

**UNIVERSIDADE FEDERAL DA PARAÍBA**  
**CENTRO DE CIÊNCIAS APLICADAS A EDUCAÇÃO**  
**DEPARTAMENTO DE CIÊNCIAS EXATAS**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**SuggestMe: um sistema de recomendação utilizando web  
semântica para evitar o *cold start***

Danilo de Araújo Formiga  
Orientador: Prof. Me. Yuri Malheiros

RIO TINTO – PB  
2014

Danilo de Araújo Formiga

**SuggestMe: um sistema de recomendação utilizando web  
semântica para evitar o *cold start***

Monografia apresentada para obtenção do título de Bacharel à banca examinadora no Curso de Bacharelado em Sistemas de Informação do Centro de Ciências Aplicadas e Educação (CCAEE), Campus IV da Universidade Federal da Paraíba.  
Orientador: Prof. Me. Yuri Malheiros.

RIO TINTO – PB  
2014

Ficha catalográfica preparada pela Seção de Catalogação e Classificação da Biblioteca da UFPB

F723s Formiga, Danilo de Araújo.

SuggestMe: um sistema de recomendação utilizando web semântica para evitar o *cold start*. / Danilo de Araújo Formiga. – Rio Tinto: [s.n.], 2014.

45 f.: il. –

Orientador: Prof. Ms. Yuri Malheiros.  
Monografia (Graduação) – UFPB/CCAIE.

1. Web semântica. 2. Sistema de recomendação - informática. 3. Cold start.

UFPB/BS-CCAIE

CDU: 004.43(043.2)

DANILO DE ARAÚJO FORMIGA

**SuggestMe: um sistema de recomendação utilizando web  
semântica para evitar o cold start**

Trabalho de Conclusão de Curso submetido ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba, Campus IV, como parte dos requisitos necessários para obtenção do grau de BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Assinatura do autor: \_\_\_\_\_

**APROVADO POR:**

---

Orientador: Prof. Me. Yuri de Almeida Malheiros Barbosa  
Universidade Federal da Paraíba – Campus IV

---

Prof. Me. Marcus Williams Aquino de Carvalho  
Universidade Federal da Paraíba – Campus IV

---

Prof. Me. Vanessa Farias Dantas  
Universidade Federal da Paraíba – Campus IV

RIO TINTO – PB  
2014

“Nothing can stop the man with the right mental attitude from achieving his goal; nothing on earth can help the man with the wrong mental attitude.”

Thomas Jefferson.

## AGRADECIMENTOS

Primeiramente, agradecer a Deus por toda a força e entendimento que me foi dado durante estes 4 anos no curso de Sistemas de Informação.

Aos meus pais e a minha irmã, que sempre torceram por mim. Incentivando sempre que necessário e dando toda a ajuda necessária para que eu pudesse estudar em outra cidade.

A minha esposa, por todo seu auxílio nos meus estudos, mostrando-se sempre presente em tudo que precisei dentro da universidade e no desenvolvimento deste projeto.

Ao orientador desta monografia Yuri Malheiros, por todo o apoio dado a mim, durante os estudos e elaboração da aplicação, mostrando-se presente quando necessário.

Um agradecimento mais que especial a Raphael Diniz, Pablo Lima, Digenaldo Neto, José Paulo Dantas, Kelson Victor e Rennan Felizardo, que moraram comigo durante toda minha estadia em Rio Tinto, dividindo experiências e contribuindo com minha formação.

Agradeço também a Elmano Neto, Rafael Farias, João Hellis, Smith Ascari, Thiago Oliveira, Vicente Neto, Flavinho Guedes, por todo o conhecimento compartilhado comigo, e disposição para auxiliar sempre que necessário.

A todos os meus professores e coordenadores do curso, que contribuíram de forma direta para minha formação, tanto profissional quanto pessoal.

A todos os amigos, funcionários, professores e colegas que fazem parte do DCE da Universidade Federal da Paraíba que de uma forma ou de outra, contribuíram para a conclusão deste trabalho.

## RESUMO

Sistemas de recomendação estão cada vez mais presentes no dia a dia dos usuários na Internet, eles são bastantes usados em *e-commerce* e diversas outras aplicações, sendo responsáveis por automatizar o processo social natural de recomendação e grande influenciador na decisão final do usuário.

Entretanto, os sistemas de recomendação encontram diversos desafios para realizar recomendações no início, pois, não existem dados suficientes. Este problema é denominado de cold start.

Para solucionar esse problema, uma solução comum é aplicar questionários para obter dados iniciais do usuário, entretanto, isto cria mais uma barreira para o cadastro do usuário, introduzindo passos extras para criação de uma conta de usuários em um sistema.

Este trabalho se propõe a criar um sistema de recomendação que resolva o problema do cold start usando dados disponíveis na web. É possível obter os dados do usuário através de uma rede social e com isso, realizar a recomendação utilizando web semântica.

Foi implementado um protótipo resolvendo o problema do cold start obtendo os dados do *Facebook* e realizando recomendações utilizando web semântica através do *Freebase*.

Palavras chaves: *Cold start*, Web semântica, Sistema de recomendação, *Freebase*.

## ABSTRACT

Recommender systems are increasingly present in the daily lives of users on the Internet, they are used in e-commerce and various other applications, automating the natural social process of recommendation and influencing in the final decision of the user.

However, the recommendation systems have many challenges to make recommendations at the beginning, because there is not sufficient data. This problem is called cold start.

To solve this problem, a common solution is to apply questionnaires to obtain initial user data, however, this creates another barrier for user registration, introducing extra steps to create a user account on a system.

This work proposes a recommendation system to solve the problem of cold start using data available on the web. You can obtain the user data through a social network and therefore, perform the recommendation using semantic web.

It was implemented a prototype solving the problem of cold start getting Facebook data and performing recommendations using semantic web via Freebase.

Keywords: *Cold start*, Semantic web, Recommender system, *Freebase*.

## **LISTA DE FIGURAS**

Figura 1 – Web 1.0

Figura 2 – Web 2.0

Figura 3 – Resultado da busca pelos gêneros da banda Nirvana.

Figura 4 – Busca de Bandas pelo gênero Reggae

Figura 5 – Resultados persistidos no banco de dados

Figura 6 – Importação do Bootstrap

Figura 7 – Resultado da busca de álbuns da banda The Police

Figura 8 – Diagrama de casos de uso da utilização do sistema

Figura 9 – Tela de Login do Sistema

Figura 10 – Tela principal do sistema

Figura 11 – Demonstração de uma banda que não existe no Freebase

Figura 12 – Bandas curtidas do usuário Elmano

Figura 13 – Recomendação do usuário Elmano Neto

## **LISTA DE TABELAS**

Tabela 1 – Matriz de comparação de gêneros musicais

Tabela 2 – Matriz de comparação ordenada

Tabela 3 – Cronograma do desenvolvimento do trabalho

Tabela 4 - Bandas Curtidas e Sugeridas ao usuário Elmano Neto

Tabela 5 - Bandas Curtidas e Sugeridas ao usuário Fernando Brito

Tabela 6 - Bandas Curtidas e Sugeridas ao usuário Gilmar Junior

Tabela 7 - Bandas Curtidas e Sugeridas a usuário Nathalia Dantas

Tabela 8 – Bandas Curtidas e Sugeridas ao usuário Wennio Kelson

## LISTA DE SIGLAS

<i>WWW</i>	<i>World Wide Web</i>
SR	Sistemas de Recomendação
SI	Sistemas de Informação
SIG	Sistemas de Informação Gerenciais
<i>W3C</i>	<i>World Wide Web Consortium</i>
<i>RSS</i>	<i>Really Simple Syndication</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>SQL</i>	<i>Metaweb Query Language</i>
<i>JSON</i>	<i>Javascript Object Notation</i>

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>1</b>
1.1 MOTIVAÇÃO.....	1
1.2 CARACTERIZAÇÃO DO PROBLEMA.....	1
1.3 OBJETIVOS DO TRABALHO.....	2
1.3.1 OBJETIVO GERAL.....	2
1.3.2 OBJETIVOS ESPECIFICOS.....	2
1.4 ESTRUTURA DO TRABALHO.....	3
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>4</b>
2.1 <i>WEB SEMÂNTICA</i> .....	4
2.1.1 EVOLUÇÃO DA WEB.....	5
2.2 <i>SISTEMAS DE RECOMENDAÇÃO</i> .....	7
2.2.1 BASEADO EM CONTEÚDO.....	7
2.2.2 SISTEMAS COLABORATIVOS.....	8
2.2.3 SISTEMAS HÍBRIDOS.....	8
<b>3 TRABALHOS RELACIONADOS.....</b>	<b>9</b>
3.1 CORRELAÇÃO DE ITENS E SIMILARIDADE DE USUÁRIOS.....	9
3.1 RECOMENDAÇÕES BASEADAS NA COMUNIDADE.....	10
3.1 RECOMENDAÇÕES EM GRUPO.....	11
<b>4 IMPLEMENTAÇÃO.....</b>	<b>11</b>
4.1 MOTIVAÇÃO.....	11
4.2 DIAGRAMA DE CASOS DE USO.....	12
4.3 DESENVOLVIMENTO.....	12
4.4 CALCULO DE SIMILARIDADE.....	13
4.5 TECNOLOGIAS UTILIZADAS.....	16

<i>4.5.1 PHP</i> .....	16
<i>4.5.2 API FACEBOOK</i> .....	17
<i>4.5.3 JQUERY</i> .....	17
<i>4.5.4 BOOTSTRAP</i> .....	17
<i>4.5.5 API DO YOUTUBE</i> .....	18
<i>4.5.6 FREEBASE</i> .....	18
4.5.6.1 METAWEB QUERY LANGUAGE (MQL).....	19
<i>4.5.7 TELAS</i> .....	19
<b>5 RESULTADOS</b> .....	<b>24</b>
<b>6 CONCLUSÃO</b> .....	<b>27</b>
6.1 DIFICULDADES.....	28
6.2 SUGESTÕES PARA TRABALHOS FUTUROS.....	29

# 1 INTRODUÇÃO

## 1.1 Motivação

Sistemas de recomendações são utilizados rotineiramente pelas pessoas, alguns mais explicitamente, outros menos. Por exemplo, uma pesquisa em um motor de busca, uma sugestão de uma música em um site de *streaming*, uma sugestão de compra em sites de *e-commerce*. Todos estes sistemas têm como objetivo auxiliar o usuário na escolha e descoberta de novas músicas, produtos, etc. Alguns usam o perfil do usuário unicamente, outros analisam a similaridade dos usuários, enquanto outros fazem as duas coisas.

Quando alguém deseja adquirir algo, geralmente solicita a opinião de outra pessoa, ou obtém a sugestão em algum local e mediante estas sugestões, toma a decisão. Este é um processo social natural, que sempre esteve presente no dia a dia das pessoas, antes mesmo da tecnologia. Os sistemas de recomendação tentam automatizar esse processo social natural utilizando a tecnologia. (RESNICK, P. VARIAN, H., 1997).

Um grande desafio de sistemas de recomendações é o *cold start*, caracterizado pela falta inicial de dados no sistema, que dificulta iniciar as sugestões aos seus usuários. O problema em não ter dados iniciais é como calcular a similaridade, as preferências do usuário, ou seja, não saber o que recomendar.

Os sistemas de recomendações podem ser divididos em 3 tipos: com base em conteúdo, colaboração, ou ainda híbridos, que seriam a interseção da recomendação baseada em conteúdo com a recomendação baseada em colaboração.

## 1.2 Caracterização do Problema

É notória a presença das lojas virtuais de vários nichos na Internet hoje. Com isso, tem-se um mercado cada vez mais competitivo e que tenta trazer mais facilidade e comodidade a seus clientes (RICCI, F.; ROKACH, L.; SHAPIRA B., 2011). Por exemplo, a *Amazon* tem um sistema de recomendação de livros (JANNACH, D., 2006) para usuários específicos ou grupos de usuários. Além dessas empresas, existe uma outra fatia do mercado virtual que tem como objetivo propaganda de produtos personalizados ou sugestões aos seus usuários.

Ambos têm um objetivo em comum: apresentar um item mais relevante ao usuário. Mas, como iniciar as recomendações se o sistema ainda não possui nenhuma informação sobre o usuário? Este problema é chamado de *cold start* (partida a frio) (STUART, E.; MIDDLETON, H. A; DAVID, C. R., 2002).

O *cold start* acontece no início dos sistemas de recomendação, como o próprio nome sugere, causado pela falta de dados para gerar recomendações para os usuários, o sistema acaba tendo uma recomendação pobre. Uma possível solução é utilizar um banco de dados já existente que contenha informações que possam ser utilizadas para gerar as recomendações. Mesmo os sistemas de recomendações colaborativos, que recomendam baseado nas interações entre os usuários, tendem a ter a mesma dificuldade pois os usuários não interagiram com o sistema para que possam colaborar com outros usuários.

Além do *cold start*, atualmente, os pesquisadores encontram dois problemas: escalabilidade e recomendações ruins. A escalabilidade preocupa grandes sites como Amazon, Google, Netflix, Ebay, pois estes detêm um grande volume de dados e não conseguem calcular recomendações precisas em um curto espaço de tempo (SARWAR, B. et al., 2001). Este mesmo problema aparece, em escala menor, no protótipo deste trabalho. Outro problema é como melhorar as recomendações para os seus clientes e evitar as recomendações ruins que podem tirar a credibilidade dos sistemas de recomendações. Percebe-se que são problemas opostos, pois para garantir uma recomendação bastante precisa, deveria ser analisado um grande volume de dados e para isso o sistema teria que ser bastante escalável, o que é complexo quando se tem muitas informações a serem analisadas.

Com essa introdução, será explanado o que pode ser feito para resolver esses problemas e conseqüentemente fornecer uma melhor experiência ao usuário.

## **1.3 OBJETIVOS DO TRABALHO**

### 1.3.1 Objetivo Geral

O objetivo deste trabalho é criar um protótipo que utilize web semântica para fazer recomendações de bandas musicais. Este protótipo enfrentará o problema do *cold start* utilizando bandas curtidas pelo usuário através de redes sociais.

### 1.3.2 Objetivos Específicos

Para atingir o objetivo geral, os objetivos específicos necessários são:

- Conectar o protótipo com uma rede social para obter as preferências musicais do usuário.
- Realizar consultas em uma base de dados colaborativa, com os dados obtidos da rede social.
- Realizar consultas em uma API de vídeo para obter vídeos das bandas sugeridas ao usuário.
- Criar perfil musical dos usuários.
- Calcular similaridade entre perfil do usuário e bandas/artistas.

## 1.4 ESTRUTURA DO TRABALHO

Nesta seção, será feita uma breve explicação dos capítulos deste trabalho, assim como o que será abordado em cada um.

- Capítulo 2: Explica toda a fundamentação teórica necessária para o entendimento do trabalho. Nele são apresentados os conceitos de sistemas de recomendação, web semântica, *cold start* e evolução da web.
- Capítulo 3: Neste capítulo são demonstrados trabalhos relacionados que se propuseram a resolver o *cold start*. As soluções avaliadas foram: correlação de itens e similaridade de usuários, recomendações baseadas em comunidade, e recomendações em grupo.
- Capítulo 4: São tratados detalhes do desenvolvimento do protótipo, explicando cada tecnologia que foi utilizada, individualmente.
- Capítulo 5: Neste capítulo, será mostrado os resultados que foram obtidos no decorrer do trabalho, com as dificuldades que foram identificadas e com sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados com mais detalhes os assuntos que foram pesquisados e que servem de base para esse trabalho, entre eles: web semântica, que demonstra o conceito que surge para evoluir a web 2.0, sistemas de recomendação: seus tipos e aplicações.

### 2.1.1 Web Semântica

Segundo Breitman (2005), a web atual é uma web de documentos. Nesta web, os documentos não são ligados, apenas referenciados por páginas, o que dificulta a leitura pelas máquinas, pois foi pensada apenas para leitura e interpretação por usuários. Entretanto, na web semântica(web de dados) existem ligações mais específicas, entre os dados que podem estar em uma página.

A recuperação das informações na web, possível por meio dos motores de busca (*search engines*), é baseada primariamente em palavras-chave contidas no texto dos documentos originais, o que é pouco eficaz se o objetivo for obter informações que levem em consideração a semântica dos dados. Um exemplo disto é: se for realizada a busca em uma *search engine* que não utilize web semântica dos termos: “qual é o maior país das Américas que não fala inglês” possivelmente os resultados seriam dados isolados de Américas, não fala inglês, maior país. Entretanto, se fosse utilizada web semântica, a aplicação entenderia o vocabulário e semântica empregados na pesquisa e traria o resultado que o usuário deseja obter.

Tim Berners-Lee, criador da *World Wide Web (WWW)* e atual diretor da *World Wide Web Consortium (W3C)*, definiu web semântica como (BERNERS. T.B.; HENDLER; J.; LASSILA, O., 2001):

“Web Semântica é a extensão da web atual na qual à informação é atribuído significado bem definido, permitindo que computadores e pessoas trabalhem melhor em cooperação” (2001).”

Ele cita um exemplo do que seria web semântica: Um usuário precisa marcar uma consulta em um determinado médico. Entretanto, ele não sabe onde tem um médico que

atenda pelo seu plano de saúde e muito menos os horários disponíveis. Nessa hora, entra em ação o sistema com web semântica, pois ele informa o plano de saúde e o tipo de médico que precisa, e ainda diz a sua atual localização para obter resultados mais próximos dele. Tendo isto, o sistema faz buscas baseadas nas melhores avaliações dos médicos, verifica as agendas dos médicos pré-selecionados com os horários que o usuário informou, registra a consulta no horário determinado pelo usuário e reserva automaticamente a agenda do médico naquele horário. Tudo isto baseado nos dados da web semântica.

Com este exemplo, pode-se notar que antes a web, que era apenas um mar de dados e informações desconectadas, começa a evoluir criando conexões entre as informações. Observa-se ainda que antes apenas o usuário compreendia as informações da web e com a proposta da web semântica as máquinas também passam a compreendê-las.

#### 2.1.1.1 Evolução da Web

A web 1.0 foi o passo inicial da nossa WWW, as páginas eram estáticas e os leitores acessavam textos estáticos. (BERNERS-LEE, 2001) afirma que a web 1.0 era considerada “*read-only*”, sem muita interação, como pode ser visto na Figura 1. E é possível perceber que era realmente essa a ideia das pessoas que criavam as páginas naquele tempo, algo simples que expusesse as informações que as pessoas necessitavam a qualquer hora e em qualquer lugar.

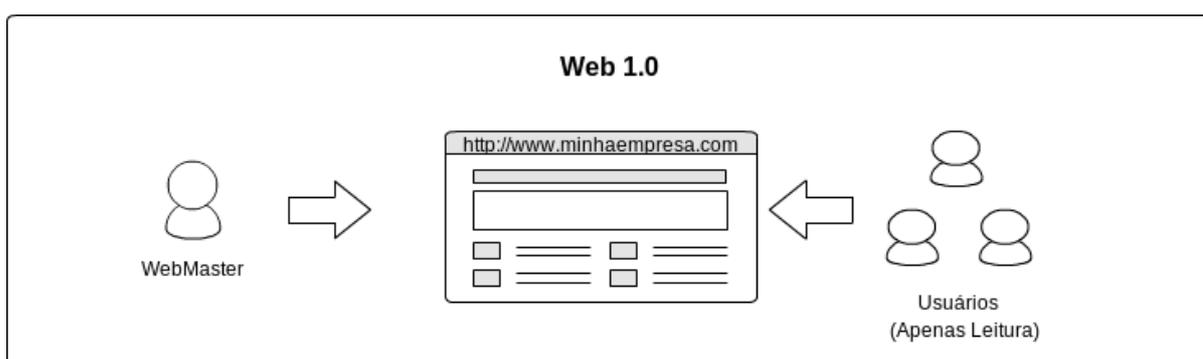


Figura 1 – Web 1.0

Na web 2.0 que predomina até hoje, teve-se a introdução da interação do usuário com as páginas que antes eram estáticas. (BERNERS-LEE, 2001) descreve como web “*read-write*”, ou seja, além de leitura o usuário pode interagir, como é possível observar na Figura 2. Além disso, a web 2.0 trouxe diversas novidades como blogs, *RSS feeds*, *e-commerce*, *e-mail*, *APIs Web*, *Wikis*, entre outros (NAIK, U.; SHIVALINGAIAH, D., 2008). Com isso percebemos que

o usuário está de fato, conectado. Entretanto, a web 2.0 ainda possui diversas limitações para o usuário. Através dela consegue-se pesquisar palavras-chave, mas não consegue responder perguntas mais complexas, por exemplo. É nesse campo que atua a web semântica.

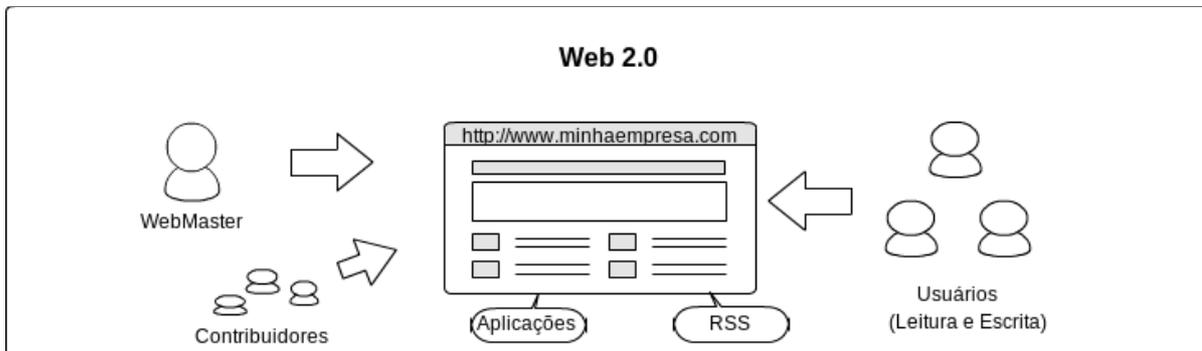


Figura 2 – Web 2.0

Essa limitação da web atual demonstra que não se pode fazer pesquisas como:

- Quem foi Napoleão Bonaparte?
- Qual o menor preço da gasolina nas proximidades?
- Qual é a previsão do tempo na cidade hoje?
- Qual é o preço do dólar?
- Qual o horário do filme x no cinema y?

A web semântica traz um grande diferencial: enquanto a web 2.0 estava focada em melhorar a experiência do usuário com tecnologias como AJAX, jQuery, entre outros, a web semântica foca em tratar as informações que estão sem referências e usar estes artefatos para automatizar a web (SZOMSZOR, M. et al., 2007), tendo em vista que as informações passam a ficar legíveis pelas máquinas, que é o real objetivo da web semântica.

Na web 2.0 os sistemas realizam as pesquisas baseadas nas palavras-chaves, que utilizando este último exemplo seriam: filme x, cinema y, horário. Mas isso não traria exatamente o que usuário desejaria e sim resultados avulsos para cada palavra-chave. O usuário precisaria ainda ler e vasculhar o conteúdo para conseguir obter a resposta que ele desejava.

Entretanto, já é possível notar a mudança com a web semântica, utilizando um recurso da empresa Google que é a pesquisa nos smartphones com Android Kit Kat 4.4. É possível ver que, usando estes mesmos termos em vez de trazer resultados baseados em palavras-chaves, o sistema trará o horário do filme x no cinema y, assim como solicitado. Além desse exemplo, existe o *Google Knowledge Graph* que já é um exemplo de buscador que utiliza a web semântica. Lançado em 2012 ele utiliza os artefatos da web semântica para aprimorar a busca e trazer respostas mais específicas, entretanto ainda é limitado. Isso contextualiza a diferença da web 2.0 para web semântica.

### 2.1.2 Sistemas de Recomendação

Sistemas de Recomendação (SR) são ferramentas que realizam sugestões de determinados itens para um grupo de usuários, ou exclusivamente para um usuário (MAHMOOD, T.; RICCI, F., 2009). Estas recomendações realizadas envolvem diversos processos para obter mais sucesso na sugestão feita ao usuário, que vão desde músicas, produtos para comprar, notícias de jornais e/ou revistas, entre outros.

Uma definição formal para o problema da recomendação foi expressa por (ADOMAVICIUS, G.; TUZHILIN, A, 2005), como:

Seja  $C$  o conjunto de todos os usuários de um determinado sistema, e seja  $S$  o conjunto de todos os possíveis itens que podem ser recomendados como livros, filmes, restaurantes etc. Seja  $u$  a função utilidade que mede o quão útil é um determinado item  $s$  para um determinado usuário  $c$ , i.e.,  $u: C \times S \rightarrow R$ , onde  $R$  é um conjunto ordenado pela utilidade de itens recomendados. Então, para cada usuário  $c \in C$ , procura-se um item  $s' \in S$  que maximiza a utilidade do usuário. Isto pode ser expressado pela equação abaixo:

$$\forall c \in C, \quad s'_c = \arg \max_{s \in S} u(c, s).$$

#### 2.1.2.1 Sistemas de Recomendação baseado em conteúdo

Os SRs baseados em conteúdo aprendem a recomendar itens que são semelhantes a itens que o usuário aprovou no passado. Ele utiliza como base as características destes itens para fazer sugestões de itens com características semelhantes (ADOMAVICIUS, G.; TUZHILIN, A. 2005).

Um exemplo seria a recomendação de um filme para um determinado usuário. Baseado nas informações que o sistema possa ter das preferências de filmes do usuário, é possível verificar a similaridade com gêneros, diretores, atores, assuntos, entre várias outras variáveis, e com isso fazer recomendações baseadas nestas preferências. O problema do *cold start* ataca diretamente aqui: quando não há informação nenhuma das preferências deste usuário, o sistema não consegue efetuar recomendações corretamente. Entretanto, estas preferências podem ser adquiridas explicitamente do usuário, fazendo questionários para obter essas informações, ou implicitamente ao monitorar o comportamento dele.

Nos SRs baseados em conteúdo, é de grande importância a recuperação da informação (BAEZA-YATES, R.; RIBEIRO-NETO, B., 1999). A pesquisa nessa área vem crescendo bastante para auxiliar a obter informações de usuários, e atualmente é possível obter estas informações de diversas fontes como redes sociais, fóruns que o usuário participa, sistemas semelhantes, etc.

#### 2.1.2.2 Sistemas de Recomendação Colaborativos

Estes tipos de sistemas surgiram na década de 90, em universidades e empresas, inspirado nas abordagens colaborativas de filtragem de informação (MEDEIROS, G. R. I. 2013), que consistem em recomendações de diversos tipos de artefatos, tendo como base as preferências de usuários similares ao usuário utilizador.

Como exemplo, é possível demonstrar um sistema de recomendação de filmes, em que o sistema verifica usuários semelhantes ao perfil do usuário utilizador e consulta quais foram suas preferências de filmes. Com isso, os filmes recomendados são os filmes que usuários semelhantes ao usuário utilizador gostaram.

Os sistemas de recomendações colaborativas podem ser classificados em duas categorias: Baseado em memória ou baseado em modelo (BREESE, J. S.; HECKERMAN, D.; KADIE, C. 1998).

#### 2.1.2.3 Sistemas Híbridos

Muitos sistemas utilizam este tipo de abordagem, combinando métodos colaborativos e baseados em conteúdo, o que evita as limitações de ambas as abordagens (BALABANOVIĆ, M.; SHOHAM, Y, 1997). Existem várias formas de utilizar a abordagem híbrida (CLAYPOOL, M. et al. 1999), as duas mais utilizadas são:

1. Construir dois sistemas de formas independentes e realizar a combinação de seus resultados.
2. Construir um único sistema com as duas abordagens, com predominância de uma técnica ou outra.

### **3 Trabalhos Relacionados**

Com o objetivo de entender as possíveis soluções para o problema do *cold start*, foram analisadas diversas iniciativas que se propunham a resolvê-lo. Foram avaliadas soluções utilizando: Correlação de Itens e Similaridade de Usuários, Recomendações Baseadas em comunidade e Recomendações em Grupo.

Nota-se que não existe apenas um meio de tentar solucionar o objeto de estudo deste trabalho. Com base nessa diversidade de soluções serão mostradas as propostas e os resultados obtidos por cada trabalho.

#### **3.1 Correlação de Itens e Similaridade de usuários**

Uma maneira de resolver o *cold start* foi proposta no trabalho de (MAHAPATRA, S.; TAREEN, A.; YANG, Y. 2002). Eles criaram um sistema de recomendação de vinhos. Sendo assim, eles enfrentaram o *cold start* ao fazer recomendações a novos usuários, e como a preferência por vinhos é algo extremamente delicado, eles tinham receio de corroer o fator de confiança ao gerar uma má recomendação.

Para resolver isso, o usuário informa previamente as suas preferências por vinhos. Com isso, o sistema tem uma base de dados inicial para realizar o cálculo de similaridade.

Outra característica deste sistema é que ele procura usuários com preferências semelhantes para efetuar recomendações usando uma abordagem colaborativa. Assim, o sistema proposto no trabalho de (MAHAPATRA, S.; TAREEN, A.; YANG, Y. 2002), utiliza uma abordagem híbrida para garantir uma melhor recomendação.

Para realizar o cálculo de similaridade, eles utilizaram o coeficiente de Jaccard (MAHAPATRA, S.; TAREEN, A.; YANG, Y. 2002), que compara o número de vinhos em comum de dois usuários, com o número total de vinhos destes usuários. Este resultado é a similaridade da quantidade de usuários testados.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

A maneira proposta para gerar os dados iniciais e com isso atacar o *cold start*, foi um questionário ao usuário. É possível ver a disparidade com o objeto de estudo deste trabalho, que tem como objetivo não pedir informações ao usuário, mas se aproveitar de dados já existentes em redes sociais, tornando o processo de entrar no sistema pela primeira vez mais rápido e fácil para o usuário.

### 3.2 Recomendações baseadas na comunidade

(SAHEBI, S.; COHEN, W. William. 1997) sugerem como maneira para resolver o *cold start* utilizar outros sistemas que tenham dados do usuário para obter as preferências dele. Dado o problema do *cold start*, a solução proposta é recuperar informações deste usuário através de uma rede social, por exemplo.

Eles citam exemplos práticos: um novo usuário de um determinado site de *streaming*, onde o sistema de recomendação não conhece as preferências pessoais deste usuário. Nesse caso, o sistema obtém através de uma rede social as preferências dele e alimenta o sistema de recomendação do site de *streaming*.

Outro exemplo prático que eles abordam é chamado de rede multidimensional de detecção quando o usuário não tem preferência no mesmo gênero da recomendação, entretanto, tem preferências em outro. Exemplificando, o sistema de recomendação de filmes não encontra dados preferenciais de filmes do usuário X, mas, encontra dados preferenciais de livros e com base nesses dados ele encontra usuários com o mesmo gosto de livros e que possuem preferências em filmes, sendo assim, o sistema de recomendação oferece os filmes destes usuários ao usuário X.

Eles utilizaram 3 algoritmos para obter o cálculo dessa similaridade: similaridade de cosseno, correlação de pearson e similaridade euclideana (BREESE, J. S.; HECKERMAN, D.; KADIE, C., 1998).

Com isso, eles isolam o problema do *cold start* obtendo os dados de outra fonte. Entretanto, dado que um usuário X não tenha ligação nenhuma nos locais previstos, este

usuário continua sendo alvo do *cold start*. Este problema também ocorre no SuggestMe, tendo em vista que usuários que não curtiram nenhuma banda em rede social, continuam sendo alvo do *cold start*.

É possível ver semelhança com a solução proposta por este trabalho, pois ambos propõem usar fontes de outros sites para obter dados dos usuários. Entretanto, o SuggestMe não utiliza a filtragem colaborativa, ou seja, não realiza o cálculo de similaridade usando como parâmetros outros usuários.

### **3.3 Recomendações em Grupo**

Em sua grande maioria, os sistemas de recomendações fazem sugestões individualmente a cada usuário. É possível observar no trabalho de (QUIJANO-SÁNCHEZ, L.; BRIDGE, D. 2012) que o sistema de recomendação deles se propõe a solucionar o *cold start* para um grupo de usuários, ou seja, eles deixam de avaliar e fazer sugestões particularmente.

O objetivo é dividir usuários por grupos. Estes usuários teriam semelhanças nas suas preferências, e com isso, o sistema poderia realizar recomendações para um maior número de usuários, esta ideia é reflexo do que acontece na sociedade. Na sociedade determinados grupos são criados naturalmente através da similaridade das pessoas e este sistema de recomendação tenta simular isto.

Para resolver o *cold start* ele cadastra grupos pré-definidos que são usados como métricas de comparação. Novos usuários são selecionados para grupos que eles tenham semelhança, essa semelhança é calculada baseada primariamente em dados como: faixa etária, gênero, cidade, entre outros. Existem grupos mais bem definidos que os usuários são alocados assim que o sistema obtém preferências mais específicas dele.

Este tipo de abordagem é bem distinta da solução proposta por essa monografia, que busca evitar o *cold start*, tratando individualmente cada usuário.

## **4 IMPLEMENTAÇÃO**

Neste capítulo são apresentados detalhes do desenvolvimento do aplicativo que foi construído no decorrer do desenvolvimento deste trabalho.

## 4.1 MOTIVAÇÃO

Após o levantamento bibliográfico, foi realizado um estudo para criação do protótipo do sistema de recomendação, com o objetivo de fazer recomendações de bandas e/ou artistas musicais ao usuário utilizador. Todas as recomendações têm como base as preferências pessoais dele.

Um outro ponto importante do sistema de recomendação é a sua estratégia para resolver o problema do *cold start*, dado que o sistema não tinha dados para realizar a recomendação. A solução foi utilizar uma grande base de dados colaborativa denominada Freebase, que possui dados das mais variadas fontes e dispõe de bastantes informações. Especificamente para este protótipo foram utilizadas apenas as informações de bandas e artistas.

A criação deste protótipo ocorreu com êxito e a seguir será apresentado o seu desenvolvimento.

## 4.2 DIAGRAMA DE CASOS DE USO

Os diagramas de casos de uso, descrevem funcionalidades e processos que um novo projeto terá e geralmente são criados após o levantamento dos requisitos funcionais.

Como forma de contextualizar as funcionalidades e a utilização deste sistema de recomendação, foi criado o diagrama de caso de uso, na Figura 8.

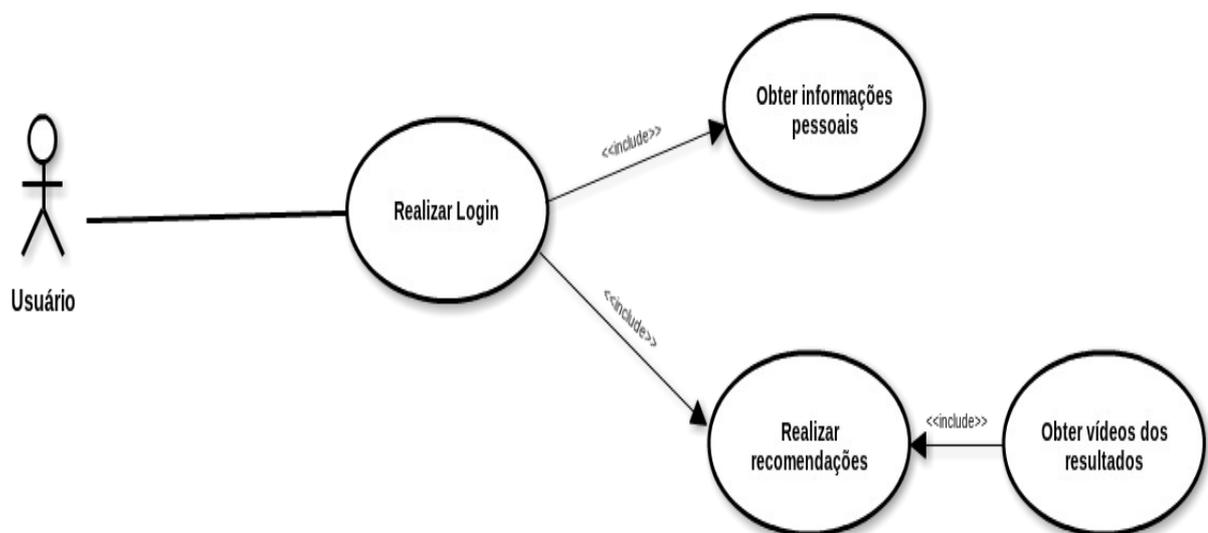


Figura 8 – Diagrama de casos de uso da utilização do sistema

O sistema consiste basicamente nestas funcionalidades: o usuário realiza o login, automaticamente o sistema obtém os dados necessários para recomendação. e com isto realiza as recomendações e as sugestões de vídeos.

### **4.3 DESENVOLVIMENTO**

O desenvolvimento do sistema consistiu em:

1. Criação da arquitetura do sistema
2. Obtenção dos dados do usuário.
3. Cálculo de similaridade das bandas para sugestão ao usuário.
4. Melhoria na interface do sistema.
5. Exibição dos resultados ao usuário.
6. Persistência de todos os dados gerados pelo sistema.

A linguagem utilizada no *backend* do sistema de recomendação foi o PHP, a escolha foi justificada pela simplicidade e facilidade na integração com as outras tecnologias utilizadas no protótipo.

Em posse das preferências dos usuários, obtidas através da rede social Facebook, é realizado o cálculo de similaridade das bandas.

#### **4.3.1 CÁLCULO DE SIMILARIDADE**

Para realizar o cálculo de similaridade, o sistema primeiramente precisa obter as bandas e/ou artistas que o usuário curtiu no Facebook. Para isto, foi utilizada a API do Facebook, ela solicita a autenticação do usuário na rede social e a liberação do aplicativo SuggestMe. Feito isto, ele retorna ao sistema todas as informações públicas do usuário, inclusive sua lista de bandas/artistas curtidas.

Com as informações obtidas do Facebook, o sistema realiza consultas utilizando a API do Freebase para descobrir os gêneros das bandas/artistas curtidas pelo usuário. Por exemplo, ao buscar os gêneros da banda Nirvana, são retornados os gêneros: Punk Rock, Grunge, Alternative Rock, Rock Music, Hardcore Punk, Progressive Rock, Psychedelic Rock,

Symphonic rock, Power pop, Soft rock, Baroque pop, como mostrado na Figura 3. Esta busca é realizada para cada banda que o usuário curtiu, individualmente.

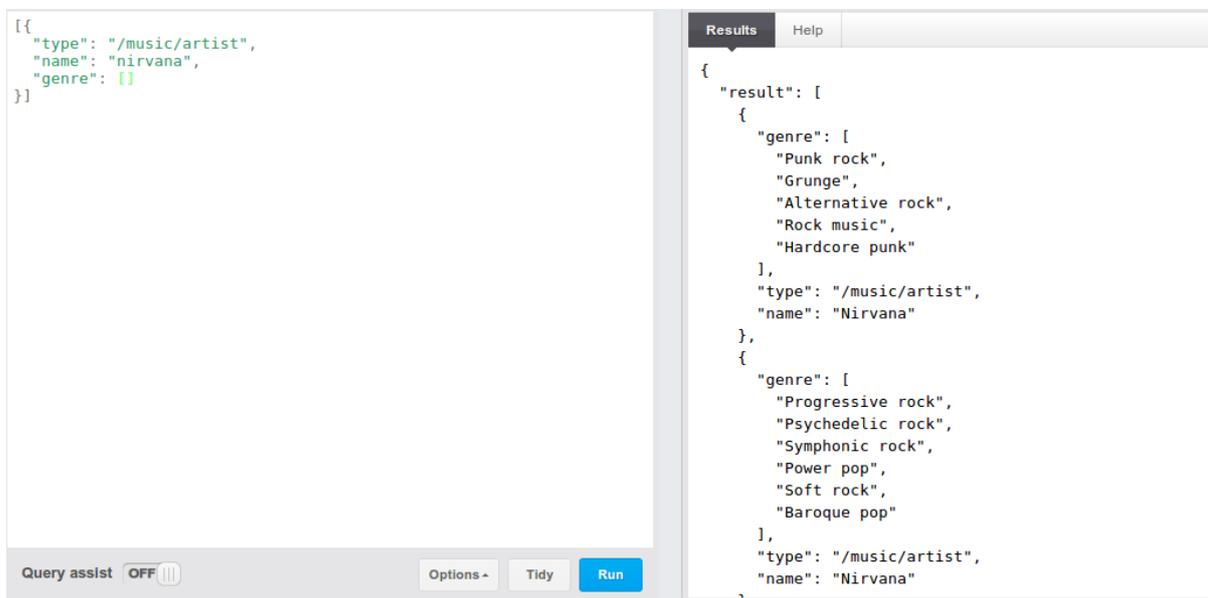


Figura 3 – Resultado da busca pelos gêneros da banda nirvana.

O sistema cria uma matriz de comparação que guarda gênero e valor, sendo cada gênero adicionado à matriz de comparação com o valor da *flag* 1, caso não exista o gênero na matriz ainda, do contrário é incrementado 1 no campo flag. O campo flag é do tipo inteiro no banco de dados.

O sistema realiza a busca de todos os gêneros para todas as bandas. Exemplo de matriz genérica na tabela 1.

Gênero	Flag
Punk Rock	1
Alternative Rock	2
Funk	1
Hard Rock	4
Reggae	3
Trance	4

Tabela 1 – Matriz de comparação de gêneros musicais

Ao término do preenchimento da matriz de comparação, é feita uma ordenação pelo campo *flag*, ou seja, ordena-se de forma decrescente para trazer os gêneros que mais se repetiram, como na Tabela 2.

Gênero	Flag
Hard Rock	4
Trance	4
Reggae	3
Alternative Rock	2
Funk	1
Punk Rock	1

Tabela 2 – Matriz de comparação ordenada

Com estes dados, é possível saber quais são os gêneros que o usuário tem mais afinidade, dado que uma banda tem inúmeros gêneros e em alguns casos um gênero específico de uma banda pode não corresponder com o gosto musical do usuário, e recomendar através desse gênero poderia gerar uma má recomendação. Por isso, foi criada esta fórmula de encontrar os gêneros que o usuário tem mais afinidade. O sistema de recomendação seleciona os *n* gêneros que têm o maior valor na coluna *flag*, o valor de *n* é igual ao número de bandas curtidas pelo usuário. Um exemplo disto: um usuário tem 4 bandas curtidas e estas bandas geram 30 gêneros na matriz, ele consideraria apenas os 4 gêneros com maior número na coluna *flag* da matriz de comparação. O número 4 foi escolhido de forma aleatória, é possível alterar para fazer recomendações utilizando apenas o principal resultado, caso altere este número para 1, por exemplo. Os resultados obtidos dos gêneros de maior afinidade do usuário são inseridos em um vetor de gêneros.

O próximo passo é buscar outras bandas com base no vetor de gêneros que foi formado nos cálculos anteriores. Novamente são feitas buscas utilizando a API da Freebase passando como parâmetros individualmente cada gênero do vetor. Um exemplo desta busca está na figura 4.

```
[[{"type": "/music/artist",
  "name": [],
  "genre": "reggae"}]]
```

```
Results Help
{
  "result": [
    {
      "name": [
        "The Police"
      ],
      "genre": "Reggae",
      "type": "/music/artist"
    },
    {
      "name": [
        "Solabeat Alliance"
      ],
      "genre": "Reggae",
      "type": "/music/artist"
    },
    {
      "name": [
        "Sounds Like Chicken"
      ],
      "genre": "Reggae",
      "type": "/music/artist"
    },
    {
      "name": [

```

Query assist OFF  Options ^ Tidy Run

Figura 4 – Busca de Bandas pelo gênero Reggae

Como o conjunto de resultados é grande, é selecionado apenas 1 resultado de cada gênero. Todos os resultados destas consultas são salvos em outro vetor chamado de vetor resultado, que é utilizado para mostrar os resultados finais ao usuário.

Por fim, é sugerido os vídeos mais assistidos de cada banda e/ou artista com a API do YouTube, após isso, todos os dados do usuário são salvos para futuras utilizações, entre eles: nome do usuário, idfacebook, bandas curtidas, bandas sugeridas, localização, entre outros. Como mostrado na figura 5.

```
mysql> select * from t_usuario; select * from t_bandascurtidas; select * from t_bandassugeridas;
```

codigo	nome	idfacebook	sexo	localizacao
0	Danilo Formiga	100001019334126	male	João Pessoa, Brazil

1 row in set (0.00 sec)

codigo	nomeBanda	idfacebook
1	Red Hot Chili Peppers	100001019334126
2	Rodolfo Abrantes Oficial FP	100001019334126
3	Creed	100001019334126

3 rows in set (0.00 sec)

codigo	nomeBanda	idfacebook	resultado
13	Wilco	100001019334126	
14	Beastie Boys	100001019334126	
15	GG Allin	100001019334126	
16	Steve Miller Band	100001019334126	

4 rows in set (0.00 sec)

```
mysql>
```

Figura 5 – Resultados persistidos no banco de dados

## 4.4 TECNOLOGIAS UTILIZADAS

Para criação deste protótipo, foram utilizados: a linguagem de programação PHP, a API do Facebook para obter os dados pessoais do usuário que serão utilizados para realizar a recomendação, jQuery e Bootstrap para melhor apresentação da interface gráfica ao usuário, além da API do YouTube para realizar sugestões de vídeos.

### 4.4.1 PHP

PHP (<http://php.net/>) é uma linguagem interpretada livre que surgiu em 1995, criada por Rasmus Lerdof. Seu principal objetivo é o desenvolvimento de aplicações web de forma dinâmica no lado do servidor. É bastante utilizada em todo o mundo, tendo como principais características: ser orientada a objetos, mas permitir o uso procedural, ter tipagem fraca e dinâmica e ser multiplataforma.

#### 4.4.2 API FACEBOOK

Surgiu com a necessidade de haver integração das redes sociais com toda a web. Atualmente vários sites na internet através dessa API (<https://developers.facebook.com/>) permitem aos usuários realizarem login utilizando os seus dados do Facebook.

Esta API fornece muitas configurações e funcionalidades para acessar as informações contidas no Facebook. Além de recuperar diversas informações do usuário, podemos ainda: publicar pelo perfil, obter as páginas curtidas, entre várias outras opções.

#### 4.4.3 JQUERY

jQuery (<http://jquery.com/>) é uma biblioteca de JavaScript, que foi desenvolvida para facilitar o uso de JavaScript para os desenvolvedores e trazer mais facilidade para interagir com o HTML. Foi lançada em 2006 por John Resig. É a biblioteca JavaScript mais utilizada hoje no mundo, conta a marca de 77,8% de uso nos 10 mil sites mais visitados, segundo as estatísticas do site <http://trends.builtwith.com/javascript/jquery>.

#### 4.4.4 BOOTSTRAP

O Bootstrap é um framework para desenvolvimento *front-end*, que tem como objetivo facilitar o desenvolvimento de páginas html ao oferecer componentes pré-definidos para utilização. O *framework* foi criado por Mark Otto e Jacob Thornton da equipe do *Twitter*.

O trabalho do desenvolvimento *front-end* de aplicações é algo que demanda tempo e recursos humanos, muitas vezes para realização de tarefas básicas. A solução proposta pelo *Bootstrap* é retirar este trabalho dos desenvolvedores, ao utilizar um *framework* eles abstraem algumas tarefas que teriam que ser realizadas para montar o layout.

A utilização do Bootstrap é bastante simples, basta realizar o download em (<http://getbootstrap.com/>) e adicionar no HEAD do HTML os links para os arquivos do bootstrap.

```
<html>
<head>
.....
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="plugins/bootstrap/bootstrap.css" rel="stylesheet">
```

Figura 6 – Importação do Bootstrap

É possível ver na Figura 6 a importação do arquivo do Bootstrap, após isto, é possível chamar os componentes do Bootstrap e fazer a utilização deles.

#### 4.4.5 API DO YOUTUBE

A API do YouTube (<https://developers.google.com/youtube/>) permite a integração do conteúdo e funcionalidades do YouTube a um determinado site, aplicativo ou dispositivo.

O YouTube disponibiliza duas API's: API do Player e API de dados.

O objetivo da *API* de dados é disponibilizar pesquisa de vídeos, recuperar *feeds* de canais, enviar vídeo, ver conteúdos similares, entre outros. Enquanto, a API do player visa oferecer controle da execução dos vídeos do YouTube. Alguns dos controle do player são: tamanho do player, tipo do player, entre outros.

Neste trabalho, foram utilizadas as duas API's, A API de dados foi utilizada para recuperar informações das bandas sugeridas e a API do player foi utilizada para exibir os vídeos recuperados pela API de dados.

#### 4.4.6 FREEBASE

O *Freebase* (<http://freebase.com>) é uma base de dados colaborativa que possui várias formas de obtenção de dados da web, totalmente livre, criado pela empresa *Metaweb* e liberada ao público em março de 2007 (YAO, X.; BERANT, J.; DURME, V. B. 2014). Tem como objetivo principal criar um local de informações de fácil acesso às pessoas e máquinas.

As consultas realizadas na API da *Freebase* utilizam um tipo de linguagem específica, denominada *Metaweb Query Language (MQL)*, que foi utilizada no sistema de recomendação para obtenção dos resultados da consulta.

#### 4.4.6.1 METAWEB QUERY LANGUAGE (MQL)

Este nome faz referência ao nome da empresa que criou o *Freebase* chamada *Metaweb*. É uma linguagem criada para utilizar os serviços do *Freebase*, consiste basicamente em comunicação com o *Freebase* através de arquivos *JSON(JavaScript Object Notation)*.

Um exemplo de utilização: Deseja-se obter os álbuns de uma banda chamada “The Police”, usando o MQL para consulta, o resultado está na figura 7.

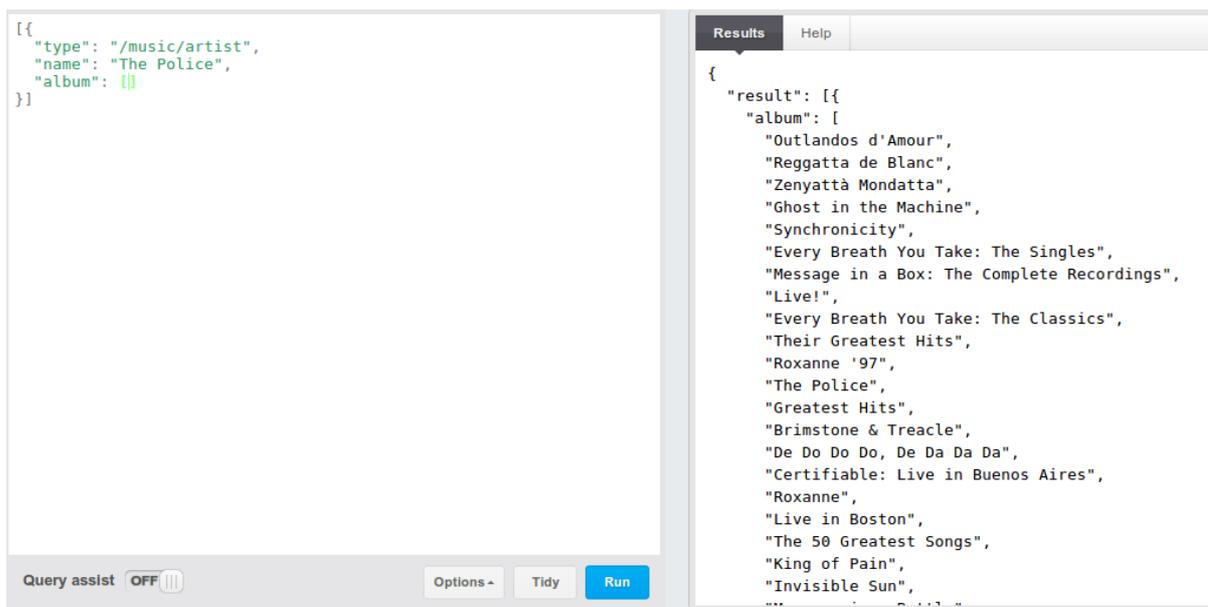


Figura 7 – Resultado da busca de álbuns da banda The Police

## 4.5 TELAS

Nesta seção, serão exibidas imagens do protótipo criado neste projeto. São telas iniciais e resultados obtidos com a utilização do protótipo.

# Receba **recomendações musicais!**

Sistema de recomendação musicais



ENTRE E FAÇA UM TESTE :) ↗

Figura 9 – Tela de Login do Sistema

Está é a tela inicial ao usuário acessar o sistema, nela é possível ler informações como:

- O que é o sistema?
- Como Funciona?
- Depoimentos.

O sistema traz apenas um botão para o usuário começar a obter as recomendações. O botão é denominado: “Entre e faça um teste”.

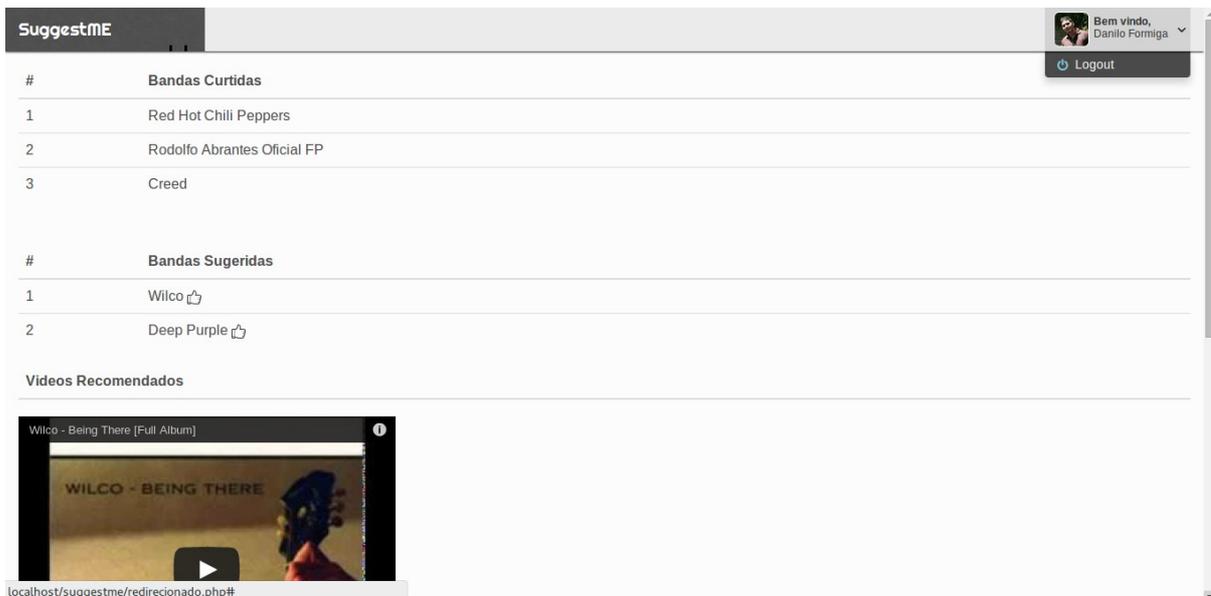


Figura 10 – Tela principal do sistema

Nesta tela é possível ver as recomendações feitas ao usuário que curtiu as bandas: Red Hot Chili Peppers, Rodolfo Abrantes Oficial FP, Creed.

As recomendações sugeridas são: Wilco, Deep Purple. É possível visualizar logo abaixo, os vídeos com mais acessos das bandas sugeridas.

É possível ver a utilização do protótipo em outro dispositivo além do computador, na figura 12 e 13 é possível ver a utilização em um *smartphone*. Provando assim, que o sistema é responsivo na sua utilização.



Figura 12 – Bandas curtidas do usuário Elmano

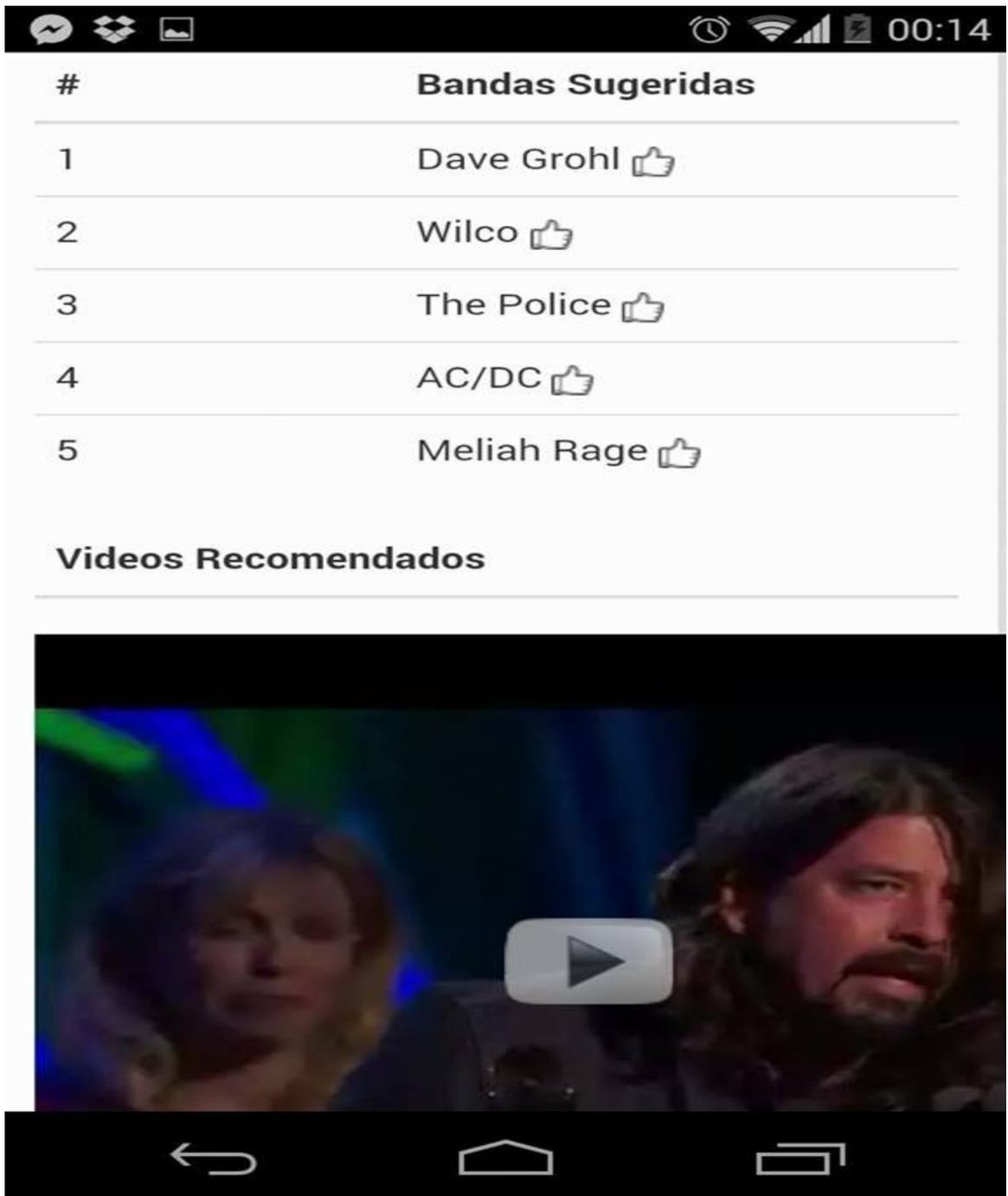


Figura 13 – Recomendação do usuário Elmano Neto

## 5 RESULTADOS

Nesta seção, serão exibidos resultados de usuários que utilizaram o sistema e enviaram as recomendações que receberam. Estas informações estão presentes no banco de dados, entretanto, as tabelas facilitam a interpretação dos resultados.

Na tabela 4, é possível ver as recomendações feitas ao usuário Elmano, com base nas bandas curtidas dele.

<b>No.</b>	<b>Bandas Curtidas</b>	<b>Bandas Sugeridas</b>
1	Grupo Fora de Área	Dave Grohl
2	Bezerra da Silva	Wilco
3	Velhas Virgens	The Police
4	Mastruz com Leite	AC/DC
5	Pearl Jam	Melian Rage
6	Nightswish	
7	Aviões do Forró	

Tabela 4 - Bandas Curtidas e Sugeridas ao usuário Elmano Neto

Na recomendação da tabela 5, o protótipo realizou o cálculo de similaridade baseado nas bandas que o usuário Fernando curtiu e fez as recomendações com base nisto.

<b>No.</b>	<b>Bandas Curtidas</b>	<b>Bandas Sugeridas</b>
1	Orquestra Sinfônica UFPB	Steve Miller Band
2	Pink Floyd	The Raspberries
3	Maria Rita	Hawkwind
4	Chico Buarque	Moonlight
5	Val Donato	No Doctors
6	Frédéric Chopin	
7	Antônio Carlos Jobim	
8	Vinicius de Moraes	
9	Seu Jorge	
10	The Beatles	
11	Lenine	
12	Os Gonzagas	

13	Adriana Calcanhoto	
14	Maria Rita	
15	Gilberto Gil	
16	Tom Jobin	

Tabela 5 - Bandas Curtidas e Sugeridas ao usuário Fernando Brito

Como o usuário Gilmar Junio, possui bastante bandas e/ou artistas brasileiros, o Freebase encontrou poucas informações destas bandas. Entretanto, ainda foi realizada uma recomendação de uma banda brasileira. Uma possível solução para este problema é utilizar outras redes sociais que contenham mais dados de bandas e/ou artistas brasileiros.

<b>No.</b>	<b>Bandas Curtidas</b>	<b>Bandas Sugeridas</b>
1	Gabi Luthai	Wilco
2	Banda Malta Oficial	Chiquinha Gonzaga
3	Jamz	Dragon Ash
4	Juan Varriento	Steve Miller Band
5	Marcelo Falcão	The Police
6	Hamilton de Holanda	
7	Manzinha e seu teclado	
8	Claudia Leitte	
9	Daniel	
10	Lulu Santos	
11	Carlinhos Brown	
12	Ramon Schnaydern	
13	Nosso Jeito	
14	Thiaguinho	
15	Davidson Silva	

Tabela 6 - Bandas Curtidas e Sugeridas ao usuário Gilmar Junio

Na tabela 7, é possível observar as recomendações realizadas a usuária Nathalia. É importante perceber que como a usuária Nathalia tem várias bandas e/ou artistas do gênero samba e pagode, a matriz de comparação de gêneros pontuou o máximo nos tipos de gênero samba e pagode e com isso foi realizada a recomendação.

<b>No.</b>	<b>Bandas Curtidas</b>	<b>Bandas Sugeridas</b>
1	Só Pra Contrariar	Zeca Pagodinho
2	Jorge e Mateus	Steve Miller Band
3	China CDs	Cypress Hill
4	E.Mind	Dragon Ash
5	Sorriso Maroto - “Trechos Musicais”	
6	Aviões do Forró	
7	Samba da Elite	
8	Alexandre Pires	
9	Roberto Carlos (Singer)	
10	Marcelo Rossi	
11	Exaltasamba	

Tabela 7 - Bandas Curtidas e Sugeridas ao usuário Nathalia Dantas

Na tabela 8 é demonstrado um problema, quando o usuário curtiu bandas que não existem no freebase o sistema não consegue realizar o cálculo da similaridade, este caso é mais frequente com bandas brasileiras pouco conhecidas no mercado.

<b>No.</b>	<b>Bandas Curtidas</b>	<b>Bandas Sugeridas</b>
1	Pedro Henrique & Fernando OFICIAL	
2	Vera Lúcia cantora católica	

Tabela 8 – Bandas Curtidas e Sugeridas ao usuário Wennio Kelson

O sistema não consegue realizar o cálculo de similaridade, pois não consegue encontrar no Freebase informações sobre as bandas “Pedro Henrique & Fernando OFICIAL” e “Vera Lúcia cantora católica” sendo assim, ele não consegue realizar recomendações.

## 6 CONCLUSÃO

Este trabalho teve como objetivo principal o desenvolvimento de um protótipo de sistema de recomendação, utilizando web semântica para evitar o *cold start*. Foram utilizadas diversas abordagens para que o usuário não precisasse informar nenhum dado para obter recomendações musicais, apenas fornecer acesso aos dados do seu perfil do Facebook.

Os problemas encontrados no *cold start* estão entre os principais desafios dos novos sistemas de recomendação, e a contribuição deste trabalho foi trazer uma nova perspectiva para a solução deste problema. Uma vez que o sistema de recomendação não tenha dados para recomendar ao usuário, ele poderia obter de uma base colaborativa.

A sugestão proposta por este trabalho dá margem para varias abordagens futuras, como por exemplo, utilizar o *Freebase* enquanto não houver dados significativos para gerar recomendações, e após conseguir gerar dados suficientes, abandonar o uso do *Freebase* e gerar as recomendações autonomamente.

Neste protótipo a ideia é não abandonar o *Freebase*, pois o sistema de recomendação não precisa se preocupar com atualização de dados, ou com a infraestrutura desse banco de dados, ele apenas consome o serviço que é oferecido por terceiros.

O *Freebase* é alimentado por um grande número de usuários. Por ele ser colaborativo, o crescimento das informações de bandas e/ou artistas presentes nele deixa as recomendações mais dinâmicas, tendo em vista que novas bandas que não eram avaliadas em recomendações podem ter seus dados inseridos e serem avaliadas em outra utilização do sistema.

É importante frisar que este trabalho não fica preso apenas em recomendações de bandas. Com algumas alterações é possível fazer esse sistema recomendar diversos itens ao usuário, tendo em vista que o *Freebase* contém informações dos mais variados itens. Por exemplo: livros e filmes, é possível descobrir através do Facebook quais os livros ou filmes que o usuário curtiu ou leu/viu e com isso, realizar sugestões ao usuário.

## 6.1 DIFICULDADES

Durante o desenvolvimento do trabalho foram encontradas dificuldades que ainda precisam ser resolvidas. Por exemplo, a *Freebase* contém poucos dados de bandas e/ou artistas brasileiros. Com isso, bandas e/ou artistas pouco conhecidos não puderam ser avaliados pelo projeto, não conseguindo gerar assim recomendações para os usuários. É possível ver um exemplo disto na Figura 11, utilizando a banda Detonautas (banda de rock brasileira).

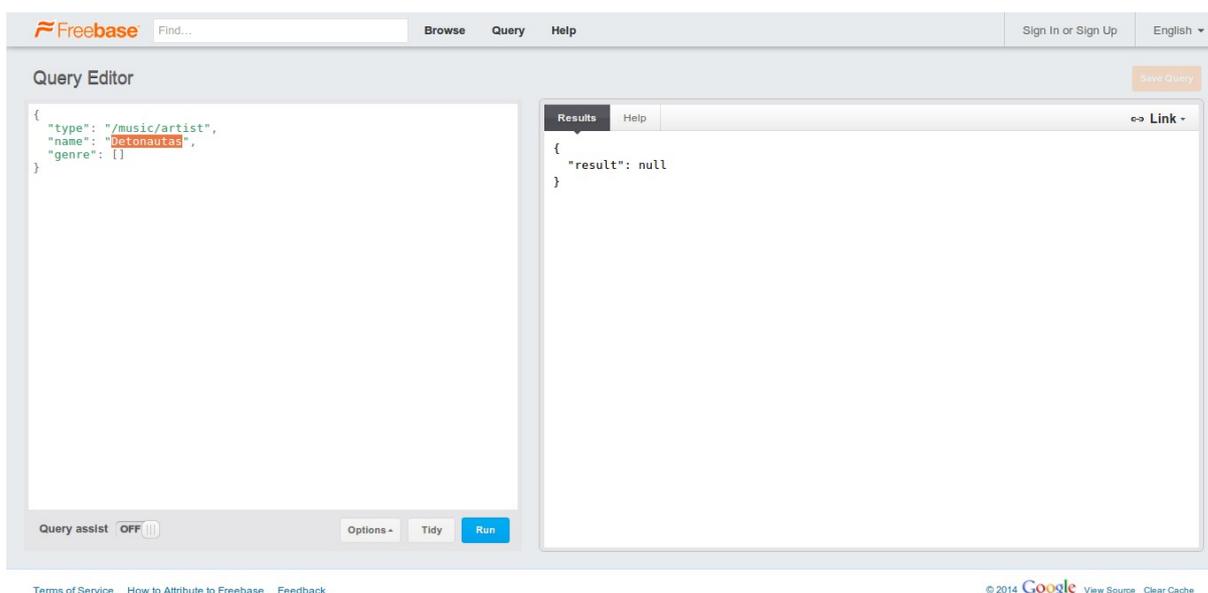


Figura 11 – Demonstração de uma banda que não existe no *Freebase*

Outra dificuldade encontrada, é caso o usuário não tenha nenhuma banda curtida na rede social Facebook, ele será vítima do *cold start* tendo em vista que não será possível efetuar o cálculo da similaridade.

## 6.2 SUGESTÕES DE TRABALHOS FUTUROS

Para trabalhos futuros, é indicado que o cálculo da similaridade seja melhorado, utilizando a similaridade do cosseno e incluindo variáveis como dados de outros usuários, amigos na rede social.

Uma maneira de influenciar o cálculo de similaridade utilizando como variável os amigos, consiste em obter os amigos que curtem bandas e/ou artistas semelhantes ao perfil do

usuário, e verificar bandas que este amigo curtiu que o usuário utilizador ainda não curtiu. Este é um exemplo de como os amigos do usuário influenciariam no cálculo de similaridade.

Além disso, o sistema poderia ainda obter dados de outras redes sociais como o *Spotify*, que é um aplicativo de reprodução de músicas por *streaming*. Uma melhoria significativa no cálculo de similaridade seria obter os artistas e/ou bandas que o usuário escuta através deste aplicativo. Ainda existem outras redes sociais que contém o mesmo tipo de informação.

Uma outra sugestão, é colocar um botão de curtir nas sugestões realizadas pelo sistema, com isso, seria possível refinar as recomendações feitas aos usuários. Se os usuários curtissem as recomendações, poderia ser gerado um ranking que seria usado como variável no cálculo de similaridade.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ADOMAVICIUS, G.; TUZHILIN, A. **Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.** Knowledge and Data Engineering, IEEE Transactions on, v. 17, n. 6, p. 734-749, 2005.
- BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval.** Addison-Wesley, 1999.
- BALABANOVIĆ, M.; SHOHAM, Y. **Fab: content-based, collaborative recommendation.** Communications of the ACM, v. 40, n. 3, p. 66-72, 1997.
- BERNERS. T.B.; HENDLER; J.; LASSILA, O. **The Semantic Web.** Scientific American, maio 2001. Disponível em: <<http://www.cs.umd.edu/~golbeck/LBSC690/SemanticWeb.html>> Acesso em: 17 junho de 2014.
- BOOCH, G; RUMBAUGH, J; JACOBSON, I. **UML: guia do usuário.** Rio de Janeiro: Elsevier, 2005. 474 p.
- BREESE, J. S.; HECKERMAN, D.; KADIE, C. **“Empirical Analysis of Predictive Algorithms for Collaborative Filtering,”** Proc. 14th Conf. Uncertainty in Artificial Intelligence, July 1998.
- BREITMAN, K. **Web Semântica: a Internet do futuro.** Rio de Janeiro: LTC, 2005.
- CLAYPOOL, M. et al. **Combining content-based and collaborative filters in an online newspaper.** Proceedings of ACM SIGIR workshop on recommender systems, 1999. Citeseer.
- DANTAS, S. B. A. **Sistemas de Recomendação para clientes de vídeo locadoras baseado em redes SOM.** 2009. 48 f. Trabalho de Conclusão de Curso (Engenharia da Computação) – Departamento de Sistemas e Computação, Escola Politecnica de Pernambuco, Pernambuco, 2009.
- DELGADO, J.; ISHII, N. **“Memory-Based Weighted-Majority Prediction for Recommender Systems,”** Proc. ACM SIGIR ’99 Workshop Recommender Systems: Algorithms and Evaluation, 1999.
- FILHO, F. M. F.; GEUS, L. P.; ALBUQUERQUE. P. J. **Sistemas de Recomendação e Interação na Web Social.** Disponível em: [http://www.academia.edu/download/30832515/websocial\\_ihc08.pdf](http://www.academia.edu/download/30832515/websocial_ihc08.pdf) Acesso em: 03 de Julho de 2014.
- JANNACH, D. **Finding preferred query relaxations in content-based recommenders.** In: 3rd International IEEE Conference on Intelligent Systems, pp. 355–360 (2006).
- MAHAPATRA, S.; TAREEN, A.; YANG, Y. **A Cold Start Recommendation System Using Item Correlation and User Similarity.** 2002. ACM. New York.

- MAHMOOD, T.; RICCI, F. **Improving recommender systems with adaptive conversational strategies**. In: C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp. 73–82. ACM (2009)
- MEDEIROS, G. R. I. **Estudo Sobre Sistemas de Recomendação Colaborativo**. 2013. 42 f. Trabalho de Conclusão de Curso (Graduação em Ciências da Computação) – Centro de Informatica, Universidade Federal de Pernambuco, Pernambuco. 2013
- NAIK, U.; SHIVALINGAIAH, D. **Comparative Study of Web 1.0, Web 2.0 and Web 3.0.**, 2008. Disponível em: <http://ir.inflibnet.ac.in/bitstream/1944/1285/1/54.pdf> Acesso em: 19 de junho de 2014.
- QUIJANO-SÁNCHEZ, Lara; BRIDGE, Derek; DÍAZ-AGUDO, Belén; RECIO-GARCÍA A. Juan. **A Case-Based to the Cold-Start Problem in Group Recommenders**. ICCBR. 2012.
- RESNICK, P. VARIAN, H. **Recommender Systems**. Disponível em: <http://www.inf.unibz.it/~ricci/ISR/papers/resnick-varian97.pdf>, 1997. Acesso em 11/06/2014.
- RICCI, F.; ROKACH, L.; SHAPIRA B. **Introduction to Recommender Systems Handbook**, 2011. Disponível em: <http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf>. Acesso em: 15/06/2014.
- SAHEBI, Shaghayegh; Cohen, W. William. **Community-based Recommendations: a Solution to the Cold Start Problem**. 1997. ACM. Texas.
- SARWAR, B. et al. **Item-based collaborative filtering recommendation algorithms**. Proceedings of the 10th international conference on World Wide Web, 2001. ACM. p.285-295
- SOUZA, R. R.; ALVARENGA, L. **A Web Semântica e suas contribuições para a ciência da informação**. *Ci. Inf.*, Brasília, v. 33, n. 1, p. 132-141, jan./abr. 2004.
- STUART, E.; MIDDLETON, H. A; DAVID, C. R. **Exploiting Synergy Between Ontologies and Recommend Systems**, 2002. Disponível em: <http://arxiv.org/pdf/cs.LG/0204012.pdf> Acesso em: 11/06/2014.
- SZOMSZOR, M. et al. **Folksonomies, the Semantic Web, and Movie Recommendation**. 2007. Disponível em: <http://eprints.soton.ac.uk/264007/>. Acesso em: 11/06/2014
- YAO, X.; BERANT, J.; DURME, V. B. **Freebase QA: Information Extraction or Semantic Parsing?**. Disponível em: <http://www.allenai.org/Content/Publications/ie-vs-sp.pdf>. 2014. Acesso em: 15 de agosto de 2014.