

**Programa Institucional de Bolsas
de Iniciação em Pesquisa Científica e Tecnológica
PIBIC/CNPq - PIBITI/CNPq - PIBIC_Af/CNPq - BIPI/UFSC
2016/2017**

**Arquitetura Baseada em Serviços como Suporte para
Sistemas de Recomendação**

Relatório Final

**Bolsista:
THIAGO RAULINO DAL PONT**

Orientador: ALEXANDRE LEOPOLDO GONÇALVES

Agosto de 2017

RESUMO

O avanço da Web e das tecnologias sem fio tem promovido novas formas de interação por parte de usuários resultando no aumento do volume de informações, bem como em uma maior dificuldade na tomada de decisão. Contudo, este cenário gera oportunidades na construção de sistemas capazes de auxiliar usuários em suas escolhas. Este projeto propõe uma arquitetura de serviços com foco em Sistemas de Recomendação e Recuperação de Informação. No contexto do trabalho implementou-se uma arquitetura de serviços como forma de melhorar a interatividade e a localização de conteúdos estruturados e não estruturados que possam ser de interesse de determinado usuário. Para a avaliação da arquitetura foram desenvolvidas duas aplicações, sendo uma para a consulta e sugestão de documentos e outra para a comunicação com sensores sem fio e a sugestão de produtos. Ambas as aplicações interagem com a camada de serviços e permitem a obtenção de recomendações. Neste sentido, pode-se concluir que a arquitetura desenvolvida se mostrou adequada, tanto na captura das interações resultantes da utilização das aplicações quanto na recomendação de conteúdos de potencial interesse para determinado usuário.

Palavras-chave: Coleta de Informações, Recuperação de Informação, Sistemas de Recomendação, Sensores sem Fio.

SUMÁRIO

1. INTRODUÇÃO.....	5
1.1 PROBLEMÁTICA.....	6
1.2 OBJETIVOS.....	7
1.2.1 Objetivo Geral	7
1.2.2 Objetivos Específicos	7
1.3 MATERIAIS E MÉTODOS	7
1.4 DETALHAMENTO DAS FASES.....	8
2. FUNDAMENTAÇÃO TEÓRICA	10
2.1 SISTEMAS DE RECOMENDAÇÃO.....	10
2.2 SISTEMAS DE RECOMENDAÇÃO.....	11
3. ARQUITETURA, DETALHAMENTO E EXPERIMENTOS.....	14
3.1 APRESENTAÇÃO DA ARQUITETURA	14
3.2 RECOMENDAÇÃO DE CONTEÚDO NÃO ESTRUTURADO....	16
3.3 RECOMENDAÇÃO BASEADA EM SENSORES SEM FIO.....	18
Sensores iBeacon.....	18
Tags NFC	19
Aplicações desenvolvidas	19
Biblioteca AltBeacon	29
Servidor web	31
Base de dados do aplicativo	34
Cadastro de Imagens	Erro! Indicador não definido.
Exibição das recomendações.....	Erro! Indicador não definido.
4. ATIVIDADES DO BOLSISTA.....	37
5. AVALIAÇÃO DO BOLSISTA EM RELAÇÃO AO PIBIC	40

6. CONSIDERAÇÕES FINAIS	41
7. REFERÊNCIAS	43

1. INTRODUÇÃO

A Web 2.0 permitiu que os usuários agregassem novos conteúdos e novos sites com facilidade, possibilitando a conexão destes conteúdos com outras páginas (O'REILLY, 2007). A Web tornou-se bidirecional, ou seja, uma Web participativa com leitura e escrita de conteúdo, potencializando as formas de comunicação onde o usuário deixa de ser apenas o receptor e passa a interagir contribuindo com conteúdo. Desta forma, o conteúdo migra de um padrão estático para um padrão dinâmico (AGHAEI; NEMATBAKHS; FARSANI, 2012).

Neste cenário a Web 2.0 trouxe consigo um novo paradigma e também repercussões sociais, difundido a produção e circulação de informações, vista como uma nova plataforma que viabiliza funções online, aperfeiçoando a usabilidade e aprimorando o conceito de "arquitetura de participação". Essa arquitetura visa oferecer serviços, tais como: publicação em espaço de debate, gestão coletiva de trabalho, negociação coletiva e interação social (PRIMO, 2007).

O crescimento exponencial de informação gerada trouxe como consequência uma grande diversidade de conteúdo a disposição do usuário, afetando principalmente a sua capacidade de escolha e, portanto, de tomada de decisão tornando necessário recursos que o ajude a realizar suas escolhas. Neste contexto, as áreas Recuperação de Informação (RI) e Sistemas de Recomendação (SR) são essenciais para prover sistemas capazes de lidar adequadamente com o volume de informações e com a recomendação de conteúdos de maior interesse por parte dos usuários (FERNEDA, 2003).

Segundo Manning (2009), a Recuperação de Informação consiste na localização de materiais de natureza não estruturada que satisfazem determinada necessidade por informação a partir de uma coleção de documentos. A Recuperação da Informação abrange tecnologias de consulta e indexação e está fundamentada na análise e disponibilização automática de conteúdo, normalmente documentos textuais (FOLTZ; DUMAIS, 1992; HERLOCKER, 2000).

Para que as buscas realizadas pelos usuários sejam mais eficientes são utilizados alguns modelos que permitem determinar a relevância dos documentos no conjunto de retorno. Além das estratégias de busca é necessário um direcionamento para que o usuário saiba quais dos resultados listados satisfazem a sua necessidade, para tal, a

utilização de Sistemas de Recomendação pode proporcionar informações mais apuradas a partir de determinada consulta do usuário (CAZELLA; NUNES; REATEGUI, 2010).

Os Sistemas de Recomendação são ferramentas de software e técnicas que proporcionam sugestões de itens para serem utilizados pelo usuário (RICCI et al., 2011) em que o conceito de item pode ser generalizado para vários domínios. Têm por objetivo a redução da sobrecarga de conteúdo, através da seleção de informações baseada, por exemplo, em prioridades e interesses prévios do usuário (FIGUEIRA FILHO; GEUS; ALBUQUERQUE, 2008).

1.1 PROBLEMÁTICA

A Web 2.0 possibilitou que usuários produzissem novos conteúdos nas mais diversas aplicações/plataformas passando de expectadores para produtores de conteúdo. Soma-se a isto a interação cada vez maior com sensores sem fio. De modo geral, permite que usuários assumam um papel proativo na criação e disponibilização de informação, contribuindo de maneira decisiva no aumento exponencial de dados. Este aumento gera muitos desafios tecnológicos, mas também, afeta a capacidade de escolha, ou seja, a capacidade de decisão dos usuários.

Neste contexto, as áreas de Recuperação de Informação e Sistemas de Recomendação podem prover soluções para lidarem de maneira adequada com grandes volumes de informação e a partir disto recomendar conteúdos de interesse dos usuários. Segundo Ferneda (2003), a Recuperação de Informação objetiva localizar/recuperar conteúdos que em uma coleção de documentos satisfaçam uma consulta. Pode ainda combinar estratégias para que dados estruturados sejam integrados com dados não estruturados, textos. Por outro lado, Sistemas de Recomendação visam prover conteúdos analisando as preferências de um ou mais usuários (FIGUEIRA FILHO; GEUS; ALBUQUERQUE, 2008).

Considerando estes desafios, torna-se relevante o desenvolvimento de uma arquitetura que facilite tanto o processo de localização de informação quando o processo de sugestão de conteúdo baseado na experiência de outros usuários ou perfil de determinado usuário.

A partir do exposto acima, este projeto possui como pergunta de pesquisa “Como elaborar uma arquitetura de serviços que auxilie na recuperação, análise e escolha de conteúdo por parte de determinado usuário considerando seu interesse?”.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral da pesquisa constitui-se na proposição de uma arquitetura baseada em serviços como suporte para a realização de recomendações de conteúdo estruturado ou não.

1.2.2 Objetivos Específicos

Com a finalidade de atingir o objetivo geral, têm-se os seguintes objetivos específicos:

- Realizar um levantamento bibliográfico das áreas de suporte a este projeto, a saber, Recuperação de Informação e Sistemas de Recomendação;
- Propor uma arquitetura de software baseada em serviços que promova suporte a recomendação de conteúdo estruturado e não estruturado;
- Desenvolver um protótipo que possibilite avaliar o conjunto de serviços de geração e provimento de recomendações;
- Realizar uma discussão dos resultados obtidos através da utilização do protótipo desenvolvido.

1.3 MATERIAIS E MÉTODOS

O trabalho aqui apresentado, sob o ponto de vista de sua natureza é caracterizado por uma pesquisa aplicada uma vez que **objetiva gerar conhecimentos para aplicação prática e que sejam dirigidos à solução de problemas específicos.**

Para atingir os objetivos deste projeto de pesquisa, os procedimentos metodológicos a serem executados são:

- Revisão da literatura científica relevante para o desenvolvimento deste trabalho, sendo: (a) indexação e recuperação de informação textual; (b) integração de informação estrutura e não estruturada; (c) sistemas de recomendação;
- Estudo dos materiais, métodos e ferramentas computacionais utilizados neste trabalho, entre eles: (a) sistemas de recuperação de informação; (b)

frameworks e bibliotecas que possibilitem o desenvolvimento de serviços voltados a recomendação; e (c) sistemas de recomendação;

- Prototipação da arquitetura de serviços que promova suporte a recomendação de conteúdo estruturado e não estruturado;
- Elaboração de um cenário que possibilite avaliar a arquitetura de serviços proposta;
- Análise dos resultados obtidos através da utilização da arquitetura de serviços proposta.

1.4 DETALHAMENTO DAS FASES

Tendo os procedimentos já definidos serão detalhadas como cada uma das fases executadas considerando os materiais e métodos utilizados.

Para fase 1 do projeto foram realizadas pesquisas bibliográficas com o intuito de entender melhor sobre a manipulação e tratamento de informação estruturada e não estruturada. Neste sentido, entendimento das áreas de Banco de Dados e Recuperação de Informação se fez necessário. Para o processo de sugestão foi estudada a área de Sistemas de Recomendação. Visando permitir a aquisição de dados que permitissem uma contextualização mais adequada em determinados domínios onde a área de recomendação se aplica, estudou-se a área de sensores sem fio. Por fim, para viabilizar o objetivo geral do trabalho fez necessário o estudo do conceito de arquitetura de serviços.

Para a fase 2 foram identificados os softwares e bibliotecas necessários para a realização do projeto. Para o desenvolvimento da arquitetura de serviços foi utilizado o ambiente de desenvolvimento Eclipse®, o qual possui elementos adequados para a criação de servidores de aplicação baseados em serviços. Além disso, utilizou-se o ambiente Android Studio®, pois este possibilita o desenvolvimento de aplicações móveis, bem como, a emulação de dispositivos móveis para teste. Visto que a arquitetura prevê a utilização de sensores sem fio para permitir a coleta das interações realizadas por usuários em locais físicos foram utilizados sensores *Bluetooth Low Energy*® (BLE) 4.0. Para manipular o sensor e permitir a coleta de dados foi utilizada a biblioteca AltBeacon®. Ambos, sensor e biblioteca, serão discutidos com maiores detalhes no Capítulo 3. Para a manipulação das informações textuais, ou seja, o conteúdo não estruturado, utilizou-se a biblioteca Apache Lucene®.

Na fase 3 do projeto foi realizado o desenvolvimento a arquitetura composta por:

- a) uma camada de serviços capaz de registrar as interações de usuários com sistemas web tradicionais ou aplicações móveis que tenham coletado dados a partir de sensores sem fio; b) uma aplicação Web estilo mecanismo de busca para a recuperação e análise de recomendações de documentos textuais; c) uma aplicação móvel utilizada para a coleta de dados de sensores sem fio visando determinado contextos mais específicos quando da interação de determinado usuário com objetos físicos (por exemplo, o interesse por parte do usuário em determinado produto ao aproximar o *Smartphone* deste em uma loja); d) desenvolvimento da camada de serviços de recomendação que analisa as interações e perfis correlatos ao perfil de determinado usuário para realizar sugestões de itens (produtos, artigos, livros, roteiros, etc.); e e) um módulo *backend* que analisa as interações e carrega estruturas agregadas que fornecerão suporte para os serviços de recomendação.

A fase 4 teve como objetivo avaliar os protótipos e a partir desta avaliação promover melhorias na arquitetura desenvolvida. Para simular determinado cenário foram desenvolvidas três aplicações capazes de demonstrar as funcionalidades desenvolvidas. A primeira aplicação constitui-se em um gerador de dados fictícios que simula a interação de usuários com a arquitetura. Os dados gerados são então armazenados no modelo proposto (maiores detalhes no Capítulo 3). A segunda é composta por um mecanismo de busca que possibilita a recuperação de documentos e, posteriormente, a interação com os documentos visando obter sugestões de documentos relacionados. Em seguida ocorreu o desenvolvimento de um aplicativo para dispositivos móveis que captura a interação do usuário com determinado sensor sem fio e envia este dado para a arquitetura, permitindo que o mesmo seja armazenado. A partir dos dados de interação existe um módulo/processo que analisa todos os dados armazenados e, após os cálculos, realiza sugestões que podem ser consultadas pelas aplicações via a camada de serviços.

Por fim, realizou-se a última fase do projeto (fase 5) consistindo na apresentação de sugestões a partir da camada de serviços tanto de artigos científicos (informação não estruturada armazenada na base de dados), quanto de produtos a partir da análise de determinado perfil de usuário. Realizou-se também uma análise das recomendações providas pela arquitetura desenvolvida neste projeto.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, é apresentada a fundamentação teórica referente às principais áreas de suporte para a realização deste projeto.

2.1 SISTEMAS DE RECOMENDAÇÃO

A Web vem promovendo um crescimento exponencial da informação o que promove uma grande diversidade de conteúdo a disposição, tornando-se necessária a presença de recursos que auxiliem usuários menos experientes a realizar suas escolhas.

De modo geral, o usuário quando recebe uma recomendação sobre um item que está buscando, sente-se seguro para realizar sua escolha com eficácia (CAZELLA; NUNES; REATEGUI, 2010). Neste sentido, os Sistemas de Recomendação surgiram para sanar essa necessidade por meio de softwares e técnicas que proporcionam sugestões de itens para serem utilizados pelo usuário (RICCI et al., 2011).

Os Sistemas de Recomendação têm por objetivo reduzir a sobrecarga de conteúdo, por meio da seleção de informações baseadas nas prioridades do usuário (FIGUEIRA FILHO; GEUS; ALBUQUERQUE, 2008). Atualmente os SRs têm sido classificados em algumas categorias, sendo as mais mencionadas e estudadas a abordagem Baseada em Conteúdo, abordagem de Filtragem Colaborativa e abordagem Híbrida.

A filtragem realizada através de análises dos conteúdos dos itens é denominada filtragem baseada em conteúdo (Content-based Filtering) (LOPES, 2007). Esta abordagem baseia-se na disponibilidade das características de itens que podem ser criados manualmente ou extraídos automaticamente, e um perfil que atribui importância a essas características. Os perfis de usuários também podem ser derivados automaticamente e serem “aprendidos” por meio da interação do usuário com o sistema, ou podem ser obtidos por meio de informações que foram providas previamente pelo próprio usuário (JANNACH et al, 2011). A combinação destes perfis permite a sugestão de itens semelhantes.

Este tipo de filtragem busca recomendar itens que possuam descrições similares a itens de interesse, avaliados ou não pelo usuário de maneira positiva ou negativa. Utiliza-se o conceito de que se possuem descrições similares serão avaliados de maneira

semelhante (BOBADILLA et al., 2013). Segundo Garcia e Frozza (2013), a filtragem baseada em conteúdo possui maior aplicabilidade em recomendações textuais, devido à facilidade que existe em verificar o quão similar é o interesse do usuário com o texto, pois podem ser utilizadas palavras-chave para verificar esta similaridade.

Outra abordagem frequentemente estudada e referenciada é a colaborativa. Segundo Cazella, Nunes e Reategui (2010), a abordagem Colaborativa ou Filtragem Colaborativa (FC) é a indicação de itens a usuários baseada na experiência de outros usuários que possuem interesses similares. A indicação ocorre através de avaliações que estes realizaram para os itens.

Uma estratégia muito utilizada para calcular a similaridade entre itens é o algoritmo K-Vizinhos mais Próximos (K-Nearest Neighbor) que pode ser dividido em duas abordagens: Recomendação dos Vizinhos mais Próximos baseada no Usuário (User-based Nearest Neighbor Recommendation) e Recomendação dos Vizinhos mais Próximos baseada no Item (Item-based Nearest Neighbor Recommendation) (SCHAFER et al., 2007).

A combinação, chamada de Sistemas de Recomendação Híbridos, leva em conta os pontos fortes dessas abordagens com o objetivo de tornar o modelo híbrido mais eficiente (JANNACH et al., 2011).

2.2 SISTEMAS DE RECOMENDAÇÃO

O termo Recuperação de Informação (Information Retrieval) foi definido por Calvin Mooers em 1951 (MOOERS, 1951). Possui como tarefa essencial a localização de conteúdo, geralmente textos, em resposta a uma demanda (consulta de interesse). Além disso, abrange uma fase de composição da base de documentos para consulta (indexação) e a disponibilização de sistemas capazes de possibilitar a interação do usuário.

A indexação é um método que analisa o conteúdo e sintetiza a informação relevante do documento, criando um vocábulo mediador entre o usuário e o documento (VIEIRA, 1988). Durante o pré-processamento da indexação podem ocorrer algumas etapas, sendo, a extração de termos (Tokenization), eliminação de termos por meio de uma lista de termos (Stop List), extração de radicais (Stemming) e utilização de tesauro (Thesaurus).

Segundo Manning (2009) a extração de termos (Tokenization) é a tarefa de partir em tokens o documento textual, ignorando alguns caracteres, como a pontuação e o espaçamento. A etapa de listas de termos (Stop List) identifica um conjunto de palavras irrelevantes (Stop Words), na qual não alteram a essência do texto, que se repetem com frequência em um corpus, como artigos, conjunções, pronomes e preposições (SALTON; MCGILL, 1983; VAN RIJSBERGEN, 1975). Segundo Srinivasan (1992), tesouros (Thesaurus) são estruturas importantes para os Sistemas de Recuperação de Informação (SRI), pois fornecem um vocabulário preciso e controlado utilizado para coordenar a indexação e a recuperação dos documentos.

Após a etapa de pré-processamento criam-se os índices, onde todos os termos dos textos são inseridos em uma lista (índice invertido) com referências aos documentos onde estes estão presentes, além de armazenarem a posição onde se encontram no documento.

Uma vez que os índices estejam disponíveis torna-se essencial a disponibilização de sistemas que permitam que usuários realizem buscas. O retorno dessas buscas deve considerar algum critério de ordenação determinado por um modelo. Entre os mais conhecidos e implementados está o modelo vetorial, também conhecido como espaço vetorial. Caracteriza-se como um modelo estatístico, onde o documento é representado através de um vetor (lista de termos) no espaço n-dimensional, onde cada entrada do vetor representa um termo com seu respectivo peso, ou relevância em um determinado documento (SOUZA, 2006).

Para realizar o cálculo do grau de similaridade de determinado vetor de entrada (documento ou consulta) em relação a todos os documentos de um corpus é aplicada a medida do cosseno θ (representado na fórmula abaixo). O resultado esperado será um valor compreendido entre “0” e “1”, sendo que quanto mais próximo de “1” maior é sua relevância (FERNEDA, 2003):

$$\text{sim}(t,q) = \frac{\sum_{i=1}^n (t_i \times q_i)}{\sqrt{\sum_{i=1}^n (t_i)^2} \times \sqrt{\sum_{i=1}^n (q_i)^2}}$$

O retorno de uma consulta utilizando o modelo é uma lista ordenada de documentos obtida através do grau de similaridade, ou seja, a ordenação é realizada considerando a relevância entre os documentos e os termos (argumentos) de busca. Além disso, segundo Souza (2006), o modelo vetorial possui um bom desempenho e

apresenta uma implementação simples.

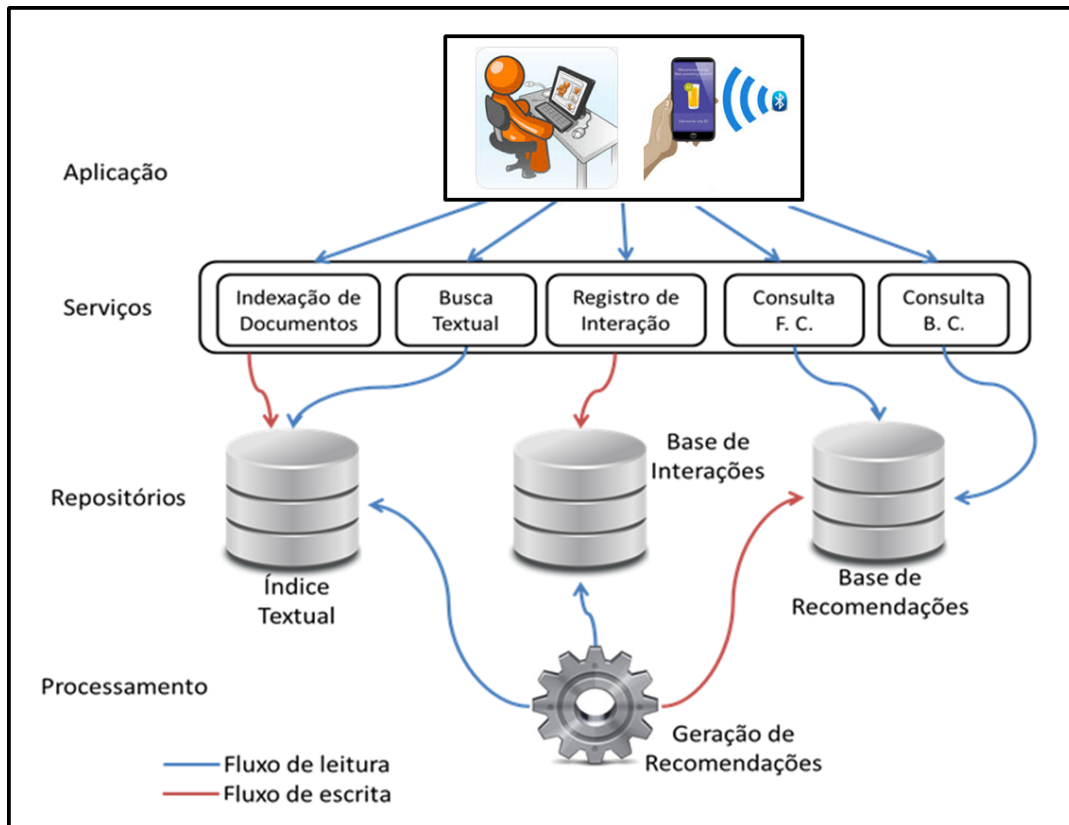
3. ARQUITETURA, DETALHAMENTO E EXPERIMENTOS

As seções a seguir detalham a arquitetura desenvolvida e apresentam os experimentos realizados.

3.1 APRESENTAÇÃO DA ARQUITETURA

Esta seção apresenta a arquitetura proposta detalhando a interconexão entre as camadas que a compõem. A Figura 1 apresenta as quatro camadas da arquitetura proposta Aplicação, Serviços, Repositórios e Processamento.

Figura 1: Detalhamento da arquitetura proposta



Fonte: Autores

Na Camada de Aplicação constam uma aplicação Web (mecanismo de busca) e uma aplicação móvel que se comunica com sensores sem fio. Estas aplicações têm como objetivo coletar toda e qualquer interação do usuário (através da aplicação Web ou da aplicação móvel), e enviar os dados para serem armazenados em bancos de dados. Neste sentido, a camada de aplicação permite capturar as interações dos usuários e

enviar tais interações para a próxima camada.

A camada seguinte é composta por um conjunto de serviços que promovem todas as funcionalidades na arquitetura proposta. A comunicação com a camada de serviço envolvendo requisições e respostas são realizadas através do padrão JSON (*JavaScript Object Notation*). O primeiro serviço possibilita a indexação de documentos utilizando, ou uma interface de administração em que um arquivo pode ser enviado ou através de um processo em *background* que avalia novos documentos disponíveis que são coletados e armazenados em um sistema de *cache*. Este processo é incremental e permite a formação da base de índices que neste trabalho é suportada pela biblioteca *Lucene*TM.

Uma vez que o índice esteja disponível o usuário pode realizar consultas indicando um conteúdo (palavras-chave) de interesse. O serviço de consulta então realiza a pesquisa na base de índices retornando um conjunto de documentos ordenados segundo sua similaridade em relação ao conteúdo da consulta. A recuperação desses documentos também é provida pela biblioteca *Lucene*TM que disponibiliza um modelo vetorial.

Após o retorno da consulta o usuário pode interagir com o resultado, ou seja, pode selecionar ou avaliar determinado documento. No caso da interação por meio da avaliação esta ocorre atribuindo-se um valor que varia de 1 a 5, sendo 1- muito ruim, 2- ruim, 3-regular, 4-bom e 5-excelente. Durante a etapa de interação do usuário com o sistema todas as ações deste são armazenadas por meio do serviço de Registro de Interações na base de interações. Com o objetivo de possibilitar a geração de recomendações, as interações são agregadas por sessão, sendo esta determinada por uma consulta em particular, ou seja, cada nova consulta produz uma sessão que irá registrar toda a atividade.

Este serviço também permite o registro de qualquer interação com sistemas que não envolvam a avaliação de documentos, como discutido acima. Por exemplo, em um sistema de *e-Commerce* as interações poderiam ser a compra, o interesse (o usuário analisa o produto, mas não compra), ou mesmo, a avaliação do item em determinado site. Estas informações são registros de interação e são enviadas para serem armazenados na base de integrações. Possibilita ainda armazenar as interações entre uma aplicação móvel utilizada por um usuário e sensores sem fim. Sensores sem fio fornecem um meio de contextualizar mais claramente o interesse de usuários em determinado produto quando este está disponível em um local físico. Por exemplo, o

usuário se interessa por um produto qualquer em um supermercado e, para obter maiores informações aproxima o dispositivo móvel. Esta ação irá capturar a identificação do sensor e enviar para a camada de serviço. De posse desta informação e analisando o que já está processando na base de recomendações, o usuário poderá receber informações contextualizadas indicando outros produtos que possam ser de seu interesse.

De maneira complementar aos serviços existe um processo responsável pela geração das recomendações. Este processo, responsável por promover suporte aos cálculos de recomendação, frequentemente inspeciona as bases de interação e o índice textual. O primeiro tipo de recomendação é a colaborativa e é calculada através da análise do perfil de usuários, ou seja, produtos podem ser sugeridos para determinado usuário considerando a similaridade do seu perfil com outros usuários de perfil semelhante. O outro tipo de recomendação é a baseada em conteúdo e se utiliza da base de índices para a localização/recuperação e sugestão de documentos.

Por fim, os serviços de consulta às recomendações permitem sugerir, para cada item, os itens mais relacionados. O serviço de consulta por filtragem colaborativa (F. C.) provê itens associados considerando a interação dos usuários. Por outro lado, o serviço de consulta baseado em conteúdo (B. C.) sugere documentos relevantes levando em conta a similaridade de determinado documento. Em ambos os casos as informações são obtidas na base de recomendações geradas pelo processo descrito anteriormente.

3.2 RECOMENDAÇÃO DE CONTEÚDO NÃO ESTRUTURADO

Como apresentado na descrição da arquitetura uma das características importantes refere-se a possibilidade de realização de recomendações a partir de conteúdo não estruturado. Neste sentido, visando constituir uma base de dados para o teste da camada de serviço realizou-se uma coleta de dados a partir da base de artigos ScienceDirect®. De modo que se pudesse extrair o texto completo, o acesso foi realizado em modo restrito, ou seja, utilizando a rede da Universidade Federal de Santa Catarina. Para o estudo foram coletados 400 documentos em texto completo em formato PDF.

A partir dos documentos foi efetuada a extração do texto e este conteúdo foi então utilizado no processo de indexação. Visando prover informações agregadas que pudessem representar critérios estruturados de filtragem foi elaborado para cada documento um arquivo de meta-informação composto por título, autores, organização

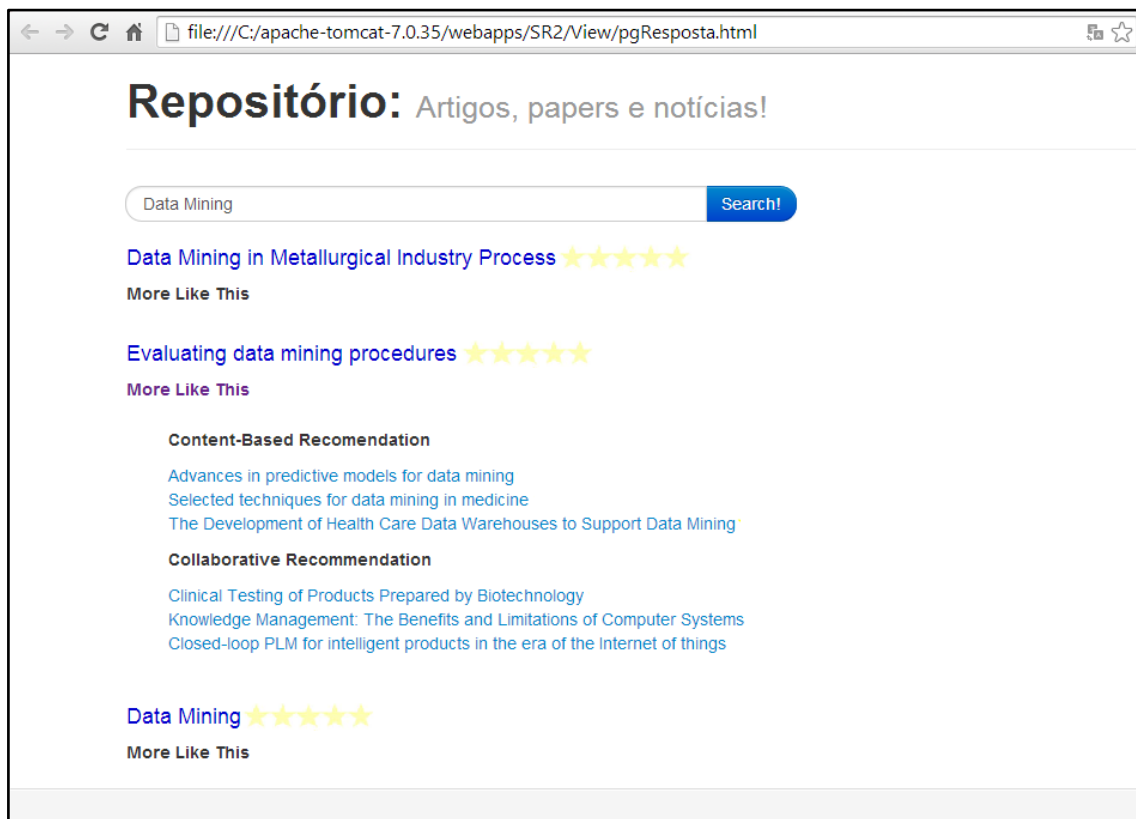
dos autores, palavras-chave e ano de publicação. Este conteúdo, juntamente com o texto do documento, foi integrado à base de índices visando permitir a recuperação de documentos por meio do serviço identificado como “Busca Textual” e a recomendação de novos documentos a partir de determinado documento de interesse através do serviço “Consulta B. C. – Consulta Baseada em Conteúdo”.

Os documentos coletados representam o conjunto de itens que podem ser consultados ou analisados durante a interação de determinado usuário com uma aplicação. Visando o estabelecimento de testes mais robustos considerando as interações de usuários elaborou-se uma aplicação que popula a base de dados. Foram geradas milhares de sessões (simulando uma sessão do usuário utilizando uma aplicação Web, por exemplo), em que cada sessão possuía um número mínimo e máximo de interações distintas, ou seja, sem a duplicação de itens. As interações foram geradas considerando a tipificação de acesso, leitura de determinado documento ou a avaliação de um documento. Como resultado do processo de carga mais de 200.000 interações foram geradas. A partir da base de interações realizou-se a aplicação de algoritmos visando calcular correlações e similaridades entre os itens de modo que, levando em conta determinado item de interesse (produto ou documento), o usuário pudesse receber recomendações que estivessem aderentes aos seus interesses passados.

Com o intuito de demonstrar a utilização do serviço de recomendação para informação não estruturada foi elaborada uma aplicação (protótipo) similar a um mecanismo de busca. Através desta aplicação o usuário pode localizar documentos utilizando-se de palavras-chave e receber um conjunto de documentos. Uma vez apresentados os documentos e, em havendo interesse por parte do usuário, este pode selecionar uma opção que irá localizar na base de recomendações outros documentos semelhantes. A aplicação Web (estilo *search engine*) (Figura 2) foi desenvolvida por André e Laurindo (2014) e utilizada neste trabalho para testar a arquitetura e a camada de serviço no que se refere a recomendação de conteúdo não estruturado.

Conforme se observa na Figura 2, uma consulta foi realizada utilizando o termo “*Data Mining*” tendo como resposta documentos que mencionam em seu conteúdo a palavra *Data* e *Mining* na sequência. A partir da apresentação do resultado o usuário pode interagir com cada documento de três formas, sendo, o acesso/abertura do texto completo do documento, a avaliação do documento através da indicação da relevância do mesmo (informando a quantidade de estrelas que denota um *rating*), e a obtenção das recomendações por meio da opção “*More Like This*”.

Figura 2: Aplicação para a recuperação e recomendação de documentos



Fonte: André e Laurindo (2014)

Ao selecionar a opção “*More Like This*” de um documento em particular o usuário recebe a sugestão de documentos relacionados. As recomendações podem ser de dois tipos, Recomendações Baseadas em Conteúdo e Recomendações Colaborativas. No primeiro caso a sugestão ocorre com base na avaliação da similaridade entre o conteúdo do documento desejado e o conteúdo de outros documentos armazenados na base de dados. Sendo assim, os 3 (três) documentos mais similares são apresentados. Para o segundo tipo, são analisados os interesses de usuários com perfis similares ao usuário de determinada sessão, realizando sugestões que este ainda não tenha lido ou avaliado, ou seja, utilizando o conceito de Filtragem Colaborativa.

3.3 RECOMENDAÇÃO BASEADA EM SENSORES SEM FIO

O desenvolvimento da arquitetura de recomendação baseada em sensores fundamentou-se no uso de sensores sem fio do tipo iBeacon®.

Sensores iBeacon

O iBeacon pode ser visto como um serviço de notificação desenvolvido pela

Apple®. Permitem assim, o desenvolvimento de aplicações móveis capazes de criar um ponto de interação entre o usuário e o sensor. São baseados na tecnologia *Bluetooth Low Energy* (BLE) sendo capazes de prover informações de contexto, como por exemplo, ao aproximar-se de um desses dispositivos o mesmo é ativado permitindo obter a localização, data e hora, proximidade, entre outras informações que sejam relevantes.

Para que seja possível a determinação do contexto é necessária a obtenção de informações específicas do *beacon* em questão, sendo elas o MAC (*Media Access Control*), que funciona como um identificador, o *Major* e *Minor*, informações que agregam mais especificidade sobre a identificação do dispositivo, além da potência de sinal ou RSSI (*Received Signal Strength Indicator*), que auxilia na localização do dispositivo móvel em relação ao sensor.

Em conjunto, MAC, *Major*, *Minor* e RSSI permitem conhecer o dispositivo em si, o grupo de dispositivos ao qual ele pertence e a distância ao qual o sensor foi ativado.

Tags NFC

<Descrever as características da TAG, como é feita a leitura, os dados que contém, como funciona o registro de informações.>

Aplicações desenvolvidas

NFC R/W

Uma aplicação extra foi implementada anteriormente à final. Nela, buscou-se entender como ocorre a interação entre smartphone e etiquetas ou tags NFC, além do funcionamento das classes nativas que o sistema Android oferece para interação com NFC

Além de proporcionar a aprendizagem, a aplicação buscou auxiliar na implementação da aplicação final, pois, a partir da possibilidade de leitura e escrita nas etiquetas, seria possível adotar um padrão específico para o conteúdo dessas e, desse modo, otimizar a aplicação.

Implementação

O Android oferece classes que facilitam a implementação de aplicações que interajam com NFC, entre elas está a NFCAdapter, que representa o adaptador NFC. A partir dessa classe é possível ter conhecimento da compatibilidade do dispositivo utilizado com a tecnologia NFC, configurar o modo de operação, verificar se o NFC está habilitado, entre outros.

Antes de qualquer implementação da aplicação é necessário informar ao Android que será usado o NFC e, para tanto, deve ser indagado a permissão do usuário para a utilização deste. Isso é feito no manifesto do Android a partir da seguinte linha de código.

Figura 3: Comando de requisição do uso de NFC no Android

```
<!-- Requisição para uso do NFC -->  
<uses-permission android:name="android.permission.NFC" />
```

Na especificação do Android, sempre que uma tag NFC ser aproximada do smartphone, será lançada uma Intent, independentemente da aplicação em execução no momento. Uma Intent pode ser descrita como uma ação a ser executada. Nesse caso, a ação é a descoberta de etiquetas NFC, mas, ao ser lançada uma Intent, a ação já foi executada. A partir da intent é possível obter informações sobre a tag detectada e utilizá-las na aplicação desejada.

Para que uma aplicação receba os dados das etiquetas rastreadas, é necessário que ela deve filtrar alguma Intent para NFC, às quais se referem ao formato do conteúdo da tag identificada. Existem três possíveis Intents filtráveis:

- ACTION_NDEF_DISCOVERED
- ACTION_TECH_DISCOVERED
- ACTION_TAG_DISCOVERED

Para essa aplicação fez-se uso da ACTION_NDEF_DISCOVERED. A configuração da filtragem dessa e das demais Intents é feita no arquivo manifesto da aplicação Android com base nas seguintes linhas de código:

```
<intent-filter>  
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />  
  <category android:name="android.intent.category.DEFAULT" />  
  <data android:mimeType="text/plain" />  
</intent-filter>
```

Além do tipo de Intent, é configurada o tipo de dados Mime, ou seja, o formato do conteúdo armazenado, nesse caso em texto.

Para definir o que será feito quando a Intent de NFC for lançada e filtrada, é necessária implementação do método `onNewIntent` na mesma classe da Activity em que se pretende interagir com NFC.

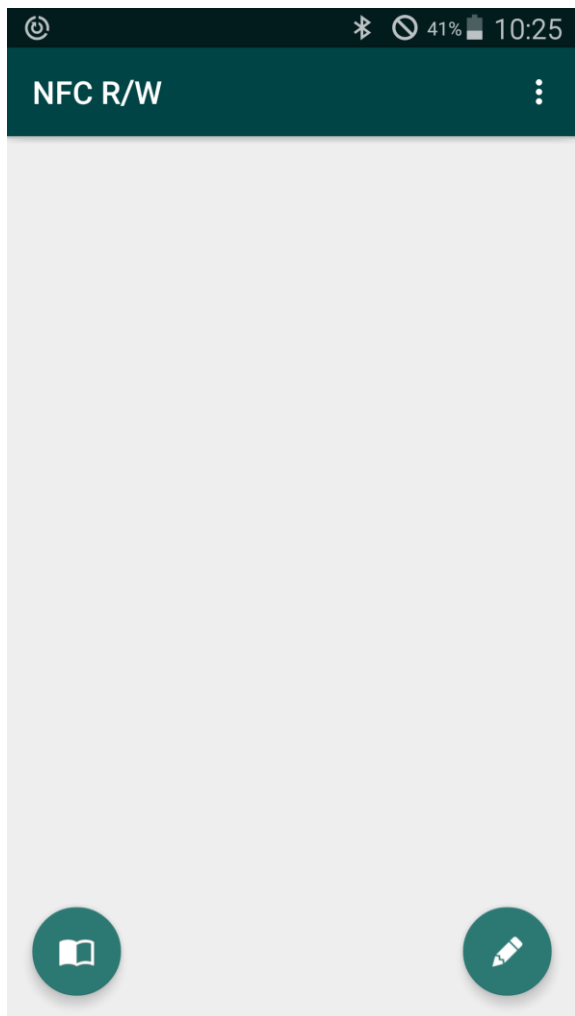
```
@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);

    // Implementar ações aqui
}
```

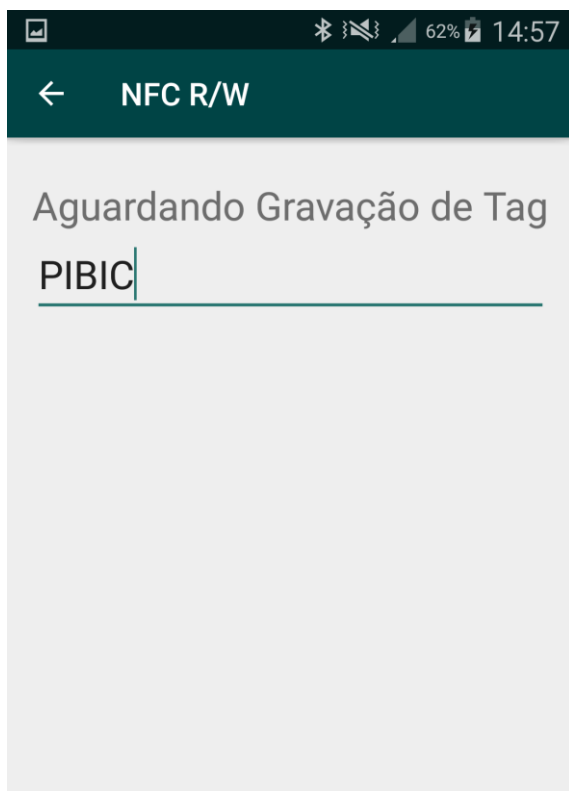
A partir da instância `intent` será possível, então obter informações sobre a etiqueta descoberta.

Funcionamento

Ao abrir o app, depara-se com a seguinte janela.

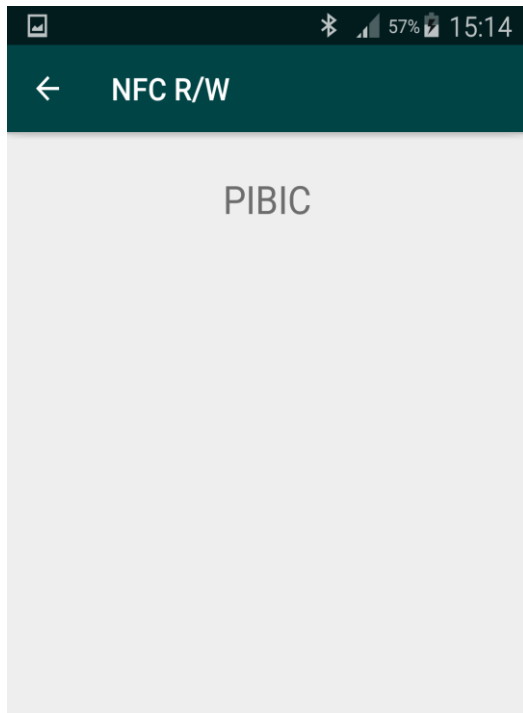


Existem dois botões na parte inferior da janela principal: leitura, representada por um livro e escrita, representada pelo lápis. Pressionando sobre o botão de escrita, a janela para escrita de tag se abre. Apresenta-se, então, uma mensagem indicando aguardo para escrita na etiqueta, além de uma caixa de texto vazia, à qual é destinada a inserção, pelo usuário, do conteúdo que será gravado. Considerando que se queira gravar o conteúdo "PIBIC" na etiqueta. Primeiramente, o conteúdo deve ser inserido na caixa de texto:



então a etiqueta que será gravada deve ser aproximada ao smartphone. O processo iniciará automaticamente. Após a conclusão da escrita, uma mensagem confirmando a conclusão da escrita será exibida e a aplicação voltará a página principal.

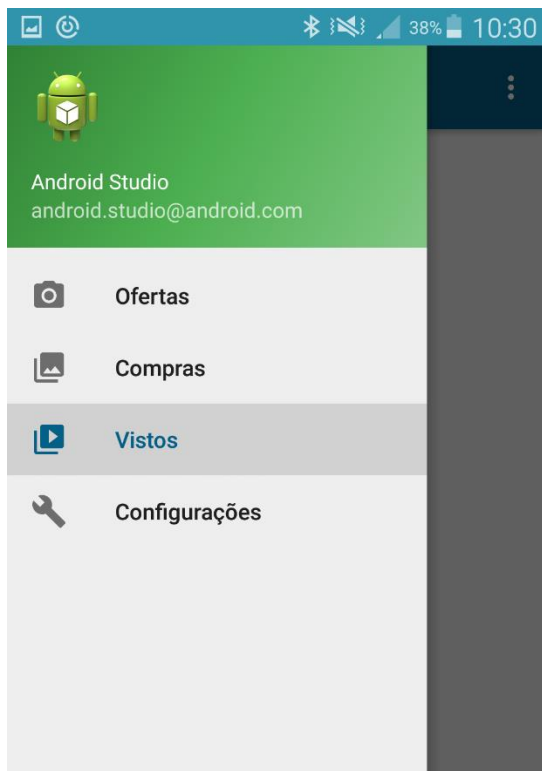
Na tela inicial, ao pressionar o botão de leitura, uma nova janela se abre. Observa-se, então, a mensagem "Passe a TAG NFC", à qual solicita que o usuário aproxime uma etiqueta ao smartphone. No entanto, caso o dispositivo não tenha suporte a NFC, a mensagem citada será substituída por “ Este dispositivo não tem suporte para NFC”. Ao aproximar uma etiqueta, o sistema a detectará, capturará o conteúdo da mesma e finalizará sobrescrevendo inicialmente mensagem mencionada com os dados presentes na tag.



RecStore

A aplicação final desenvolvida envolve a recomendação de produtos com base em dispositivos iBeacons e etiquetas NFC para detectar o contexto do usuário. No caso do iBeacon, este será utilizado para determinar se o usuário está parado em frente a um conjunto à uma vitrine de shopping center ou prateleira de supermercado. Já a etiqueta tem como função estar afixada a um produto e capturar a interação direta do usuário com este. Com base nessas duas formas de interação é possível identificar dois níveis de interesse por parte do usuário. Primeiramente, se ele se interessa por determinado conjunto de produtos ou se um produto em específico é de interesse para ele.

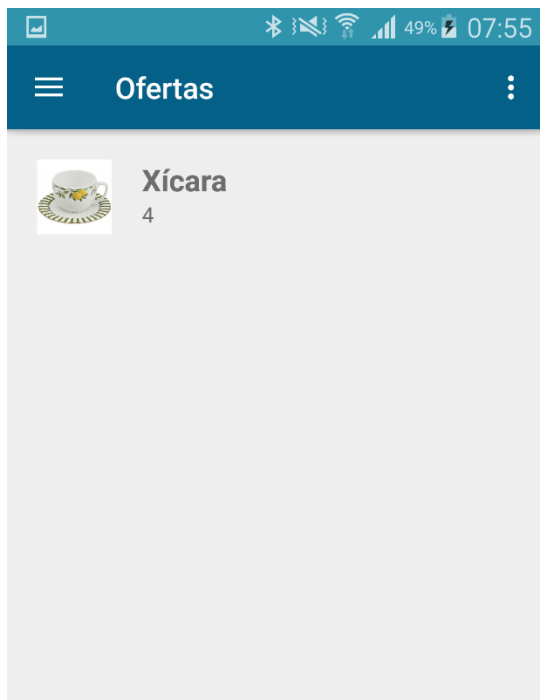
A aplicação foi pensada como um conjunto de janelas que podem ser acessadas por um menu lateral como mostrado na Figura X.



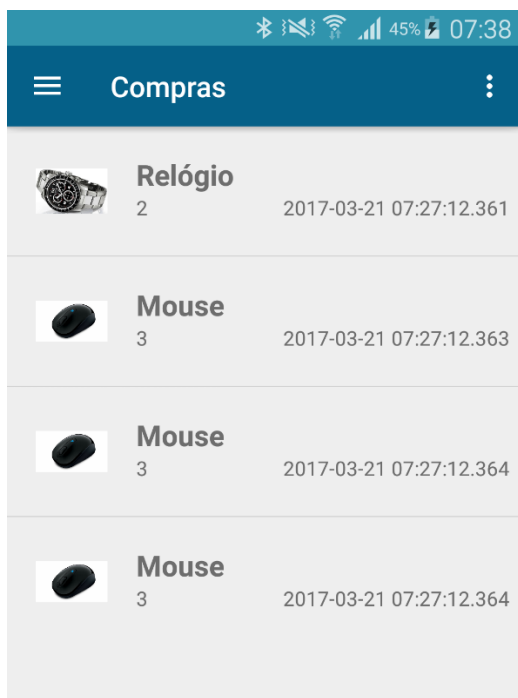
Na aplicação implementou-se seis (6) funcionalidades distintas que envolvem: as interações com os dispositivos beacon e etiquetas NFC, recomendações de ofertas, listagem de compras e de visualizações de produtos, a possibilidade de configuração de parâmetros da aplicação e, por fim, login de usuários. A primeira dessas funcionalidades envolve a exibição de informações relacionadas a um produto representado pelo beacon ou pela etiqueta, tais como nome, código de identificação e preço.



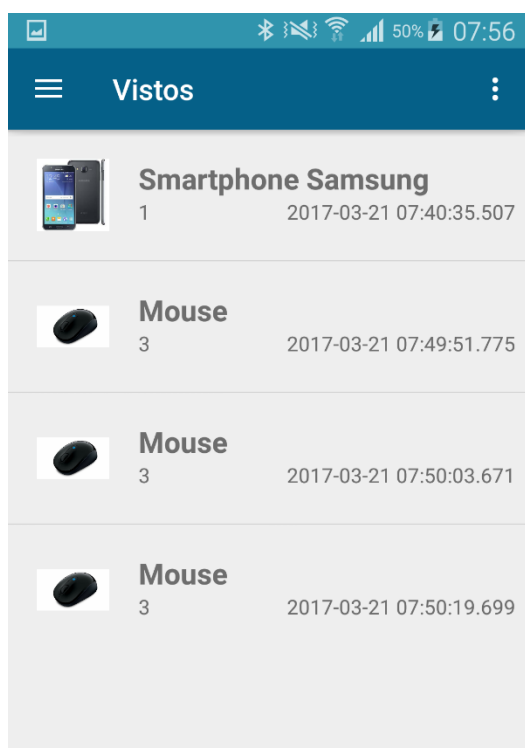
Já a segunda funcionalidade se refere a uma listagem de produtos recomendados a partir de filtragem colaborativa, onde os itens indicados advêm das interações de usuários com gostos mais semelhantes, mas que o usuário em questão ainda não tem conhecimento. A Figura X mostra o funcionamento.



A terceira funcionalidade se refere a listagem de produtos adquiridos pelo utilizador da aplicação com informações de data, o produto em si e quantidade, no entanto, vale destacar que essas informações não são originadas da aplicação, ou seja, são armazenadas por um sistema de compras e o aplicativo aqui apresentado têm o papel de resgatar tais informações e exibi-las ao usuário. A Figura X mostra o funcionamento.



A quarta função é semelhante à anterior, e busca mostrar ao usuário os produtos aos quais ele interagiu seja direta, através dos produtos com etiqueta, ou indireta, através da prateleira com beacon. A quinta funcionalidade permite que se tenha controle sobre dois parâmetros do software, sendo eles o IP do servidor de registro de interações e o tempo mínimo necessário de aproximação para que um beacon seja ativado. A última funcionalidade permite que diferentes usuários utilizem a aplicação em um mesmo dispositivo, além de maior segurança aos dados de cada usuário.



Há uma funcionalidade secundária de configurações. Ela permite que se indique o IP do servidor, que pode ser útil ao realizar testes em diferentes servidores, além da configuração do tempo necessário para ativação do beacon, como mostrado.



A interação com as etiquetas NFC foram implementadas com base na aplicação já descrita anteriormente, onde as classes implementadas foram reutilizadas. A partir dessas classes é possível, então, ter o dado necessário sobre a identificação da etiqueta, aqui descrita como endereço MAC (Medium Access Control).

A detecção de iBeacons, envio de suas respectivas informações, como já descrito, MAC, Major, Minor e RSSI, ao servidor e exibição das recomendações recebidas a partir do serviço de recomendação. Utilizou-se, então, no projeto uma biblioteca de código aberto chamada AltBeacon que possibilita agregar à aplicação a capacidade de interagir especificamente com dispositivos do tipo iBeacon baseados em BLE.

Biblioteca AltBeacon

A biblioteca AltBeacon fornece os formatos e especificações de comunicação de dispositivos de proximidade padrão BLE. Mais informações podem ser encontradas na página Web www.altbeacon.org. A biblioteca é disponibilizada na plataforma de Serviço de Web Hosting Compartilhado, Github®, através do link

<https://github.com/AltBeacon/android-beacon-library>.

Após o *download*, a biblioteca foi utilizada na detecção e na recuperação de informações dos *beacons* próximos, mais especificamente através da opção “Recomendação” que, ao ser selecionada, abre uma nova página e inicia a detecção de dispositivos próximos.

Para um correto uso de toda a gama de funções providas pela AltBeacon se fez necessária a configuração através de um conjunto de instruções (Figura 3):

Figura 3: Configuração da biblioteca AltBeacon

```
45 // Configuração do detector de beacons.
46 beaconManager = BeaconManager.getInstanceForApplication(this); // beaconManager é do tipo BeaconManager.
47 beaconManager.setBackgroundMode(true);
48 beaconManager.setBackgroundScanPeriod(10001);
49 beaconManager.setBackgroundBetweenScanPeriod(11);
50 beaconManager.getBeaconParsers().add(new BeaconParser().setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24,d:25-25"));
51 beaconManager.bind(this);
```

Fonte: Autores

Resumidamente, as linhas descritas na figura acima configuram um gerenciador de *beacons* (linha 46), habilitam o escaneamento em segundo plano (linha 47) durante 1 (um) segundo (linha 48) a cada 1 (um) milissegundo, definindo as características dos *beacons* proprietários que estarão aptos a serem encontrados (linha 50).

Além da etapa de configuração é essencial informar à aplicação as ações que devem ser realizar quando ocorrem eventos como detecção e saída de um *beacon* do raio de cobertura (Figura 4).

Figura 4: Detecção de *beacons*

```
84 @Override
85 public void didRangeBeaconsInRegion(Collection<Beacon> collection, Region region) {
86     // Caso tenha mais de um beacon na região e o número em relação à última vez mudou.
87     if (collection.size() > 0) {
88         LOG(TAG, "Iniciando detecção de beacons: " + collection.size() + " beacon(s) encontrado(s).");
89
90         for (Beacon beacon : collection) {
91             if (!beaconEnviado.contains(beacon)) {
92                 enviaDadoBeacon(beacon); // AsyncTask
93                 beaconEnviado.add(beacon);
94             }
95         }
96     }
97 }
```

Fonte: Autores

A Figura 4 indica que, em todo momento que *beacons* estiverem no raio de detecção e estes não foram detectados anteriormente, seus dados serão enviados ao servidor.

Servidor web

A integração com o servidor web consistiu no envio dos dados capturados dos sensores BLE e das etiquetas NFC para um servidor através do uso de um URL (*Uniform Resource Locator*) utilizando o padrão REST (*Representational State Transfer*), na qual indica o endereço do servidor, alguns parâmetros de configuração próprias da aplicação, além dos dados em si. Além disso, o armazenamento de dados foi efetuado em uma base de dados não relacional a partir da aplicação *Mongo DB*®.

O aplicativo desenvolvido oferece, como já descrito, mais de uma categoria de interação com o servidor, ou seja, listagem de compras e visualizações, resgate de informações sobre um produto etc. Viu-se, portanto, a necessidade de categorizar a forma como eram feitas as consultas REST. Assim, quatro (4) categorias de URL foram criadas: compra, registro, visto, oferta e login, sendo cada uma delas representadas por um número, respectivamente, de um (1) a cinco (5).

Além da diferenciação por tipo de operação, foi necessário distinguir entre dispositivos beacon e etiquetas. Da mesma maneira como as operações, estes forem numerados respectivamente em um (1) e dois (2).

Por fim, tem-se alguns exemplos de URL aceitas pela servidor implementado:

1. `http://ipservidor:8080/RecomendacaoServer?type=1&user_id=5`
2. `http://ipservidor:8080/RecomendacaoServer?type=2&device_tech=1&user_id=3&device_tech=1&device_mac=00:11:22:33:44&beacon_minor=12&beacon_major=34&beacon_rssi=-35`
3. `http://ipservidor:8080/RecomendacaoServer?type=5&user_login=pibic&user_password=cnpq@#2017`
4. `http://ipservidor:8080/RecomendacaoServer?type=3&user_id=7`

No exemplo 1, uma consulta de requisição da lista de compras representada pelo parâmetro *type* para o usuário identificado pelo código cinco (5) como mostra o parâmetro *user_id*. Já no exemplo 2, tem-se o registro de uma interação com um dispositivo beacon, conforme o parâmetro *device_tech*, de um usuário identificado como três (3). Além disso, tem-se informações específicas sobre o beacon, o endereço MAC (00:11:22:33:44) identificado por *device_mac*, o minor (12) descrito por *beacon_minor*, o major (34), informado por *beacon_major*, o RSSI (-35dB), identificado por *beacon_rssi*. No terceiro exemplo é realizada uma consulta de login

para verificação do usuário, neste caso “pibic”, representado por *user_login* com a senha “cpnq@#2017” informada pelo parâmetro *user_password*. Por fim, o exemplo 4 demonstra uma requisição da lista de produtos visualizados pelo usuário.

É notável nos exemplos de consultas HTTP que os parâmetros variam exceto pelo tipo de consulta efetuada (parâmetro *type*). Para cada categoria de consulta há um conjunto específico de parâmetros, que podem ou não existirem em outras categorias. No entanto, é necessário ressaltar que os parâmetros precisam corresponder ao tipo de operação, caso contrário, a aplicação não responderá adequadamente.

Quando o servidor receber fará a verificação do tipo de operação que será realizada e, então, executa o conjunto de ações mais apropriado, conforme Figura X.

```
125 switch (type) {
126 case InteractionDefinition.TYPE_URL_PURCHASE:
127     return purchaseRequest(userId);
128 case InteractionDefinition.TYPE_URL_RECORD:
129     return recordRequest(userId, deviceTech, deviceMac, beaconMinor, beaconMajor, beaconRssi);
130 case InteractionDefinition.TYPE_URL_SEEN:
131     return seenRequest(userId);
132 case InteractionDefinition.TYPE_URL_OFFER:
133     return offerRequest(userId);
134 case InteractionDefinition.TYPE_URL_LOGIN:
135     return loginRequest(userLogin, userPassword);
```

Como é possível visualizar, existe uma classe com atributos estáticos sendo acessados na verificação do tipo de consulta. Como é mais fácil identificar linguagem textual ao invés de números, implementou-se tais atributos para conter os números indicados anteriormente para cada operação. Além disso, para cada categoria de operação há uma chamada de função específica para ela em conjunto com seus parâmetros. No entanto, todas as funções retornam algo. Nessa implementação do servidor, o retorno será uma cadeia de caracteres que será diretamente enviada como resposta para o dispositivo que fez a consulta HTTP. Mais especificamente, a cadeia de caracteres utiliza o formato JSON (), o qual define um conjunto de chaves e valores. Portanto, para cada chamada de função, um conjunto de dados diferentes de JSON será retornado. Por exemplo, a chamada de função para *loginRequest* retorna um JSON no seguinte formato

1. {"result": "granted", user_id = 3}
2. {"result": "denied"}

Nos exemplos, o JSON apresenta um ou dois parâmetros: resultado da verificação (garantido ou negado), e caso o acesso seja conseguido, a identificação do usuário sendo esta mantida durante toda a execução do programa para a criação das demais URLs.

As demais chamadas de funções, isto é, visto, compras, ofertas possuem um formato de JSON muito semelhante contendo um conjunto de produtos. A depender da operação alguns dos parâmetros do produto são dispensáveis, ou ainda, novos dados podem ser necessários. Por exemplo, as consultas de produtos vistos e comprados pode ter os seguintes formatos, respectivamente.

1. { products : [{ "_id" : { "\$oid" : "58cfe9ba61c8d720e570e6a9" }, "product_id" : 3, "product_name" : "Mouse", "product_price" : 35.0, "product_url" : "http://site.com/mouse.png", "timestamp" : "2017-04-15 19:35:53.51" }, { "_id" : { "\$oid" : "58cfe9ba61c8d720e570e6a9" }, "product_id" : 3, "product_name" : "Caneta", "product_price" : 35.0, "product_url" : "http://website.com/caneta.png", "timestamp" : "2017-04-15 19:35:53.517" }] }
2. { products : [{ "_id" : { "\$oid" : "58cfe9ba61c8d720e570e6a8" }, "product_id" : 2, "product_name" : "Relógio", "product_price" : 50.0, "product_url" : "http://imagens.com/relogio.png", "timestamp" : "Sat Apr 15 20:06:12 BRT 2017" }, { "_id" : { "\$oid" : "58cfe9ba61c8d720e570e6a8" }, "product_id" : 2, "product_name" : "Relógio", "product_price" : 50.0, "product_url" : "http://imagens.com/relogio.png", "timestamp" : "Tue Apr 18 10:09:59 BRT 2017" }, { "_id" : { "\$oid" : "58cfe9ba61c8d720e570e6a8" }, "product_id" : 2, "product_name" : "Mouse", "product_price" : 35.0, "product_url" : "http://site.com/mouse.png", "timestamp" : "Wed Apr 26 20:55:19 BRT 2017" }] }

Em ambos os exemplos, há um JSON que contém uma lista de produtos representado por *products*. Cada produto é representado por um JSON contido na lista, além de possui os seguintes parâmetros: o *_id* criado automaticamente pela base de dados, a identificação do produto (*product_id*), o nome do produto (*product_name*), o preço (*product_price*), o endereço URL para uma imagem ilustrativa do produto (*product_url*) e um registro do data e hora que o produto foi visualizado ou comprado (*timestamp*).

Por fim, o registro de interações armazena os dados do usuário e do dispositivo além de retornar apenas um JSON com dados de um produto em específico, com os

parâmetros indicados anteriormente, exceto o *timestamp*.

Base de dados do servidor

A base de dados é responsável por armazenar informações a cerca das interações, dos dispositivos cadastrados bem como os respectivos produtos por eles representados e os usuários da aplicação. Para essa aplicação fez-se uso, como já descrito da aplicação aberta de bancos de dados não-relacional MongoDB, na qual permite armazenar dados com base em NoSQL. A partir desse conceito, é possível criar bases com grande volume de dados não estruturados, ou seja, sem um conjunto específico de características armazenadas.

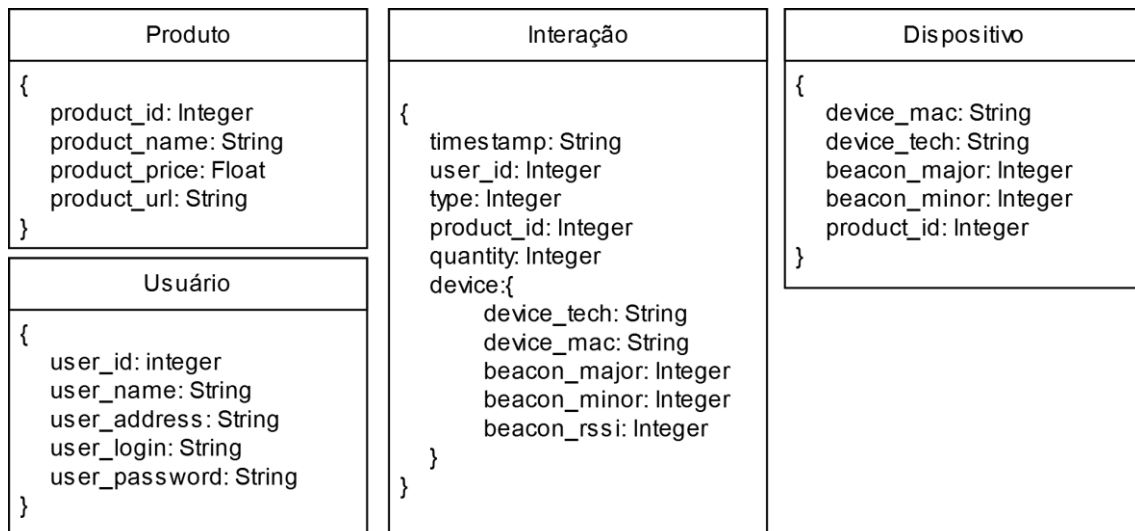
O NoSQL permite implementar alguns tipos de bases de dados como baseada em documentos, em grafos entre outros. Na aplicação MongoDB faz-se uso de base de dados baseadas em documentos, onde cada documento representa um registro e é composto por um conjunto de pares de chave e valor, da mesma forma com um JSON. Há, também, alguns aspectos semelhantes entre bases não relacionais e relacionais. Por exemplo, as tabelas do modelo relacional são representadas como coleções no modelo relacional, já as tuplas são representadas como documentos (no caso do MongoDB).

Nessa aplicação fez-se o uso de base de dados não-estruturados devido a dinamicidade das interações, ou seja, compras, registros etc., e, desse modo, pode-se em um único conjunto de dados armazenar todas as interações da aplicação. Já o cadastro de produtos, de dispositivos e usuários, em virtude da sua consistência poderia ser implementado em bancos de dados relacionais comuns. No entanto, para fins de simplificação do desenvolvimento e a partir possibilidade da utilização de base não-relacional para os cadastros citados, todo a base do servidor foi implementada em MongoDB.

A base foi implementada considerando quatro coleções: usuário, dispositivo, produto e interação. O primeiro conjunto contém informações sobre cada usuário sendo elas, identificação, nome, login etc. Já o segundo refere-se às informações sobre os dispositivos como endereço MAC, além de dados como major e minor no caso de beacons. Além disso, a diferenciação entre etiqueta e beacon é feito pelo parâmetro de tipo. A terceira coleção se refere aos dados de cada produto, contendo sua identificação, nome, preço e o endereço da URL da imagem ilustrativa. A última coleção, que recebe e provê constantemente dados para a aplicação móvel, registra as diversas interações do

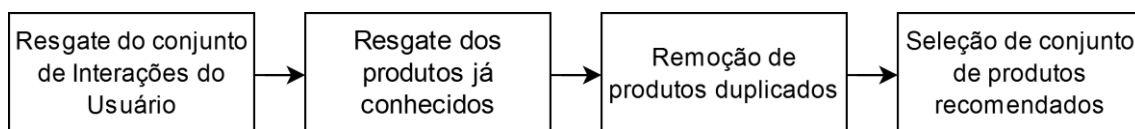
usuário com etiquetas e beacons. Percebe-se que ela guarda identificadores de usuário, do produto e do dispositivo responsável pelo registro.

A Figura X ilustra as coleções bem como as informações armazenadas e seus respectivos tipos de dados. No entanto, vale lembrar que parâmetros foram definidos apenas para se manter um padrão, sendo que em cada coleção qualquer outro tipo de registro com dados totalmente diferentes poderia ser armazenado sem problemas.



Base de recomendações

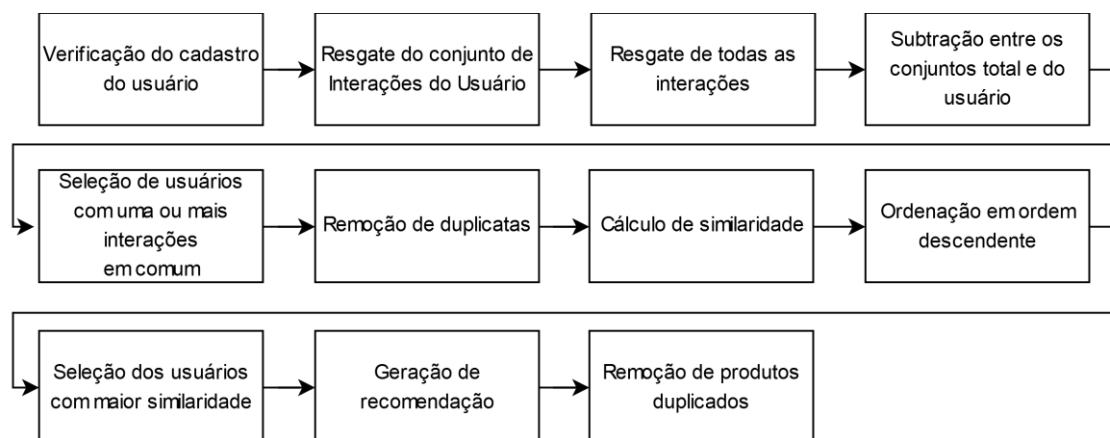
Conforme os usuários interagem com a aplicação móvel, os registros destes são armazenados na base de dados já descrita e recomendações podem ser feitas, a partir desses dados. Nesse trabalho, primeiramente e para fins de simplificação do desenvolvimento da aplicação desenvolveu-se um algoritmo de recomendações aleatórias, onde cada item proposto ao usuário era conhecido por ele. Para um entendimento do algoritmo, considere a Figura X.



Inicialmente, todas as interações do usuário, sejam elas compras ou visualizações são resgatadas e mantidas em uma lista. Em seguida, com base na lista de interações, é extraído um conjunto de produtos relacionados a cada uma delas. Como o

usuário pode interagir múltiplas vezes com um certo item, fez-se necessário o terceiro estágio, de remoção de duplicatas. Por fim, com um conjunto de produtos definido, realizou-se uma seleção de até cinco (5) produtos para serem exibidos como recomendações para o usuário.

Com a aplicação já implementada, o algoritmo de recomendação foi aprimorado. Nessa reformulação, um algoritmo mais eficiente de recomendação foi utilizado, ou seja, o algoritmo de filtragem colaborativa. As etapas realizadas são ilustradas na Figura X.



A primeira etapa consiste na validação do usuário, ou seja, se este está cadastrado no sistema ou não. Com a identidade do usuário assegurada é possível prosseguir com a seleção dos registros de interação relacionados a ele, bem como todos os registros de interação. Desse modo, é possível definir os produtos relacionados às interações bem como os usuários que possuem alguma interação registrada. O próximo passo então é subtrair o conjunto de interações do usuário do conjunto com todas as interações, para se ter um novo conjunto com as interações dos demais usuários. A partir disso, é possível obter uma lista com todos os usuários que interagiram com o sistema. No entanto, como já dito, diversas interações podem estar registradas para um mesmo usuário ou produto. Portanto, se faz necessária a remoção de usuários duplicados na lista. O próximo passo, então, é efetuar o cálculo de similaridade entre cada um dos demais usuários e o usuário atual. Na aplicação fez uso do método do cosseno, baseado no produto escalar Euclidiano. Com a similaridade definida, é necessário obter-se os usuários com maior similaridade. Fez-se a seleção de até cinco (5) usuário com maior similaridade e, para cada um destes, um conjunto de produtos foi recomendado com base na subtração de conjuntos, ou seja, alguns produtos que o usuário ainda não interagiu são selecionados para a recomendação. Em seguida, os conjuntos de recomendação são unidos para formar o conjunto final de recomendação. No entanto, é possível que produtos tenham sido

recomendados mais que uma vez por usuários semelhantes distintos. Por isso, é necessária a remoção de produtos duplicados da lista. Ao final, a lista sem duplicatas é enviada para a aplicação móvel como recomendação.

4. ATIVIDADES DO BOLSISTA

A seguir são apresentadas as atividades do bolsista **Thiago Raulino Dal Pont** conforme descritas no plano de atividades definido no projeto. Para cada atividade é indicado se a mesma foi realizada ou não. Em caso negativo é provida uma explicação.

1. Estudos e pesquisa bibliográfica:

Revisão da literatura científica relevante para o desenvolvimento do projeto. [Atividade Realizada].

2. Estudo de *frameworks* e bibliotecas que possibilitem o desenvolvimento da arquitetura de serviços:

Estudo da linguagem de programação necessária ao desenvolvimento do projeto. Estudo de ambientes, *frameworks* ou bibliotecas que auxiliem no desenvolvimento do projeto. [Atividade Realizada].

3. Estudo de tecnologias capazes de produzir contexto, por exemplo, sensores *Bluetooth* que facilitem a recomendação de conteúdo:

Estudo, desenvolvimento de aplicação e testes de comunicação com sensores sem fio. [Atividade Realizada]

4. Proposição e desenvolvimento de um modelo de dados capaz de armazenar perfis de usuários visando subsidiar a recomendação de conteúdo:

Análise dos requisitos necessários para a elaboração do modelo de dados. Criação do modelo em banco de dados relacional. [Atividade Realizada].

5. Elaboração de um cenário com dados que possibilitem a demonstração de viabilidade da arquitetura proposta:

Desenvolvimento de uma aplicação de carga com o objetivo de importar um conjunto de dados disponível publicamente que possibilite a definição de cenário voltado à recomendação de conteúdo. Empresas como Netflix® ou o site Movielens® disponibilizam quantidades expressivas de dados que possibilitam a avaliação, tanto de algoritmos quanto de desempenho na tarefa de recomendação.

Obs: para o cenário foi desenvolvida uma aplicação com o intuito de realizar a produção de dados aleatórios que pudessem ser carregados no modelo de dados. Neste sentido, as

bases públicas não foram utilizadas. Visto a abrangência do trabalho decidiu-se por dedicar mais tempo no entendimento e desenvolvimento da solução de coleta de dados baseado em sensores sem fio ao invés de dispendar tempo no entendimento destas bases que possuem modelos complexos, bem como, na realização de um sistema de carga para suportar tal demanda. [Realizada, contudo sem considerar bases de dados públicas providas, por exemplo, pelo Movielens®].

6. Desenvolvimento do protótipo da arquitetura de serviços voltada à recomendação de conteúdo:

Estudo das tecnologias que possibilitam a definição de serviços. Desenvolvimento da camada de serviço que possibilite a recomendação de itens (artigos ou produtos) a partir da invocação de serviços. Desenvolvimento dos processos de *backend* para geração de recomendações baseados no interesse ou perfil do usuário. Evolução do protótipo visando mapear cenários mais complexos. [Atividade realizada]

7. Integração da arquitetura de serviços com a tecnologia de produção de contexto:

Desenvolvimento de uma aplicação móvel que possibilite a configuração e a comunicação com sensores sem fio e com a camada de serviços da arquitetura. A aplicação móvel também interage com os serviços de modo a receber recomendações a partir de determinado item de interesse. [Atividade realizada]

8. Avaliação do desempenho da arquitetura em manipular quantidades expressivas de dados:

Elaboração de uma aplicação de teste que possibilite avaliar o desempenho da arquitetura.

Obs: Atividade não realizada. No projeto decidiu-se pelo aprofundamento do estudo e desenvolvido da aplicação móvel a interação desta com sensores sem fio. Está é uma área promissora, uma vez que possibilita que sistemas tenham ciência do contexto em que estes estão inseridos. Foram desenvolvidas duas aplicações para avaliar a interação e a correta indicação/sugestão de itens.

9. Documentação do projeto:

Elaboração dos relatórios parcial e final do projeto. Elaboração de documentação necessária ao encaminhamento da arquitetura desenvolvida para registro junto ao DIT/UFSC.

Obs: A documentação para encaminhamento do projeto ao DIT/UFSC está em fase final de produção e deve ser submetida em Setembro/Outubro de 2016.

5. AVALIAÇÃO DO BOLSISTA EM RELAÇÃO AO PIBIC

Apesar de ter iniciado o projeto este ano tenho muito a agradecer pela oportunidade de estar vinculado a um projeto de pesquisa. A experiência de participar do Programa Institucional de Bolsas de Iniciação Científica (PIBIC) demonstrou ser de grande valia para a minha formação acadêmica.

A experiência tem sido extremamente relevante visto que o projeto vai de encontro aos meus interesses e tem relação com o curso de Engenharia de Computação. Demonstrou ser atual e promoveu diversos desafios durante o período em que participei. Tais desafios me fizeram conhecer mais a área de Sistemas de Recomendação e áreas mais técnicas de suporte ao projeto como um todo, a saber, linguagens e técnicas de programação, arquitetura de software baseada em serviços e coleta de dados a partir de sensores sem fio.

Vejo grande potencial de evolução e em função disso irei continuar no projeto referente ao período de 2016/2017. Entre as evoluções do projeto citam-se, o estudo e a interação com outras tecnologias de sensores (utilizamos Bluetooth® neste projeto e esperamos utilizar NFC – *Near Field Communication* no projeto que se inicia), aprimoramento dos serviços de recomendação considerando novos aspectos e algoritmos e melhorias na camada de serviço permitindo a coleta de grandes volumes de dados.

Por fim, gostaria de agradecer a UFSC/CNPq pela oportunidade e ao orientador, Prof. Alexandre Gonçalves, pelo profissionalismo, conhecimento e condução do projeto.

6. CONSIDERAÇÕES FINAIS

O objetivo geral do projeto constituiu na proposição de uma arquitetura baseada em serviços como o intuito de promover subsídios para a tarefa de recomendação de itens considerando informações estruturadas ou não estruturadas.

Visando cumprir o objetivo, foram realizadas pesquisas bibliográficas pelo bolsista para entender melhor o contexto das áreas de suporte ao projeto, Sistemas de Recomendação e Recuperação de Informação.

Para a realização e desenvolvimento do projeto outras áreas também foram investigadas, entre elas, linguagens e técnicas de programação, arquitetura de software baseada em serviços e coleta de dados a partir de sensores sem fio. A partir desta investigação definiu-se a linguagem de programação, *frameworks* de suporte e tecnologias. Para o desenvolvimento a linguagem de programa Java® foi utilizada. Quanto aos *frameworks*, foram utilizados o ambiente de programação Eclipse® para desenvolvimento da camada de serviços com base no padrão REST, o ambiente de desenvolvimento de aplicações móveis Android Studio® e a biblioteca AltBeacon® para a comunicação com sensores sem fio Bluetooth®. Para o armazenamento dos dados referentes às interações e recomendações utilizou-se um banco de dados relacionado (MySQL®), enquanto que para o armazenamento de conteúdo não estruturado (documentos) foi utilizada a biblioteca Apache Lucene®.

A partir da definição da infraestrutura foi desenvolvida a arquitetura de serviços proposta neste trabalho. De modo geral, a arquitetura é composta por funcionalidades que possibilitam a indexação e recuperação de conteúdo textual, o registro das interações dos usuários a partir da camada de aplicação e os serviços de recomendação, sendo um baseado em conteúdo e o outro utilizando a abordagem de filtragem colaborativa.

Para testar a camada de serviços disponibilizada através da arquitetura foram desenvolvidas duas aplicações visando definir cenários para avaliação. Para a recomendação de conteúdo textual foi elaborada uma aplicação Web (estilo mecanismo de busca). Esta aplicação possibilita ao usuário consultar documentos através de palavras-chave e receber sugestões de documentos relacionados aos documentos previamente recuperados. Além disso, foi implementada uma aplicação móvel (App)

que interage com sensores sem fio e a partir disso, se comunica com a camada de serviços recebendo recomendações de potenciais itens de interesse.

Considerando as aplicações de teste e os cenários elaborados pôde-se verificar que a arquitetura atendeu aos objetivos estabelecidos. Tanto a recomendação de documentos quanto a recomendação de produtos apresentaram resultados adequados no que tange a qualidade da informação retornada.

Como trabalhos futuros vislumbra-se a evolução da pesquisa no que tange a novos algoritmos de recomendação, principalmente, as abordagens que permitem mineração de opiniões e a ciência de contexto.

No que tange ao desenvolvimento da arquitetura de serviços novas funcionalidades podem ser vislumbradas, bem como, o acréscimo de novos serviços. Entre os novos serviços destacam-se os destinados a promover suporte aos tomadores de decisão. Considerando o volume de dados obtidos, áreas como *Big Data* e *Data Mining* podem promover suporte para um melhor entendimento do comportamento de usuários em determinado domínio de problema.

Um das áreas de grande interesse atualmente é a Internet das Coisas. Essencialmente, esta se refere ao conjunto de tecnologias que permite uma ampla comunicação entre sistemas através de sensores sem fio. Quando se fala em Sistemas de Recomendação sensores sem fio promovem meios de definir mais claramente o contexto de interação entre usuários e itens físicos. Com este objetivo no projeto atual utilizou-se a tecnologia Bluetooth®. Outra possibilidade seria a utilização da tecnologia NFC, pois esta promove uma maior precisão na interação entre usuários e sensores por meio de algum dispositivo.

7. REFERÊNCIAS

ANDRÉ, P. B.; LAURINDO, S. M. Uma arquitetura de serviços voltada à recuperação de informação e recomendação de conteúdo. 2014. 77 f. Trabalho de Conclusão de Curso – Curso de (Tecnologias da Informação e Comunicação, Centro Araranguá, Universidade Federal de Santa Catarina, Florianópolis, 2014.

AGHAEI, Sareh; NEMATBAKHS, Mohammad Ali; FARSANI, Hadi Khosravi. Evolution of the World Wide Web: From Web 1.0 to 4.0. International Journal of Web & Semantic Technology, v. 3, n. 1, jan. 2012.

BOBADILLA, Jesús et al. Recommender Systems Survey. Journal Knowledge-based Systems, Amsterdam, v. 46, p.109-132, jul. 2013.

CAZELLA, Sílvia César; NUNES, Maria Augusta S. N. ; REATEGUI, Eliseu. A Ciência da Opinião: Estado da Arte em Sistemas de Recomendação. In: André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski. (Org.). Jornada de Atualização de Informática-JAI 2010- CSBC2010. Rio de Janeiro: PucRIO, 2010, v. 1, p. 161-216.

FERNEDA, Edberto. Recuperação de informação: análise sobre a contribuição da ciência da computação para a ciência da informação. 2003. 147 f. Tese (Doutorado) - Curso de Ciências da Comunicação, Universidade de São Paulo, São Paulo, 2003.

FIGUEIRA FILHO, Fernando M.; GEUS, Paulo Lício de; ALBUQUERQUE, João Porto de. Sistemas de Recomendação e Interação na Web Social. In: I WORKSHOP DE ASPECTOS DA INTERAÇÃO HUMANO-COMPUTADOR NA WEB SOCIAL, Porto Alegre, p.24-27, 2008.

FOLTZ, Peter W.; DUMAIS, Susan T.. Personalized Information Delivery: An Analysis of Information Filtering Methods. Communications Of The Acm. New York, p. 51-60. dez. 1992.

GARCIA, Cássio Alan; FROZZA, Rejane. Sistema De Recomendação De Produtos Utilizando Mineração De Dados. Tecno-lógica, Santa Cruz do Sul, v. 1, n. 17, p.78-90, jan/jun. 2013. Disponível em: <<http://online.unisc.br/seer/index.php/tecnologica/article/viewFile/3283/2692>>. Acesso em: 06 abr. 2014.

HERLOCKER, Jonathan Lee. Understanding and Improving Automated Collaborative Filtering Systems. 2000. 148 f. Tese (Doutorado) - Curso de Philosophy, Faculty Of The Graduate School Of The University Of Minnesota, Minnesota, 2000.

JANNACH, Dietmar et al. Recommender Systems: An Introduction. New York: Cambridge University Press, 2011. 335 p.

LOPES, Gisele Rabello. Sistemas de Recomendação para Bibliotecas Digitais sob a Perceptiva da Web Semântica. 2007. 69 f. Dissertação (Mestrado) - Curso de Programa

de Pós Graduação em Computação, Departamento de Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. An Introduction to Information Retrieval. Cambridge: Online Edition (c) 2009 Cambridge Up, 2009.

MOOERS, C. N. Zatocoding applied to mechanical organization of knowledge. American Documentation. v. 2, p. 20-32, 1951.

O'REILLY, Tim. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. International Journal of Digital Economics, Munich, N.65, p. 17-37, mar 2007.

PRIMO, Alex . O aspecto relacional das interações na Web 2.0. E- Compós (Brasília), v. 9, p. 1-21, 2007.

RICCI, Francesco et al (Ed.). Recommender Systems Handbook. New York: Springer, 2011.

SALTON, Gerard; MCGILL, Michael J.. Introduction to modern information retrieval. New York: Mcgraw-hill, 1983.

SCHAFER, J. Ben; KONSTAN, Joseph; RIEDL, John. 1999. Recommender systems in e-commerce. In Proceedings of the 1st ACM conference on Electronic commerce (EC '99). ACM, New York, NY, USA, 158-166.

SOUZA, Renato Rocha. Sistemas de recuperação de informações e mecanismos de busca na web: panorama atual e tendências. Perspectivas em Ciência da Informação, Belo Horizonte, v. 11, n. 2, p.161-173, maio/ago 2006.

SRINIVASAN, Padmini . THESAURUS CONSTRUCTION. In: FRANKS, William B.; BAEZA-YATES, Ricardo (Ed.). Information Retrieval: Data Structures & Algorithms. New Jersey: Prentice Hall, 1992. Cap. 9. p. 179-292.

VAN RIJSBERGEN, Cornelis. J. Information Retrieval. London: Butterworths, 1975. 147 p.

VIEIRA, Simone Bastos. Indexação automática e manual: revisão de literatura. Revista Ciência da Informação, Brasília, v. 17, n. 1, jan./jun. 1988. Disponível em: <http://repositorio.unb.br/bitstream/10482/12901/1/ARTIGO_IndexacaoAutomaticaManual.pdf> Acesso em: 13 nov. 2013.