também são similares.

Sistemas de recomendação são utilizados em diversas situações atualmente, portanto é necessário analisar os trabalhos relacionados para entender a importância deste tipo de sistema e também do sistema proposto neste trabalho de pesquisa.

¹⁰ <https://developer.android.com/studio/index.html?hl=pt-br>

¹¹ <http://tools.android.com/recent/eclipseadtbundleupdated>

¹² <http://fabricadeaplicativos.com.br/>

¹³ <https://www.easyeasyapps.net/>

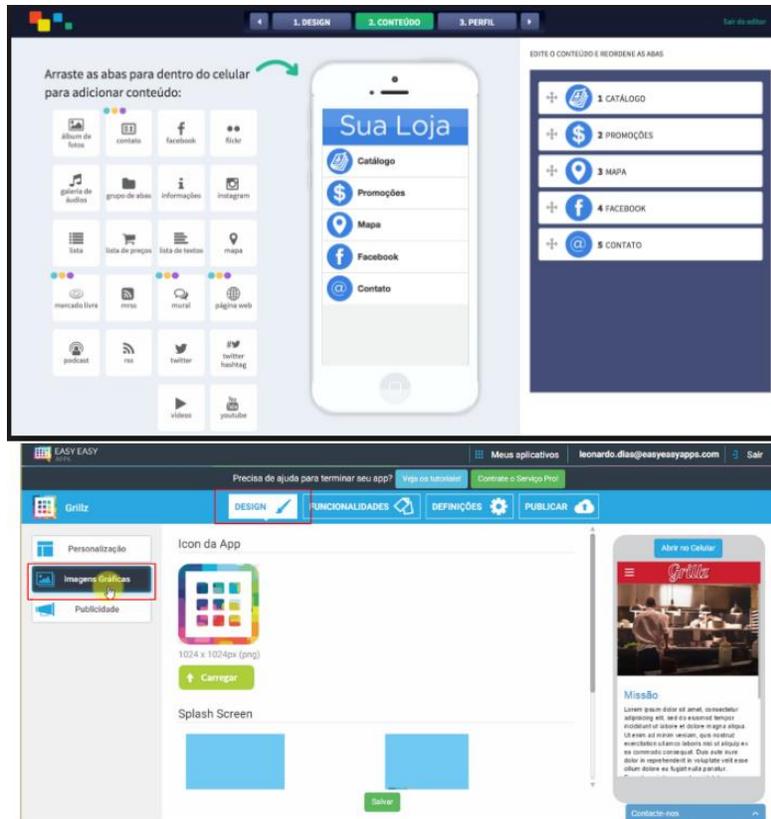


Figura 4 – Comparação entre a Fábrica de Aplicativos e o Easy Easy Apps.

3 Trabalhos Relacionados

Existem diversos tipos de sistemas de recomendação, porém cada sistema é desenvolvido de acordo com os dados que serão trabalhados. Este trabalho utiliza um sistema de filtragem baseada em conteúdo, ou seja, o sistema tenta recomendar itens ao usuário com base em informações adicionadas e classificadas no passado (BOBADILLA et al., 2013).

Ao procurar por sistemas de recomendação disponíveis nos repositórios de aplicativos *on-line* é possível perceber que não há um desenvolvimento de aplicativos voltados para esta finalidade. Diversos aplicativos de lojas online como Amazon, Mercado Livre, OLX, livrarias, dentre outros possuem internamente implementados sistemas de recomendação que, baseado nas visualizações e/ou compras realizadas pelos usuários retorna itens de interesse.

Considerando o estilo de aplicativo desenvolvido neste trabalho, outra busca interessante é por aplicativos de listas. Existem diversos aplicativos implementados para a

construção de listas, dentre eles os melhor avaliados na Google Play Store são: Lista de Tarefas¹⁴, *Simplest Checklist*¹⁵, Cor Listas¹⁶ e Todoist: Lista de Tarefas¹⁷. Todos estes aplicativos possuem interfaces consideradas amigáveis pelos usuários que facilitam a entrada de dados, a leitura dos dados e também a organização dos dados. O problema é que em nenhum destes aplicativos existe um sistema de recomendação baseado em conteúdo, ou de qualquer outra natureza. Ao se especificar a busca por aplicativos de lista de compras, os melhores aplicativos, segundo a classificação da Play Store são: Lista de Compras – Softlist¹⁸, Lista de Compras¹⁹, Lista de Compras – Meu Carrinho²⁰ e Bring! Lista de Compras²¹.

Ao analisar as descrições dos aplicativos de listas, juntamente com as avaliações de usuários, é possível perceber que neste tipo de aplicativo a preocupação maior é com a interface (Figura 5) do aplicativo e suas funcionalidades básicas. O mais próximo que se atinge da implementação de um sistema de recomendação é a funcionalidade de sugestões de compra de acordo com o horário do dia disponível no aplicativo Lista de Compras – Meu Carrinho. O problema é que as sugestões não são feitas de acordo com os dados inseridos pelo usuário e sim pela hora do dia coletada diretamente do sistema Android.



Figura 5 – Interfaces dos aplicativos gerenciadores de listas mais bem classificados no Google Play Store

¹⁴ <https://play.google.com/store/apps/details?id=com.splendapps.splendo>

¹⁵ <https://play.google.com/store/apps/details?id=jakiganicystems.simplestchecklist>

¹⁶ <https://play.google.com/store/apps/details?id=com.kenzap.checklist>

¹⁷ <https://play.google.com/store/apps/details?id=com.todoist>

¹⁸ <https://play.google.com/store/apps/details?id=br.com.ridsoftware.shoppinglist>

¹⁹ <https://play.google.com/store/apps/details?id=com.tksolution.einkaufszettelmitspracheingabe>

²⁰ <https://play.google.com/store/apps/details?id=com.meucarrinho>

²¹ <https://play.google.com/store/apps/details?id=ch.publisheria.bring>

Considerando o cenário atual dos aplicativos de listas disponíveis para o sistema Android, este trabalho de pesquisa propõe uma alternativa de recomendação de itens baseada em listas de compras cadastradas pelo usuário que, após ser desenvolvida, poderá ser utilizada nos aplicativos gerenciadores de listas já existentes e também em novos aplicativos.

4 Método e Ferramentas

A ideia principal do sistema surgiu do fato de que geralmente quando se elabora uma lista, principalmente uma lista de compras, várias questões estão envolvidas e diversos estudos são realizados sobre esta questão desde 1950 (STEINMAN, 2009). Um dos principais interesses destes estudos é descobrir relações entre produtos que não são percebidas diretamente pelos consumidores como, por exemplo, é possível que sempre que um consumidor compre determinado item de higiene também compre alguma bebida ou comida.

Considerando esta questão, a possibilidade de que um programa que gerencia listas sugerisse itens de acordo com a lista criada pelo usuário se mostrou interessante e motivou o desenvolvimento do sistema deste trabalho. Como as necessidades de produtos são questões muito pessoais a ideia é que não exista predição, e sim uma análise de dados das listas de compras inseridas pelos usuários.

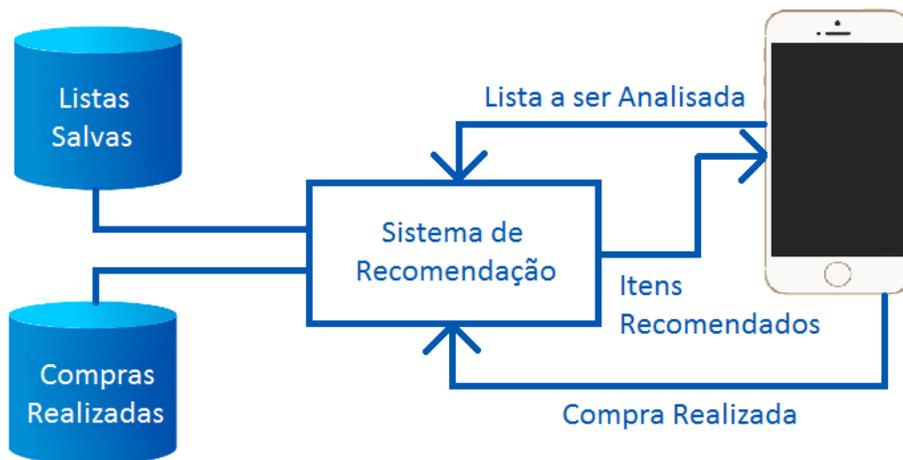


Figura 6 – Diagrama do sistema de recomendação.

A Figura 6 apresenta um diagrama do sistema de recomendação deste trabalho. O sistema basicamente possui acesso a duas bases de dados (que podem ser implementadas de forma unificada): uma base que contém as listas que o usuário preencheu antes de realizar a compra e a outra base que possui as listas de itens que foram comprados. Internamente o sistema configura cada item como um conjunto de atributos, dentre eles: uma lista de quais itens estão relacionados com o item em questão e seu respectivo grau de relacionamento.

Para cada item comprado em determinada lista um procedimento de atualização é feito, onde: para cada um dos outros itens comprados, se eles já estiverem na lista o seu grau de relacionamento é incrementado, senão são adicionados na lista com o grau de relacionamento iniciado em 1. Dessa forma, cria-se um ranking de itens que compõe as recomendações. Para retornar os itens recomendados o sistema ordena os itens relacionados de acordo com seu grau de relacionamento e apresenta a lista em ordem decrescente. Basicamente o sistema de recomendação funciona em dois modos: o modo de recomendação e o modo de atualização. O diagrama de estados de Figura 7 demonstra os possíveis estados do sistema e suas transições.



Figura 7 – Diagrama de estados simplificado do sistema

A ferramenta atual de desenvolvimento móvel para Android indicada pelo desenvolvedor do sistema operacional é o Android Studio. Atualmente em sua versão 2.2.0.12 o software evoluiu rapidamente proporcionando uma interface de desenvolvimento mais amigável, apresentando melhorias e atualizações constantes, conseqüentemente se

tornando a ferramenta principal de desenvolvimento para o sistema operacional Android²². Este trabalho utilizou, portanto, a ferramenta Android Studio em sua versão 2.2.0.12 no sistema operacional Ubuntu na versão 16.04.

A metodologia de desenvolvimento de software adotada foi o desenvolvimento Ágil XP (*eXtreme Programming*) juntamente com a prototipação. A escolha foi feita com base na equipe de desenvolvimento composta apenas por uma pessoa e também no fato de que não se elaborou um documento formal de requisitos para o software sendo que os *feedbacks* dos usuários serão utilizados para o desenvolvimento de novas e aprimoradas versões do sistema.

5 Resultados e Análise

O sistema foi inicialmente desenvolvido e testado em linguagem C para desktop e em seguida em um protótipo para celular. As operações básicas implementadas para o menu principal do aplicativo são a criação de uma nova lista, a inserção de um elemento na base de dados (que no caso é uma nota de compras) e também a recuperação de uma lista salva como apresentado na Figura 7.

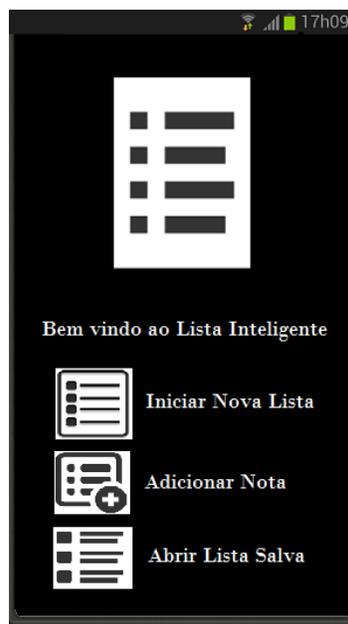


Figura 7 – Menu inicial desenvolvido para o programa Lista Inteligente

²² <http://androidstudiofaqs.com/conceptos/android-studio-vs-eclipse>

O armazenamento das informações é feito em uma matriz de relacionamento (Figura 8) que é percorrida em busca do item inserido na lista para que sejam retornados os itens relacionados a ele como sugestão para o usuário.

| | Pão | Cerveja | Manteiga | Refrigerantes | Detergente | Amaciante | Carne | Arroz | Feijão | Óleo |
|---------------|-----|---------|----------|---------------|------------|-----------|-------|-------|--------|------|
| Pão | 0 | | | | | | | | | |
| Cerveja | 3 | 0 | | | | | | | | |
| Manteiga | 2 | 5 | 0 | | | | | | | |
| Refrigerantes | 4 | 3 | 5 | 0 | | | | | | |
| Detergente | 6 | 2 | 0 | 8 | 0 | | | | | |
| Amaciante | 9 | 7 | 9 | 8 | 2 | 0 | | | | |
| Carne | 7 | 1 | 2 | 2 | 6 | 1 | 0 | | | |
| Arroz | 8 | 2 | 4 | 2 | 4 | 3 | 4 | 0 | | |
| Feijão | 2 | 6 | 6 | 6 | 6 | 5 | 6 | 2 | 0 | |
| Óleo | 6 | 9 | 9 | 9 | 9 | 6 | 9 | 7 | 2 | 0 |

Figura 8 – Exemplo de uma matriz simples armazenada

O armazenamento de dados é feito apenas na parte inferior da diagonal principal matriz, pois a informação armazenada na parte superior é idêntica e não precisa ser armazenada. Quando se insere um elemento na lista de compras que está sendo editada há uma consulta na matriz de armazenamento, que mostra os itens relacionados em ordem decrescente de valor para que o usuário escolha se deseja armazenar ou não o elemento em sua nova lista.



Figura 9 – Exemplo de funcionamento do sistema

Após o preenchimento do banco de dados ao iniciar uma nova lista o cliente se depara com a situação demonstrada pela Figura 9. Neste caso o cliente escolheu um ítem do banco de dados e, baseando-se na matriz de armazenamento, o aplicativo criou uma lista de sugestões.

Caso o usuário queira adicionar um ítem na lista que não esteja cadastrado ele pode escolher a opção adicionar ítem. Quando a matriz não possui informações sobre o grau de relação entre o produto adicionado e outros produtos a lista de sugestões não aparece para aquele ítem da lista. Testes foram realizados com até 100 itens e a lista de sugestões não foi gerada com lentidão e a memória utilizada pelo aplicativo não compromete o sistema.

No caso da inserção de uma nova nota de compras, o sistema recebe cada um dos itens e verifica sua existência ou não na matriz, caso haja algum ítem novo a inserção é feita automaticamente. Após a descrição dos itens o aplicativo automaticamente incrementa a relação de todos os elementos com todos os outros elementos em uma unidade. Após esta inserção o aplicativo retorna ao menu principal.

O sistema utiliza como banco de dados o gerenciamento de arquivos salvos com registros dos dados inseridos e listas realizadas. Basicamente o sistema trabalha com dois arquivos, o primeiro para a matriz de relacionamentos e o segundo para as listas do usuário. Cada registro salvo no arquivo de listas representa uma lista do usuário. O arquivo que armazena a matriz é aberto ao se solicitar uma nova lista no menu principal e o arquivo de listas é aberto quando é solicitada uma lista feita pelo usuário.

Considerações finais

Este projeto de pesquisa apresentou o desenvolvimento de um sistema de recomendação simples baseado na relação direta entre itens de compra. O sistema desenvolvido apresentou resultados satisfatórios para os casos de teste com até 100 itens. O objetivo principal deste trabalho era a verificação da qualidade da solução proposta para o sistema de recomendação.

Foi constatado que, como a relação feita entre produtos é direta, não é necessário criar uma divisão de produtos por categoria já que a associação dos dados da matriz resulta na divisão de categorias de forma automática. Também foi constatado que os produtos inseridos

podem ser de listas de compras dos mais diversos estabelecimentos sem que isso altere a qualidade das respostas da matriz, já que novamente a relação só irá aparecer entre itens cadastrados como relacionamos.

Este fato é interessante também para que sejam relacionadas compras em estabelecimentos diferentes. Por exemplo, uma compra num estabelecimento de venda de roupas pode ser relacionada a uma compra de supermercado e o usuário pode descobrir que toda vez que compra uma blusa nova também compra refrigerante no supermercado.

O programa agora deve ser testado com um número maior de itens e também deve ser testado em diversos celulares e tablets reais para uma avaliação de desempenho. Será necessário também implementar um sistema de “*garbage collection*” que limpe a matriz de produtos relacionados de tempos em tempos para que a memória consumida no programa não seja um problema para os dispositivos móveis. Após realizados estes testes o programa pode ser então liberado em sua primeira versão nas lojas de aplicativos móveis.

Referências

- BEEL, J.; et al. Research paper recommender system evaluation: a quantitative literature survey. **Proceedings of ACM RepSys**. 2013.
- BELKIN, N. J.; CROFT, W. B. Information filtering and information retrieval: Two sides of the same coin? **Communications of the ACM**, v. 35, n. 12, p. 29-38, 1992.
- BOBADILLA, J. et al. Recommender Systems Survey. **Knowledge-Based Systems**, v. 46, p. 109-132, 2013.
- HAN, J. **Data Mining: concepts and techniques**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.
- LUA, L. et al. Recommender Systems. **Physics Reports**, n. 519, p. 1-49, 2012.
- MANBER, U.; PATEL, A.; ROBISON, J. Experience with personalization on Yahoo!. **Communications of the ACM**, v. 43, n. 8, 2000.
- RESNICK, P.; VARIAN, H. R. Recommender Systems. **Communications of ACM**, v. 40, n. 3, p. 56-58, 1997.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to Recommender Systems Handbook. In: **Recommender Systems Handbook**. Springer US, 2011. p. 842.
- STEINMAN, R. B. Projective Techniques in Consumer Research. **International Bulletin of Business Administration** - Eurojournals Inc, v. 5, 2009.