



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA
PARAÍBA**

Coordenação do Curso Superior de Tecnologia em Sistemas para Internet

RELATÓRIO DE ESTÁGIO

Identificando o Contexto do Usuário através do Facebook

Diogo de Alcântara Rodrigues Alves

João Pessoa – PB
Outubro/2012

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Unidade Acadêmica de Informação e Comunicação
Coordenação do CST em Sistemas para Internet

Identificando o Contexto do Usuário através do Facebook

Diogo de Alcântara Rodrigues Alves

Relatório de Estágio Técnico apresentado à disciplina Estágio Supervisionado da Coordenação do Curso Superior em Sistemas para Internet do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba como requisito parcial para obtenção do grau de Tecnólogo em Sistemas para Internet.

Orientador: Profa. Dra. Damires Yluska de Souza Fernandes

Supervisor: Profa. Dra. Damires Yluska de Souza Fernandes

Coordenador do Curso de Sistemas para Internet: Thiago José Marques Moura

Empresa: IFPB

Período: 01/03/2012 a 30/06/2012

Prof.º DAMIRES YLUSKA DE SOUZA FERNANDES, DSc.

Orientador

Prof.º DAMIRES YLUSKA DE SOUZA FERNANDES, DSc.

Supervisor de Projeto

Diogo de Alcântara Rodrigues Alves

Aluno de Sistemas para Internet

“O uso das redes sociais tem aumentado nos últimos tempos: evidências estatísticas indicam que não só existem mais pessoas participando dessas comunidades, mas há um aumento na quantidade média de tempo gasto.”

The Nielsen Company (Whitepaper). Global faces and networked places: A nielsen report on social networking's new global footprint, March 2009, tradução.

AGRADECIMENTOS

Agradeço a todos que direta ou indiretamente contribuíram na conclusão desse trabalho, em especial a professora Damires Souza que me ofereceu a oportunidade de participar em seus projetos de pesquisa e compartilhou um pouco de seus conhecimentos estimulando sempre a atividade de pesquisa e o compartilhamento de informações entre todas as equipes.

Aos colegas de curso que fazem parte do projeto *CODI4in*, Jimmy, Marcelo, Thiago e Lívio.

Agradeço a toda equipe da Coordenação e Professores do Curso de Tecnologia em Sistemas para Internet.

RESUMO

A especificação e implementação da extensão ao *plugin CODI4In* para captura e identificação do contexto de usuários através do Facebook é o tema deste relatório de pesquisa. Será apresentada a identificação implícita do contexto do usuário utilizando seus dados capturados a partir do Facebook, e seus gêneros preferidos através do *IMDB Webservice*. Essas informações são persistidas e gerenciadas pelo *CODI4In* e são utilizadas para prover personalização de consultas na aplicação *MovieShow* – uma aplicação Web de consulta a filmes povoada com dados extraídos do *IMDB*. A personalização ocorre por meio do ranqueamento dos resultados obtidos em cada consulta. Serão mostrados também os experimentos realizados com usuários, cujos resultados demonstraram que a abordagem implementada é promissora e já traz uma significativa diferença para o usuário.

Palavras-chave: Contexto, Personalização de Consultas, Perfil do usuário, Facebook.

LISTA DE FIGURAS

Figura 1. Personalização e Customização [Eynard <i>et al.</i> 2008].....	5
Figura 2. Login com aplicativo e autorização	10
Figura 3. Fragmento da ontologia de contexto CODI-User	14
Figura 4. Exemplo de instância da CODI-User.....	14
Figura 5. Arquitetura do CODI4In	15
Figura 6. Diagrama de casos de uso do <i>CODI4In</i> acoplado ao MovieShow	16
Figura 7. Interface do MovieShow	17
Figura 8. Consulta submetida pelos Usuários Ben e Ana	18
Figura 9. Resultados personalizados da consulta submetida pelo Ben	19
Figura 10. Resultados personalizados da consulta submetida pela Ana	19
Figura 11. Login do usuário	21
Figura 12. Consulta por filmes no MovieShow	23
Figura 13. Login e autorização do CODI4In no Facebook	25
Figura 14. Retorno JSON das informações básicas e filmes do usuário.....	26
Figura 15. Página configurada como Filme	26
Figura 16. Retorno JSON do IMDB Webservice.....	27
Figura 17. Algoritmo de captura do contexto do usuário.....	28
Figura 18. Tela inicial da aplicação	29
Figura 19. Filmes e gêneros capturados	29
Figura 20. Gêneros preferidos capturados.....	30
Figura 21. Resultado da consulta sem o <i>CODI4In</i>	31
Figura 22. Resultado da consulta com o <i>CODI4In</i>	31
Figura 23. Resposta à pergunta: Frequência em que curte filmes no Facebook	32
Figura 24. Resposta à pergunta: O uso do <i>plugin CODI4In</i> melhorou o resultado da consulta.....	33

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
IMDB	<i>Internet Movie Database</i>
JSON	<i>JavaScript Object Notation</i>
OWL	<i>Web Ontology Language</i>
PMDS	<i>Peer Data Management System</i>
SDK	<i>Software Development Kit</i>
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

1. Introdução	1
1.1. Objetivos.....	1
1.2. Objetivos específicos.....	1
1.3. O Projeto.....	1
1.4. Organização do Relatório	3
2. Estado da Arte	4
2.1. Personalização de Consultas.....	4
2.2. Contexto.....	6
2.3. Redes Sociais.....	7
2.3.1. Facebook	9
2.4. Trabalhos Relacionados.....	10
3. Abordagem CODI4In	13
3.1. Conceitualizando o CODI4In	13
3.2. A Ontologia de Contexto CODI-User	13
3.3. Arquitetura do CODI4In.....	14
3.4. O Plugin.....	15
3.5. O MovieShow.....	16
3.6. O CODI4In e o MovieShow na Prática	16
4. Extensão ao CODI4In: Implementação e Resultados	20
4.1. Especificação	20
4.2. Implementação.....	23
4.2.1. Facebook API.....	23
4.2.2. IMDB API.....	27
4.2.3. O Algoritmo	27
4.3. Exemplo.....	28
4.4. Testes	32
5. Considerações Finais	34
5.1. Contribuições.....	34
5.2. Dificuldades Encontradas	34
5.3. Trabalhos Futuros	35
Referências	36
Apêndice	39

1. Introdução

Este trabalho é parte integrante de um projeto de Iniciação Científica ora em desenvolvimento no IFPB campus João Pessoa. O referido projeto tem como objetivo o desenvolvimento de um *plugin* denominado *CODI4In* que captura e gerencia informações contextuais do usuário. Neste panorama, este relatório apresenta as atividades realizadas no desenvolvimento de uma parte deste *plugin* que identifica o contexto do usuário a partir de uma rede social. O presente capítulo descreve os objetivos do projeto, introduz as atividades desenvolvidas e a organização de todo o relatório.

1.1.Objetivos

Este trabalho tem como objetivo a identificação e extração de dados referentes ao contexto do usuário a partir da rede social Facebook. Visa também a implementação deste requisito no *plugin CODI4In* existente.

1.2.Objetivos Específicos

Os objetivos específicos do presente trabalho foram:

- Familiarização com o conceito de Contexto e Personalização de Consultas;
- Estudo da abordagem do *plugin CODI4In*;
- Estudo sobre a obtenção de elementos contextuais dos usuários a partir de redes sociais;
- Estudo da API do Facebook;
- Estudo da API do IMDB;
- Especificação, implementação e validação da extensão do *plugin*;
- Análise dos resultados obtidos.

1.3.O Projeto

A busca por informações nos dias de hoje, estejam elas em um banco de dados local ou distribuídas em fontes diversas, tornou-se uma tarefa cada vez mais comum. Entretanto, o aglomerado de dados existentes dificulta esta busca e maximiza as chances de inexatidão diante das consultas que são normalmente executadas sem considerar o perfil do usuário. Para minimizar este problema, a estratégia de personalização da consulta vem sendo estudada e implementada em diferentes ambientes e soluções [Arruda *et al.* 2010, Kostadinov *et al.* 2007]. A ideia por trás da personalização da consulta é que diversos usuários, ao realizar uma consulta,

possam obter resultados diferentes como resposta, de acordo com um modelo de usuário identificado [Koutrika e Ioannidis 2005].

Segundo [Dey 2001], *contexto* se refere a qualquer informação que caracteriza a situação de uma entidade, onde uma entidade é uma pessoa, lugar ou objeto considerado relevante para a interação entre um usuário e uma aplicação. O contexto pode ser utilizado para ampliar o conhecimento que se tem sobre uma determinada situação, desempenhando um papel importante em qualquer domínio que envolva requisitos como compreensão, raciocínio, resolução de problemas ou aprendizado [Santoro *et al.* 2006]. A utilização de informações contextuais facilita a comunicação entre usuários e aplicações, ajuda a diminuir ambiguidades e conflitos, facilitando também a identificação, armazenamento e manipulação de perfis e modelos do usuário.

Consideramos que o contexto do usuário pode ajudar a construir o modelo do usuário (perfil, interesses, histórico) e, conseqüentemente, pode contribuir na personalização de consultas, provendo usuários com respostas mais significativas. O desafio nesse sentido é como automaticamente adquirir, gerenciar e usar essa informação de contexto de usuário. Neste panorama, um projeto de pesquisa desenvolvido no IFPB vem implementando o *CODI4In*, um *plugin* que trata do armazenamento e recuperação de informações contextuais do usuário [Freitas *et al.* 2012]. Por se tratar de um *plugin*, é possível acoplá-lo a diferentes aplicações computacionais que envolvam o processamento de consultas de usuários.

Uma possibilidade para construção automática do modelo do usuário, a partir de seu contexto, é através da captura de dados usando redes sociais. Para este fim, neste trabalho, optamos pelo uso do Facebook¹, a rede social mais utilizada no mundo. Para isso, inicialmente foram capturadas as informações básicas (nome, sobrenome, e-mail e sexo) do perfil e, em seguida, decidimos capturar os filmes de seu interesse. Uma vez com os nomes desses filmes, cruzamos estes dados com a API do IMDB² a fim de capturar os gêneros de cada filme para preenchermos o modelo contextual do usuário, criando um perfil, onde suas preferências armazenadas foram acessadas, interferindo nos resultados das consultas. Os usuários puderem então obter respostas mais personalizadas.

Com o intuito de testar a abordagem definida, desenvolvemos uma extensão do *plugin CODI4In* e acoplamos o mesmo ao MovieShow, uma aplicação web já existente para consulta a filmes por autor, diretor ou data de lançamento. Através do *plugin*, o contexto do usuário foi considerado em termos de suas preferências quanto aos gêneros dos filmes, e as consultas personalizadas.

¹ <http://www.facebook.com>

² <http://www.imdb.com>

Antes deste trabalho, por meio do acoplamento do *plugin CODI4In* ao MovieShow, era possível a definição explícita dos gêneros favoritos. Com a extensão implementada do *plugin* conseguimos realizar essa definição implicitamente (quando logado via Facebook, os filmes curtidos do usuário são capturados). Quando considerado o contexto, os resultados das consultas na aplicação são ranqueados de acordo com os gêneros dos filmes preferidos do usuário. Foram realizados testes e experimentos com usuários reais que demonstraram diferenças significativas em termos de satisfação dos mesmos em relação às respostas obtidas (personalizadas) quando seu contexto era considerado.

1.4.Organização do Relatório

Além do presente capítulo, este relatório está dividido como segue:

- No Capítulo 2, é descrito o estado da arte e as tecnologias envolvidas no desenvolvimento do *plugin CODI4In* e do módulo corrente deste relatório.
- No Capítulo 3, é apresentada a abordagem *CODI4In*, mostrando as funcionalidades existentes antes da versão desenvolvida neste trabalho.
- No Capítulo 4, é apresentada a especificação e a implementação realizada no *plugin CODI4In*, onde os dados são extraídos do Facebook e IMDB e utilizados para montar o contexto do usuário, melhorando os resultados das consultas.
- Finalmente, no Capítulo 5, as contribuições do trabalho são apontadas, dificuldades encontradas são relatadas e possíveis trabalhos futuros são indicados.

2. Estado da Arte

Neste capítulo será abordado o estado da arte em relação aos principais conceitos envolvidos para identificação do contexto do usuário, personalização de consultas e captura dos dados através de redes sociais. Além disso, são apontados tópicos necessários ao desenvolvimento do *plugin CODI4in* assim como são mostrados alguns trabalhos relacionados que também fazem personalização de consultas.

2.1. Personalização de Consultas

No mundo da rede global (internet), com o crescimento do volume de informação heterogênea por diversos canais, fica cada vez mais difícil responder corretamente o que o usuário precisa em cada requisição ou consulta, independente da plataforma em que ele está acessando [Tanca *et al.* 2011]. Muitas vezes, consultas submetidas resultam em dados massivamente obtidos, algumas vezes redundantes ou não relevantes ao que de fato o usuário necessita naquele momento, transformando uma tarefa que deveria ser simples e eficaz em algo bastante desafiador. Devido a essa problemática, várias pesquisas vêm sendo feitas buscando minimizá-la, através de estratégias de *personalização da consulta* [Arruda *et al.* 2010, Souza 2009, Kostadinov 2007, Koutrika 2004]. A ideia por trás da personalização da consulta é que diferentes usuários, ao realizar uma consulta, possam obter resultados diferentes como resposta, de acordo com um modelo de usuário identificado [Koutrika e Ioannidis 2004]. Uma vez que suas preferências foram coletadas podemos identificar o perfil do usuário e disponibilizarmos consultas personalizadas.

Os termos personalização e customização causam às vezes algumas dúvidas. A Figura 1 apresenta uma visão geral do funcionamento da personalização e customização e o quanto são parecidos para a aplicação (Engine), mas, ao mesmo tempo, diferentes na perspectiva do usuário. Os termos são explicados, a seguir:

- *Customização* ocorre quando o usuário altera o comportamento do aplicativo para prover o conteúdo de acordo com as suas preferências (informações fornecidas). Essas informações são especificadas explicitamente através de parâmetros de configurações (por exemplo, preenchimento de formulários na aplicação). Permitindo assim que o usuário saiba os resultados que ira encontrar (saída na tela), seja através da realização de consultas ou sugestões ao decorrer do uso do aplicativo, entretanto, muitas vezes não é conveniente informarmos os detalhes técnicos para utilização da aplicação, o que faz da customização uma tarefa inviável para iniciantes ou usuários impacientes. As configurações muitas vezes

possibilitam apenas a customização de parte do sistema, dando a falsa sensação de total controle para o usuário.

- *Personalização* significa entregar ao usuário, o que ele procura de acordo com as informações fornecidas. Essas informações podem ser providas explicitamente ou implicitamente. O perfil do usuário é construído explicitamente, no decorrer do uso da aplicação de acordo com as informações providas e consideradas relevantes (por exemplo, avaliação do conteúdo, lista de conteúdo favorito) pela aplicação para preenchimento do seu perfil. Para construção implícita do perfil do usuário é observado o histórico de navegação, tempo gasto em determinada página, produtos comprados e lista de conteúdo favorito. Algumas vezes pode ocorrer a rejeição do usuário sobre a personalização, resultando em casos de perfis construídos de forma limitada, incorreta ou incompleta. Um exemplo diz respeito aos anúncios do Google³, onde através do cookie DART anúncios são exibidos para seus usuários com base no histórico de navegação. Caso o usuário não esteja satisfeito com as sugestões dos anúncios, os mesmos podem desabilitar este cookie através da página de Política de privacidade⁴ da rede de conteúdo e dos anúncios Google.

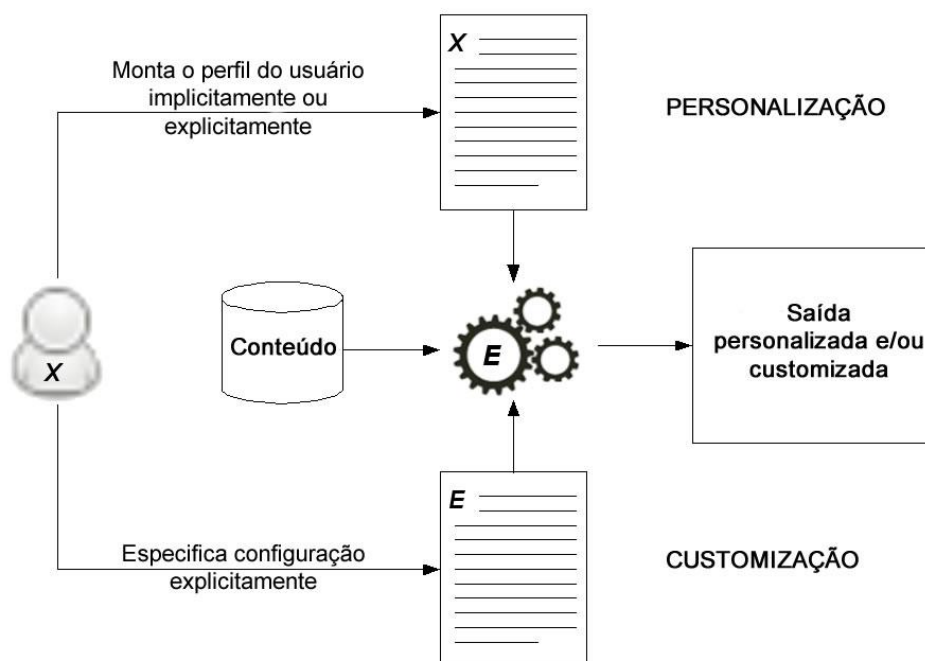


Figura 1. Personalização e Customização [Eynard et al. 2008]

³ <http://www.google.com>

⁴ <http://www.google.com/intl/pt-BR/policies/>

Mais especificamente, segundo [Koutrika e Ioannidis 2005] *personalização de consulta* é um processo dinâmico para melhorar a consulta de acordo com as preferências do usuário, armazenadas e relacionadas a um perfil visando fornecer respostas personalizadas. Quando aplicamos um dos conceitos relatados anteriormente (customização ou personalização) aos resultados de consultas, ranqueando-as de acordo com as preferências do perfil do usuário, tornamos a consulta personalizada.

[Koutrika e Ioannidis 2005] apresentam um exemplo de dois usuários, Al e Julie, acessando uma aplicação web sobre filmes procurando por filmes de comédia. Al é fã do diretor W. Allen, mas Julie não. A maioria dos sistemas irá considerar apenas a categoria comédia e retornará a mesma lista de filmes de comédia para ambos os usuários. Porém, armazenar as preferências dos usuários nos possibilita um resultado mais personalizado (algumas vezes, menor). Os resultados personalizados para Al incluiriam filmes do diretor W. Allen, enquanto para Julie não. Dessa forma, as preferências de cada usuário são relacionadas à consulta e como dinamicamente irão afetar o resultado da resposta, ocorre uma personalização de acordo com as informações coletadas do perfil do usuário.

2.2.Contexto

Segundo [Dey 2001], contexto se refere a qualquer informação que caracteriza a situação de uma entidade, onde uma entidade é uma pessoa, lugar ou objeto considerado relevante para a interação entre um usuário e uma aplicação. O contexto pode ser entendido como o que está por trás da habilidade de definir o que é ou não relevante em um dado momento [Vieira *et al.* 2006, Dey 2001].

Uma possibilidade para definirmos se é relevante ou não, seria comparar as informações personalizadas que possuímos disponíveis no momento, com o contexto ativo. A informação personalizada é construída com base nas características (por exemplo, interesses, classe social, situações) de cada indivíduo [Tanca *et al.* 2011]. Cada indivíduo possui uma situação definida pelo conjunto de informações personalizadas coletadas durante o uso da aplicação. Quando estas informações capturadas afetam a execução da consulta, podem personalizar o resultado da mesma, gerando dados mais específicos ao contexto do usuário.

Uma observação importante para uma personalização eficaz deve ser notada no momento em que o sistema calcula e reformula as informações de acordo com dados implícitos ou explícitos capturados pelo sistema a partir do ambiente externo ou do comportamento do usuário.

Alguns desafios foram identificados em relação ao uso do contexto [Tanca *et al.* 2011]:

- Gerenciamento do Contexto Dinâmico: No desenvolvimento da aplicação não é possível prever todos os contextos que serão abordados, uma vez que a cada

navegação ou consulta novos contextos aparecerão. Deste modo o projeto da aplicação (*views*) se torna um grande desafio, no sentido de antecipar contextos ou trabalhar com contextos compartilhados, onde outros usuários possam estar tendo a mesma experiência.

- Gerenciamento do Contexto Ativo: A cada momento, o contexto ativo (pode ser mais de um) precisa ser atualizado, pois como mencionado acima alguns aspectos do contexto atual podem ser derivados do ambiente (por exemplo, tempo de uso, localização) enquanto outros podem exigir uma ação explícita do usuário (por exemplo, curtir no Facebook) ou uma aplicação com capacidade de aprendizagem (por exemplo, iterar sobre o histórico de navegação). Uma vez reconhecido o contexto, a contextualização deve ser aplicada nas consultas do usuário e os dados relativos ao contexto devem ser entregues como resposta aos usuários. Preferências contextuais podem ser utilizadas para filtrar as *views* associadas aos contextos, através de um ranking de importância de dados exibindo os mais interessantes primeiramente.
- Gerenciamento de Dados de Acesso: Inúmeras aplicações web disponibilizam parte de seus dados para consulta (por exemplo, via *Web Services* [Tanca *et al.* 2011]), tornando-se preciosas fontes de informações para construção do perfil do usuário. Porém cada aplicação modelou a entidade usuário de uma maneira. Para acessarmos essas fontes de informações precisamos entender como foram organizados.

Estas questões levaram a uma demanda por mecanismos que pudessem prover o *contexto* [Tanca *et al.* 2011; Levandoski e Khalefa 2011; Vieira *et al.* 2010; Souza *et al.* 2008]. No entanto, percebe-se ainda uma lacuna entre as aplicações que precisam deste serviço e a forma como este serviço é implementado e disponibilizado. Um serviço com este propósito precisa prover o gerenciamento do contexto (por exemplo, armazenamento, recuperação e inferência) de modo que a aplicação os utilize como uma melhoria de seus serviços. No caso de consultas, o contexto capturado e manipulado deve ser gerenciado e usado como meio de filtrar ou ranquear os dados da aplicação, tornando-os mais relevantes para o usuário, de acordo com a necessidade corrente dele em relação a uma determinada consulta.

2.3.Redes Sociais

As redes sociais são *sites* que oferecem ferramentas e serviços de comunicação e interação centrados em um padrão de relacionamentos. Algumas pessoas utilizam redes interpessoais preexistentes através da comunicação mediada por computador. Outros propiciam a

produção narcísica de perfis sem vínculos obrigatórios com a realidade e estimulam a competição pelo aumento compulsivo da rede de contatos, incluindo “estranhos” [Aguilar, 2007].

Segundo [Karapantelakis *et al.* 2010], o uso de redes sociais tem aumentado nos últimos anos: evidências estatísticas indicam que não só são mais pessoas participando dessas comunidades, mas existe um aumento na quantidade média de tempo gasto. Este aumento ajudou a agregar valor a estas redes.

As empresas, percebendo o aumento do número de usuários, começaram a participar das redes sociais e tornaram-se provedores de conteúdo, fornecendo serviços personalizados aos seus assinantes. Isso não somente torna as redes sociais mais atraentes para os outros usuários, mas também uma fonte de receitas em potencial.

Atualmente, os serviços de conteúdos personalizados de redes sociais utilizam o contexto pessoal superficialmente. Alguns tipos de serviço exigem intervenção do usuário para funcionar como, por exemplo, na personalização da página do perfil e na instalação de aplicativos de terceiros. Algumas redes também fornecem conteúdo com base em recomendações [Karapantelakis *et al.* 2010]. Os critérios para a seleção dos conteúdos são de acordo com o contexto pessoal dos requisitos do usuário. Outros exemplos incluem a localização, idade, serviços de publicidade e sugestões de novos contatos com base na atual rede de amigos além de um serviço para lembrar aniversários.

Preocupando-se com a privacidade dos usuários e a sobrevivência no mercado, as redes sociais (Facebook, Myspace⁵, LinkedIn⁶ e Google+⁷) não suportam o compartilhamento do contexto pessoal (informações personalizadas) entre elas, portanto, proíbem a utilização de informação agregada que poderia ser utilizado para fornecer melhores resultados nos serviços personalizados, como por exemplo em anúncios do Google. Uma possibilidade para acesso de algumas informações personalizadas relacionadas ao contexto pessoal seria através da API das redes sociais, caso esteja disponível.

A utilidade da informação agregada é analisada através de resultados de estudos estatísticos que mostram geralmente que, caso compararmos varias redes sociais, um mesmo perfil irá apresentar resultados diferentes em cada uma dela [Karapantelakis *et al.* 2010].

A seguir, apresentamos em mais detalhes a rede social Facebook, a rede social escolhida como um dos elementos constituintes de estudo para o desenvolvimento deste trabalho.

⁵ <http://myspace.com>

⁶ <http://linkedin.com>

⁷ <http://google.com/+>

2.3.1. Facebook

Fundada em 2004, o Facebook tem como missão tornar o mundo mais aberto e conectado. As pessoas usam esta rede social para manter contato com amigos e familiares, descobrir o que está acontecendo no mundo e para compartilhar e exportar o que importa para eles.

Segundo [Newsroom 2012] na seção *Key Facts*, o Facebook com 3500 empregados expressa dados impressionantes:

- 901 milhões de usuários ativos mensais no final de março de 2012.
- Aproximadamente 80% dos usuários ativos mensais estão fora os EUA e Canadá.
- 526 milhões de usuários ativos, em média, em março de 2012.
- Em março de 2012, em média, 398 milhões de usuários estavam ativos no Facebook em pelo menos seis dos últimos sete dias.
- Mais de 125 bilhões de conexões de amigos no Facebook, no final de março de 2012. Em média mais de 300 milhões de fotos carregadas no Facebook por dia nos três meses, encerrados em 31 de março, 2012.
- Uma média de 3,2 bilhões de *Likes* e comentários gerados por usuários do Facebook por dia durante o primeiro trimestre de 2012.
- Mais de 42 milhões de páginas com dez ou mais curtidas no final de março de 2012. O Facebook está disponível em mais de 70 idiomas.

De acordo com estudos realizados através do Socialbakers⁸, um instituto especializado em mídias sociais, no Brasil a rede social ocupa o primeiro lugar, ultrapassando com uma certa distância o Orkut⁹, que até 2011 era predominante no país com pouco mais de 30 milhões de usuários. Produtos como Linha do Tempo, Registro de atividades, *Feed* de Notícias, Fotos, Vídeo, Grupos, Mensagens, Páginas, Lista de amigos, Eventos, Assinatura e Páginas compõem alguns dos diferenciais desta rede.

Os autores também disponibilizaram uma plataforma para os desenvolvedores e empreendedores criarem aplicativos sociais, *sites* e dispositivos integrados, seja através de um jogo, assistindo um filme, compartilhando seu restaurante favorito, ou ouvindo música (Figura 2) [Newsroom 2012].

⁸ <http://www.socialbakers.com/facebook-statistics/>

⁹ <http://www.orkut.com>



Figura 2. Login com aplicativo e autorização

Conforme exibido na Figura 2, uma vez que a permissão para captura dos dados seja concedida pelo usuário, através desse aplicativo podemos capturar alguns dados de seu perfil para utilização. Esta foi uma das propostas utilizadas no *plugin CODI4In* para capturar e montar o perfil do usuário. Mais detalhes serão fornecidos no Capítulo 4.

2.4.Trabalhos Relacionados

Os sistemas web são o maior foco de utilização de personalização na atualidade. Exemplos de serviços que utilizam personalização são o Google¹⁰ e o Amazon¹¹. O primeiro utiliza um histórico de cada usuário através de suas buscas e das páginas que foram mais visitadas para cada busca. O segundo, por sua vez, armazena o histórico de compras dos usuários e trabalha com sugestão de itens relacionados às compras efetuadas por eles.

Em termos de ambientes de bancos de dados, o trabalho de [Koutrika e Ioannidis 2005] desenvolveu um framework de personalização para sistemas de bancos de dados baseado nos perfis dos usuários. Os perfis são criados e armazenados, podendo ser definidos de forma explícita, quando o usuário informa suas preferências ou implicitamente pelo sistema através da análise da interação do usuário com o banco. A cada preferência é designado um escore entre 0 e 1 para auxiliar na execução da consulta e na classificação das respostas obtidas.

Kostadinov e seu grupo [Kostadinov *et al.* 2007] propõem um algoritmo para rescreita da de consulta, onde consideram que as fontes de dados são distribuídas num sistema baseado em mediadores. Para tal, eles utilizam três modelos: (i) um modelo de perfis, composto por cinco

¹⁰ <https://www.google.com.br>

¹¹ <http://www.amazon.com/>

dimensões (domínio de interesse, dados pessoais, qualidade, segurança e apresentação), (ii) modelo de contexto que trata de informações espaço-temporais, e (iii) um modelo de preferências que organiza as preferências dos usuários. Baseando-se nesses modelos apresentaram um experimento relatando o resultado das consultas reescritas e outro experimento verificando o tempo de execução do algoritmo.

[Stefanidis *et al.* 2009] criam um sistema de recomendação que expande resultados de consultas com base nas preferências dos usuários. Assim, tuplas que originalmente não seriam retornadas, são identificadas como sugestões. Por exemplo, ao se consultar por filmes dirigidos por Woody Allen, o sistema pode recomendar também sua biografia. Esse sistema pode computar os resultados das consultas utilizando o histórico do usuário, o estado corrente da consulta e da base de dados.

O projeto CareDB [Levandoski e Khalefa 2011] apresenta meios de incorporar o contexto e preferências no processamento de consultas em sistemas de banco de dados. Para isso, esse projeto foi desenvolvido como um sistema de banco de dados relacional completo, suportando funcionalidades como: (i) métodos de avaliação de preferências multi-objetivas (por exemplo, *skyline*, *topk*) e (ii) integração de informações contextuais de ambiente (por exemplo, tempo). Ambas as funcionalidades estão incluídas no processador de consulta dentro do sistema de banco de dados.

[Eynard *et al.* 2008] desenvolveram um *plugin* para o navegador Firefox¹², uma extensão chamada *RDFMonkey*, que, uma vez instalada, nos permite a visualização de conteúdo relacionado de acordo com o site visitado. Por exemplo ao visitarmos a página da banda “Metallica” na Wikipédia¹³, o *plugin* realiza uma consulta ao Freebase¹⁴, um banco de dados que disponibiliza links para sites externos de acordo com o termo pesquisado, recupera esses links acessando-os e exibindo-os na barra lateral do navegador.

Segundo [Tanca *et al.* 2011] está em desenvolvimento a PerLA¹⁵ (PERvasive LAnguage), uma linguagem declarativa de alto nível, para facilitar o trabalho com diferentes tecnologias (dispositivos GPS e outros sensores) para manipulação de contexto. Originalmente concebida para trabalhar com sensores de consulta, esta linguagem nos permite: (i) definir a estrutura do esquema de contexto e adquirir parte de informações de contexto que não pode ser capturadas através de leituras realizadas pelos sensores; (ii) criar um contexto em um esquema definido; (iii) ativar ou desativar um contexto em tempo de execução e (iv) realizar ações contextuais necessárias no sistema.

¹² <http://br.mozdev.org>

¹³ <http://www.wikipedia.org>

¹⁴ <http://www.freebase.com>

¹⁵ <http://perlawsn.sourceforge.net>

Comparando estes trabalhos ao nosso, percebe-se que, apesar da maioria deles também fazerem uso das preferências do usuário para prover personalização, em geral, eles não utilizam redes sociais para identificação do contexto do usuário. Neste trabalho, identificamos as informações desse contexto através do Facebook, capturando suas informações e montando o perfil do usuário. Também utilizamos *Web Services* de terceiros para enriquecermos as informações do perfil do usuário as quais são utilizadas para personalização dos resultados das consultas realizadas. Mais detalhes sobre esse processo serão apresentados no Capítulo 4.

3. Abordagem CODI4In

Este capítulo aborda os conceitos utilizados para construção do *plugin CODI4In*. Serão apresentadas algumas informações sobre a ontologia de contexto e os metadados utilizados, a arquitetura do *plugin* e as possibilidades de acoplamento a uma aplicação.

3.1. Conceitualizando o CODI4In

Como comentado anteriormente, o contexto pode ser entendido como o que está por trás da habilidade de definir o que é ou não relevante em um dado momento [Vieira *et al.* 2006, Dey 2001]. Dessa maneira, todo o processo de personalização de consultas, incluindo a obtenção e tratamento de preferências, histórico de utilização, captura de informações dinâmicas como localização do usuário e interface de consulta corrente, pode ser facilitado através do uso de contexto. Entretanto, gerenciar o contexto implica na implementação de tarefas como aquisição, representação, armazenamento, recuperação e uso do contexto. Neste sentido, para utilização do contexto, é necessário utilizar um modelo de representação que proveja uma mesma compreensão semântica tanto para agentes humanos quanto de software. Diante desta premissa o *CODI4In* utiliza uma ontologia como meio de representar e armazenar as informações contextuais dos usuários. A escolha de ontologias como modelo de representação de contexto possui algumas vantagens como [Souza *et al.* 2008]: a ontologia é um modelo portátil; é um modelo formal; permite reusabilidade; e provê mecanismos de inferência. A seguir, é descrita a ontologia de contexto empregada neste projeto.

3.2. A Ontologia de Contexto CODI-User

A ontologia CODI-User é uma extensão da ontologia CODI (*Context Ontology for Data Integration*). A CODI foi desenvolvida com base em requisitos da área de Integração de Dados e de gerenciamento de dados em ambientes P2P [Souza *et al.* 2008]. Na CODI, são definidos dois conceitos principais: entidade de domínio (*Domain Entity*) e elemento contextual (*Contextual Element*). A ideia é que os elementos contextuais sejam construídos e/ou capturados sobre seis entidades de domínio: *user*, *environment*, *data*, *procedure*, *association* e *application*.

Com o propósito de criar uma ontologia de contexto de usuário independente de aplicações específicas, foi realizada uma extensão da CODI com foco na entidade de domínio *User* [Freitas *et al.* 2012a]. Esta foi denominada de CODI-User. Tendo em vista que a ontologia de contexto de usuário não deveria se ater a apenas preferências ou interesses, foram incorporados à CODI-User, além destes, elementos contextuais diversos como, por exemplo, seu *expertise*, *habilidades*, *fatores físicos*. Elementos comuns a ambientes de consulta como

localização, dispositivo, interface e tarefa também foram incluídos. Uma visão da entidade de domínio *User* juntamente com alguns de seus elementos contextuais é apresentada na Figura 3. Neste fragmento, são mostrados apenas metadados, referentes a relacionamentos da entidade *User* e alguns elementos contextuais (*Location*, *Expertise*, *PersonalInfo*, *Connection*, *Interest* e *Preference*).

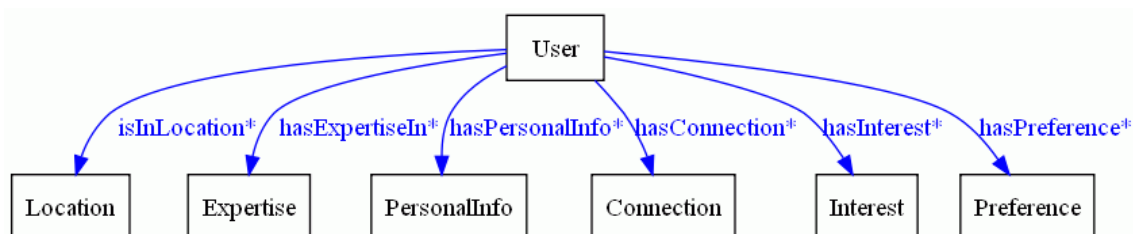


Figura 3. Fragmento da ontologia de contexto CODI-User

Utilizando o software Protege¹⁶, conseguimos visualizar um exemplo da ontologia instanciada, como mostra a Figura 4. Analisando o fragmento apresentado, visualizamos uma instância da entidade de domínio *User*, identificada como “Maria”, com seus elementos contextuais associados num dado instante. Neste momento, Maria tem interesse em música, expertise em cantar e preferências por Rock e MPB.



Figura 4. Exemplo de instância da CODI-User

3.3. Arquitetura do CODI4In

O *CODI4In* foi especificado como um *plugin* que pode ser acoplado a diversos tipos de aplicações que envolvam consultas a dados. Ele tem como objetivo prover a obtenção e persistência de informações (elementos) contextuais, de forma dinâmica e incremental, a partir das interações dos usuários e de suas consultas. Em outras palavras, trabalha como um serviço *back-end* ampliando funções de aplicações *front-end*, a partir do uso das informações contextuais armazenadas na CODI-User. A Figura 5 apresenta uma visão da arquitetura do *CODI4In* e de

¹⁶ <http://protege.stanford.edu/>

sua relação com aplicações *front-end*. Assim, qualquer aplicação que envolva consultas, como, por exemplo, o MovieShow (a ser descrito na Seção 3.5) e/ou o módulo de consultas de um sistema PDMS (*Peer Data Management System*) [Souza et al. 2011], são exemplos de aplicações *front-end* que podem usar o serviço do *CODI4In*.

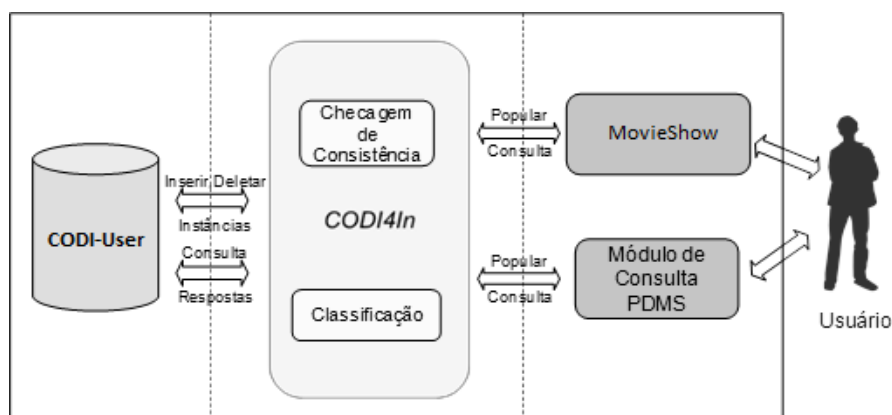


Figura 5. Arquitetura do CODI4In

Dessa maneira, no momento em que o usuário realiza uma consulta, o serviço coleta as informações (elementos contextuais) da base da CODI-User e efetua a personalização dos dados (por exemplo, através de uma filtragem ou ranqueamento), valendo-se não só dos parâmetros da consulta, mas também, dos elementos contextuais identificados e capturados pelo *CODI4In*. Como ilustração, suponha que um usuário submeta a seguinte consulta: “Mostre os pontos turísticos da cidade”. No momento da submissão, o *CODI4In* identifica que o usuário se encontra (localizado) no Rio de Janeiro. Considerando este elemento contextual (localização), o *CODI4In* efetua a personalização dos resultados, por meio de uma estratégia de ranqueamento, e exibe como primeiros resultados "visita ao Cristo Redentor" ou "bondinho do Pão de Açúcar".

3.4.O Plugin

O *CODI4In* foi desenvolvido e implementado utilizando a linguagem Java, manipulando e executando consultas na ontologia de contexto CODI-User. Ontologias como a CODI-User podem ser interpretadas e tratadas como grafos [An et al. 2006], onde conceitos são identificados como nós e os relacionamentos semânticos (como especialização, parte de) são identificados como as arestas do grafo. Considerando essa premissa, foi adotado o Neo4J¹⁷ para persistência da CODI-User. O Neo4J é um banco de dados baseado em grafo, que armazena dados como nós e arestas, e provê algoritmos de indexação dos nós. A escolha do Neo4J se justificou devido às facilidades de utilização de um banco de dados que fizesse uso do mesmo paradigma de representação das informações contextuais adotado neste trabalho. Assim, tornou-se mais

¹⁷ <http://docs.neo4j.org/>

Para que a aplicação possa utilizar os serviços do *CODI4In*, o usuário precisa, em primeiro lugar, efetuar seu cadastro. Para isso, o MovieShow disponibiliza um formulário com as requisições de informações necessárias para realizar esta ação. Uma vez cadastrado, o usuário pode, então, realizar login no sistema. Quando esta ação ocorre, o MovieShow verifica se o usuário está registrado em seu banco de dados. Em seguida, o *CODI4In* identifica este usuário e verifica se a estrutura básica de nós da CODI-User para este usuário já está armazenada na base de dados do Neo4J. Caso negativo, a CODI-User para este usuário é montada, os nós importantes relacionados ao usuário em questão são indexados e armazenados na base do Neo4J. O usuário então é direcionado à página da aplicação (Figura 7), podendo, a partir deste ponto, utilizar as três funcionalidades básicas desenvolvidas nesta versão do MovieShow:

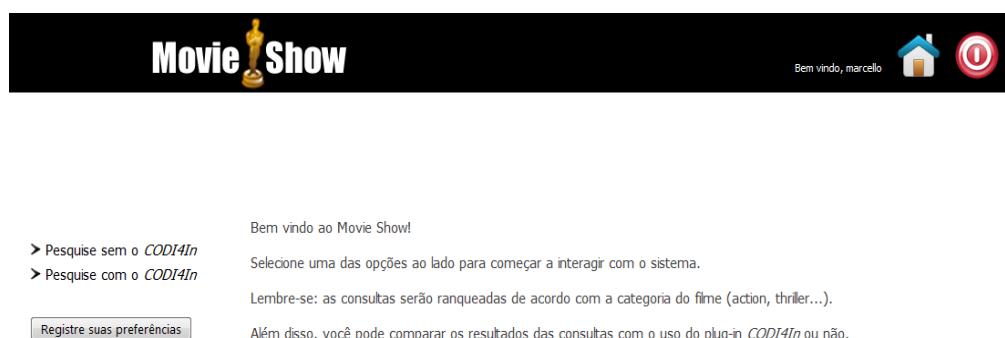


Figura 7. Interface do MovieShow

- Registro de preferências: o usuário pode informar *de forma explícita* e em formato de ranking, os gêneros de filme (por exemplo, suspense, comédia) de que mais gosta. Para isso, o usuário identifica uma lista de gêneros colando no topo dessa lista aqueles que mais lhe interessa, formando, assim, um ranking de preferências. O objetivo é que o *CODI4In* tome por base este ranking de preferências e personalize as consultas de acordo com os gêneros listados pelo usuário dando prioridade aos gêneros que estão no topo.
- Submissão de consultas sem o uso do *CODI4In*: o MovieShow apenas efetua uma consulta padrão no banco de dados relacional utilizando os parâmetros submetidos pelo usuário. O objetivo desta funcionalidade é mostrar os resultados das consultas *sem* a utilização das informações contextuais do usuário, ou seja, sem os serviços do *CODI4In*.

- Submissão de consultas com o uso do *CODI4In*: o MovieShow efetua uma consulta no banco de dados, mas os resultados são reorganizados (ranqueados) tomando por base as preferências identificadas do usuário. Nesta versão, estas preferências dizem respeito aos gêneros de filmes. O objetivo desta funcionalidade é identificar os benefícios obtidos quando considerado o contexto do usuário na realização da consulta.

Caso haja alguma atualização dos elementos contextuais do usuário, essa atualização é aplicada aos dados e o *CODI4In* as armazena em sua base de dados (Neo4J) atualizando também a estrutura montada na memória em tempo de execução.

Para melhor entendimento do uso e funcionamento da aplicação com o *CODI4In*, ilustraremos um exemplo com dois usuários distintos [Freitas *et al.* 2012b]. Neste exemplo, os usuários “Ben” e “Ana” tem um interesse em comum – filmes, entretanto possuem preferências diferentes com relação aos gêneros dos filmes. Ben gosta de filmes do tipo Documentário, Drama e Comédia, nesta ordem de prioridade. Ana, por sua vez, gosta de Drama, Comédia e Documentário, seguindo esta ordem de interesse. Na Figura 8, tela de consulta da aplicação, os usuários Ben e Ana submetem uma mesma consulta, com uso do *CODI4In*, sobre filmes estrelados pela atriz “Nicole Kidman”.

Para submeter uma consulta, complete os campos abaixo de acordo com o critério escolhido.

Nome do Ator/Atriz ▼	Nicole Kidman	Consultar
----------------------	---------------	-----------

Figura 8. Consulta submetida pelos Usuários Ben e Ana

A consulta SQL é gerada e executada tendo como condição de filtro o nome do ator/atriz selecionado. Os resultados são, então, retornados de forma ranqueada pelo MovieShow + *CODI4In*. Assim, as listas de filmes relacionados à atriz “Nicole Kidman”, exibidas para os usuários Ben e Ana são diferentes, isto é, são personalizadas, como mostram a Figura 9 e 10. Na lista exibida para o usuário Ben, os filmes do gênero Documentário são mostrados primeiro, depois os do gênero Drama e por fim os do gênero Comédia seguindo a ordem de suas preferências anteriormente cadastradas. O mesmo ocorre com o usuário Ana, que tem a sua lista de filmes exibida de acordo com suas preferências cadastradas.

Título	Gênero
101 Most Starlicious Makeovers	Documentary
76th Annual Academy Awards, The	Documentary
Climbing 'Cold Mountain'	Documentary
American Darlings	Drama, Musical
Birth	Drama, Mystery
Emma's War	Drama
Interpreter, The	Drama, Thriller
Bewitched	Comedy, Fantasy
58th Annual Tony Awards	Music

Figura 9. Resultados personalizados da consulta submetida pelo Ben

Título	Gênero
American Darlings	Drama, Musical
Birth	Drama, Mystery
Emma's War	Drama
Interpreter, The	Drama, Thriller
Bewitched	Comedy, Fantasy
101 Most Starlicious Makeovers	Documentary
76th Annual Academy Awards, The	Documentary
Climbing 'Cold Mountain'	Documentary
58th Annual Tony Awards	Music

Figura 10. Resultados personalizados da consulta submetida pela Ana

Por meio deste exemplo, pode-se perceber que usuários com modelos diferentes, construídos a partir do uso de informações contextuais, que utilizem uma aplicação de consulta a dados em um domínio comum a ambos, ao submeterem uma mesma consulta, poderão receber resultados diferentes. Estes resultados podem estar personalizados segundo estratégias diferentes, como o ranqueamento efetuado neste exemplo, de acordo com elementos contextuais específicos daquele usuário, como as preferências em relação aos gêneros de filmes. Vale salientar que, nesta versão inicial, o usuário tinha de explicitamente definir suas preferências em relação aos gêneros de filmes. Na versão desenvolvida neste corrente trabalho, estas preferências são capturadas automaticamente por meio do Facebook, como será explicado no próximo capítulo.

4. Extensão ao *CODI4In*: Implementação e Resultados

Este capítulo apresenta a abordagem que envolveu o aprimoramento do *plugin CODI4In*, desenvolvida neste trabalho. Para tal, descreve a ideia geral, mostra a especificação realizada, explicitando os requisitos funcionais e características técnicas, a implementação e os resultados obtidos.

4.1. Especificação

A proposta do *plugin* é prover resultados de consultas personalizados de acordo com o contexto do usuário, quando acoplado a qualquer aplicação que utilize a linguagem Java. Neste momento, a personalização é realizada por meio do ranqueamento dos resultados das consultas de acordo com as preferências do usuário.

Neste sentido, o MovieShow, apresentado no capítulo anterior, atua no domínio de filmes, permitindo consultas por autor, diretor ou data de lançamento. Quando ativado o *plugin*, os resultados das consultas são ranqueados de acordo com os gêneros preferidos do usuário logado. Antes deste trabalho os gêneros preferidos eram definidos explicitamente.

O desafio da extensão ao *plugin* consiste na captura implícita e automática dos elementos contextuais do usuário (informações básicas e gêneros dos filmes) relacionado à aplicação de filmes MovieShow através da rede social Facebook e o consumo do *Web Service* do IMDB para captura dos gêneros dos filmes. Dessa maneira, evita-se o esforço do usuário, uma vez que, antes deste trabalho, era necessário um cadastro com as informações básicas e o preenchimento dos gêneros favoritos para que pudessem ser utilizados no ranqueamento das consultas quando utilizado o *plugin*. Sendo assim, podemos definir os principais requisitos funcionais da extensão como:

- Recuperar nome, sobrenome, e-mail e sexo do usuário através da rede social
- Recuperar opções curtir, em específico o nome dos filmes através da rede social
- Recuperar gêneros dos filmes através de uma base de dados confiável, como base nos títulos de filmes curtidos
- Tratar os dados recuperados e persistir na ontologia CODI-User, de forma que o *CODI4In* possa gerenciar
- Ordenar a lista de gêneros de filmes preferidos de cada usuário

Como requisitos não funcionais, alguns já citados anteriormente, utilizamos a rede social Facebook e a base de dados do IMDB.

A captura dos elementos contextuais do usuário foi possível no momento do *login* do usuário através do aplicativo *CODI4In* no Facebook, quando capturamos suas informações

básicas, os nomes dos filmes “curtidos” na rede social e cruzamos esses nomes com o banco de dados do IMDB, capturando seus gêneros, associando ao usuário e persistindo na ontologia CODI-User. Assim, quando realizada uma consulta utilizando o *plugin CODI4In*, iremos consultar essas preferências e interferir no resultado da resposta, resultando em uma lista de filmes ranqueados de acordo com os gêneros preferidos do usuário. A seguir explicaremos mais detalhadamente esse processo através de diagramas e algoritmos.

O diagrama de atividades, mostrado na Figura 11, representa o momento de login do usuário utilizando o Facebook, quando montamos o perfil do usuário. As principais etapas deste diagrama são descritas a seguir:

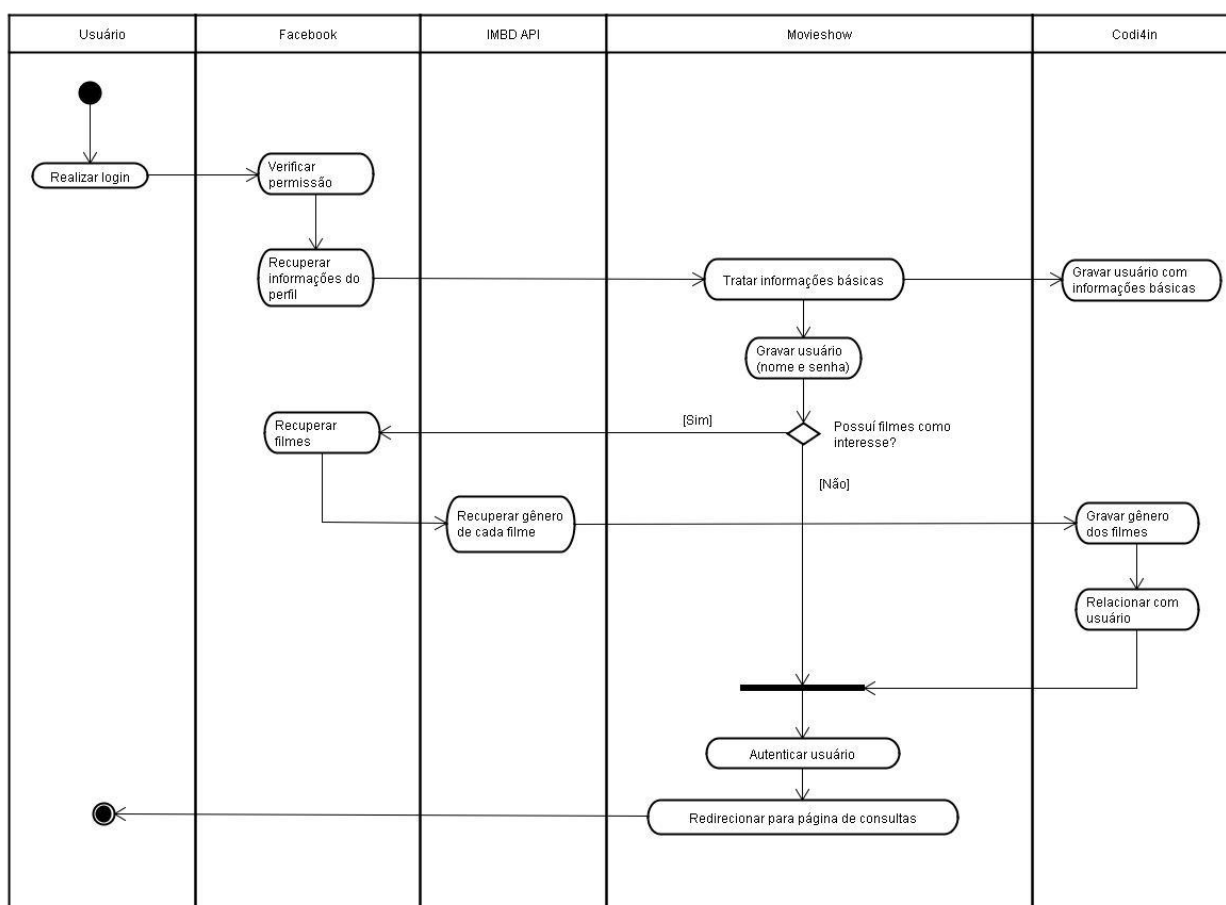


Figura 11. Login do usuário

- Realizando o login, chamamos a API do Facebook e executamos o método chamado login. O usuário precisa autorizar o aplicativo *CODI4In* no Facebook para disponibilização e recuperação das informações básicas do perfil (nome, sobrenome, e-mail e sexo) e suas opções curtir (nomes dos filmes curtidos).

- Com os dados retornados do Facebook, primeiramente tratamos e persistimos as informações básicas do perfil no *CODI4In* e no MovieShow (apenas o nome do usuário).
- Caso o usuário possua algum filme curtido no Facebook (por exemplo, ao curtir a página do filme “O Náufrago” – essa página deverá estar configurada na categoria Filmes), iteramos sobre a lista de filmes (os filmes serão consultados um por vez) e cruzamos o nome de cada filme com a API do IMDB¹⁹, assim recuperamos o gênero de cada filme e armazenamos em uma lista auxiliar.
- Uma vez que concluímos o processo de iteração dos filmes, ordenamos a lista auxiliar (onde todos os gêneros foram armazenados) pelo número de ocorrências e removemos os gêneros duplicados, relacionamos esses gêneros com o perfil do usuário, autenticamos o mesmo e redirecionamos para a página de consultas de filmes.

Conforme dito anteriormente, uma vez capturado o contexto do usuário e criado seu perfil, autenticamos o mesmo e redirecionamos para página de consulta. O diagrama de atividades mostrado na Figura 12, ilustra o momento da consulta do usuário utilizando ou não o *plugin CODI4In*. As principais etapas deste diagrama são descritas a seguir:

- Realizando a consulta, recuperamos a lista de filmes através do MovieShow. Quando o contexto do usuário é considerado, através do *plugin CODI4In* recuperamos os gêneros preferidos do usuário logado e utilizamos um algoritmo de ordenação para ranquearmos a lista de acordo com suas preferências.
- Enfim, exibimos a lista de filmes de acordo com a consulta realizada.

¹⁹ <http://www.imdbapi.com/>

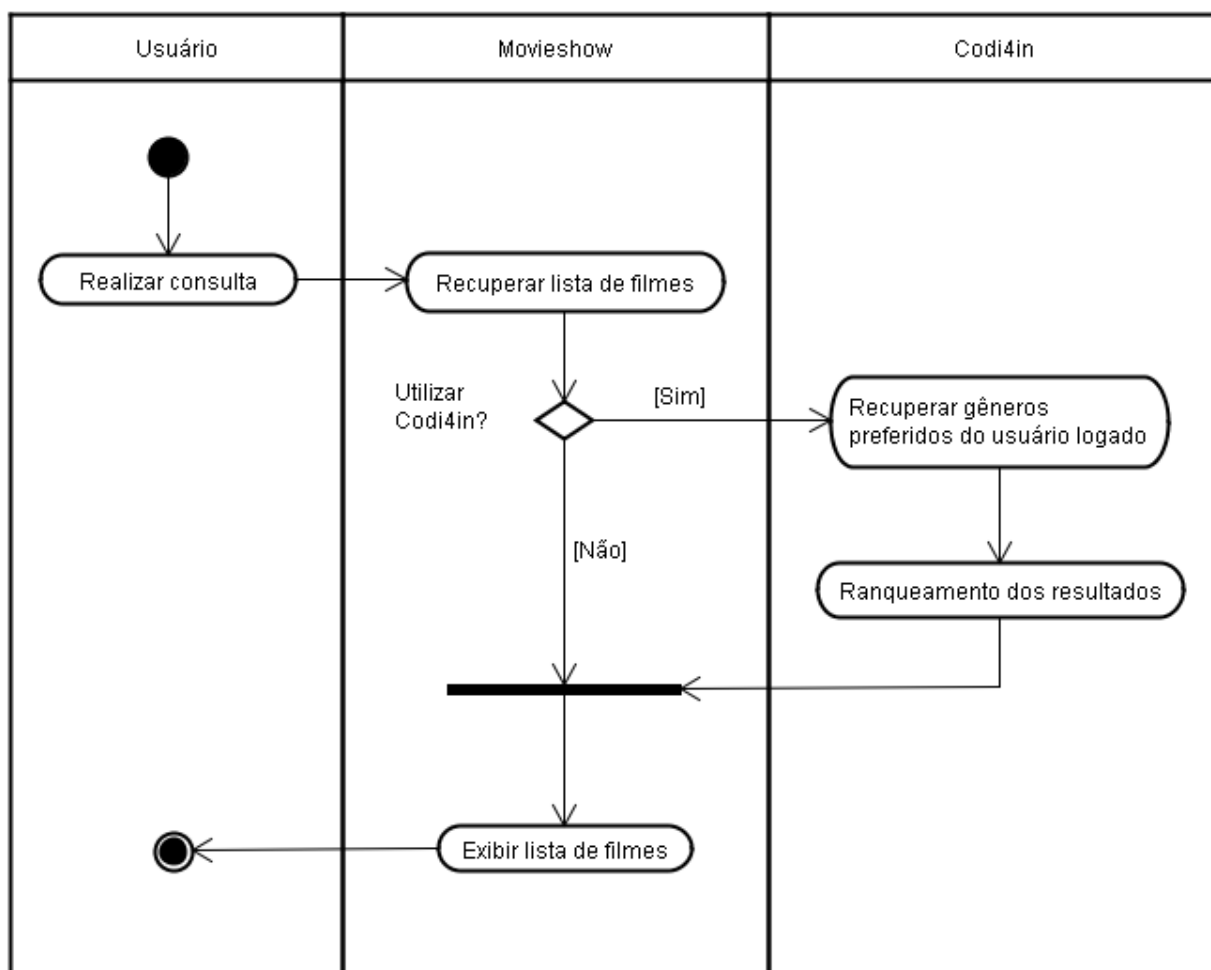


Figura 12. Consulta por filmes no MovieShow

4.2.Implementação

Em todos os trabalhos estudados nas atividades do projeto, sempre que alguma ferramenta foi proposta ou desenvolvida, a linguagem Java foi a escolha entre os desenvolvedores, a fim de mantermos a facilidade de integração com outros projetos de pesquisa em andamento na instituição que utilizam a mesma linguagem. Realizamos todo o processo de desenvolvimento de forma iterativa e incremental.

Para o trabalho com as APIs utilizamos JavaScript, uma linguagem *client-side* para navegadores web. Foram utilizadas duas APIs no desenvolvimento da ferramenta: a Facebook API utilizando a JavaScript SDK e a API do IMDB.

4.2.1. Facebook API

O núcleo do Facebook é o grafo social onde pessoas se conectam a tudo o que lhes interessam²⁰. A Graph API apresenta uma visão simples e consistente do grafo social do Facebook, através de representações uniformes dos objetos no grafo (por exemplo, pessoas,

fotos, eventos e páginas) e conexões entre esses objetos (por exemplo, relações de amizade, conteúdo compartilhado, fotos e *tags*)²⁰.

Todo objeto dentro do grafo social possui um id único, assim é possível acessar um objeto através da url `https://graph.facebook.com/ID`. Por exemplo, para buscarmos o objeto usuário, tal como o autor deste relatório, basta acessarmos `https://graph.facebook.com/diogoca`. Todas as respostas serão objetos serializados em JSON²¹, um subconjunto da notação de objeto de JavaScript. Entretanto seu uso não requer JavaScript necessariamente, pois trata-se de um formato de texto completamente independente da linguagem de programação a ser utilizada, uma vez que utiliza convenções de nomes familiares aos programadores. Essas propriedades fazem do JSON uma linguagem de troca de dados ideal. Mais adiante veremos um exemplo de retorno da API em JSON.

Exclusivamente para este trabalho, utilizamos a JavaScript SDK, um rico conjunto de funcionalidades disponível no *client-side* para realizarmos requisições na Facebook API, incluindo recursos da REST API²², Graph API e Dialogs²³.

No Facebook, na seção *Facebook Developers*, registramos um aplicativo no painel de controle, para utilização da SDK, uma vez que é necessário fornecer um ID para inicialização e utilização da API. A seguir um exemplo de inicialização da API usando o método *init*:

```
FB.init({
  appId      : 'ID_DO_SEU_APLICATIVO',
  channelUrl : '//'WWW.SEU_SITE.COM/channel.html',
  status     : true,
});
```

Com o aplicativo inicializado chamamos outros métodos para prosseguirmos com o processo de recuperação de dados do usuário. Conforme ilustrado na Figura 13, assim que o usuário realiza o login utilizando o aplicativo *CODI4In* no Facebook, recebemos o *access_token*, ou seja, um código de acesso representando a autorização do usuário para realização de futuras requisições.

²⁰ <http://developers.facebook.com/docs/reference/api/>

²¹ <http://www.json.org/>

²² <http://developers.facebook.com/docs/reference/rest/>

²³ <http://developers.facebook.com/docs/reference/dialogs/>



Figura 13. Login e autorização do CODI4In no Facebook

A seguir um trecho do código e a Figura 14 que ilustra o retorno do método *api* para recuperação das informações básicas e filmes:

```
FB.api('/me?access_token=' + TOKEN, function(me) { ...
    console.log(me);
    FB.api('/me/movies?access_token=' + TOKEN, function(movies) { ...
        console.log(movies);
    }
})
```

```

▼ Object
  ▶ education: Array[1]
    email: "diogoca@gmail.com"
  ▶ favorite_teams: Array[1]
    first_name: "Diogo"
    gender: "male"
    id: "100000684102739"
  ▶ inspirational_people: Array[2]
    last_name: "Alves"
    link: "http://www.facebook.com/diogoca"
    locale: "pt_BR"
    name: "Diogo Alves"
    timezone: -3
    updated_time: "2012-05-24T14:09:23+0000"
    username: "diogoca"
    verified: true
  ▶ __proto__: Object
facebook.jsf:34

▼ Object
  ▼ data: Array[5]
    ▼ 0: Object
      category: "Movie"
      created_time: "2012-07-17T14:46:36+0000"
      id: "102146656493710"
      name: "O professor alopado"
      ▶ __proto__: Object
    ▼ 1: Object
      category: "Movie"
      created_time: "2012-05-16T14:42:12+0000"
      id: "104044212964219"
      name: "Catch Me If You Can"
      ▶ __proto__: Object
    ▶ 2: Object
    ▶ 3: Object
    ▶ 4: Object
    length: 5

```

Figura 14. Retorno JSON das informações básicas e filmes do usuário

Iterando sobre o JSON exibido na Figura 14, conseguimos recuperar as informações básicas para persistimos o contexto do usuário. A Figura 15 ilustra uma página no Facebook configurada na categoria “Filme”, a mesma pode ser curtida, assim o nome do filme será capturado.



Figura 15. Página configurada como Filme

Para os filmes, não iremos persistir os títulos no contexto do usuário, apenas os gêneros preferidos que serão recuperados através do IMDB *WebService*, detalhado na próxima seção.

4.2.2. IMDB API

A IMDB API é um *Web Service* que disponibiliza informações sobre a maioria dos filmes cadastrados no site IMDB. Uma vez que não requer cadastro, utilizamos essa API para recuperação dos gêneros dos filmes do usuário, após o login. Fornecendo o nome do filme, conseguimos obter o gênero através da seguinte url `http://www.imdbapi.com/?t=NOME`. Existem três opções de retorno: XML, JSON e Texto. Escolhemos trabalhar com JSON, pois já estamos utilizando o mesmo formato com a Facebook API. A Figura 16 ilustra um exemplo de retorno, procurando pelo filme *Catch Me If You Can*:

```
▼ Object
  Director: "Steven Spielberg"
  Genre: "Biography, Comedy, Crime, Drama"
  Title: "Catch Me If You Can"
  Writer: "Jeff Nathanson, Frank Abagnale Jr."
  Year: "2002"
```

Figura 16. Retorno JSON do IMDB Webservice

Varrendo o objeto JSON exibido anteriormente, podemos capturar os gêneros através do atributo *Genre*, onde cada gênero é separado por vírgula. Neste trabalho, no final da captura de todos os gêneros, armazenamos os mesmos em uma lista auxiliar, ordenamos pelo número de ocorrências, removemos os nomes duplicados e recuperamos o cinco primeiros gêneros dessa lista para relacionarmos ao perfil do usuário. No MovieShow, quando realizarmos alguma consulta por filmes considerando o contexto, os gêneros preferidos serão consultados e ajudarão no ranqueamento do resultado, tornando a consulta personalizada. Na seção 4.3, iremos detalhar esses resultados.

4.2.3. O Algoritmo

Na Figura 17, apresentamos um algoritmo em alto nível que resume o momento de captura do contexto do usuário através do *plugin* e a ordenação dos gêneros de filmes. As atividades realizadas são:

```

1 getUserContext()
2 // Quando usuário realiza o login com Facebook e permite a
3 // recuperação de seus dados e opções curtir (ex., filmes)
4 token = getUserToken()
5 user = getFacebookUser(token)
6 moviesList = getFacebookLikedMovies(user)
7 // Coleção nova vazia
8 preferredGenres = []
9 for each moviesList in movie
10     genres = getGenresFromImdb(movie.name)
11     preferredGenres.push(genres)
12     if (at last one of moviesList = movie)
13         // Ordenar pelo número de ocorrências
14         sortGenresByOccurrence(preferredGenres)
15         // Remover gêneros duplicados
16         uniqueGenres(preferredGenres)
17         insertCE(user, preferredGenres)
18     end if
19 end for
20 endGetUserContext()

```

Figura 17. Algoritmo de captura do contexto do usuário

- Uma vez realizado o login com Facebook e autorizado o aplicativo pelo usuário, recebemos um token, uma chave de acesso que utilizamos para recuperarmos as informações básicas e as opções curtir do usuário.
- Através do *plugin CODI4In*, iteramos (consultamos um por um) sobre a lista de filmes de curtidos. Em cada iteração, recuperamos os gêneros do filme através da IMDB API e armazenamos na coleção *preferredGenres*.
- Quando o processo de iteração é concluído, ordenamos a coleção de gêneros preferidos de acordo com o número de ocorrências e removemos os nomes duplicados.
- No final, persistimos as informações básicas do perfil do usuário e os gêneros na ontologia CODI-USER, a qual será usada no ranqueamento das consultas realizadas, tornando-as personalizada.

4.3.Exemplo

A Figura 18 ilustra a tela inicial da aplicação web. O usuário deve realizar o login via Facebook para usufruir da implementação descrita nesse relatório.



Figura 18. Tela inicial da aplicação

Depois de autorizado o aplicativo do Facebook, redirecionamos o usuário, capturamos e tratamos automaticamente suas informações básicas e os gêneros de seus respectivos filmes. Conforme ilustrado na Figura 19, exibimos como retorno um alerta JavaScript,:

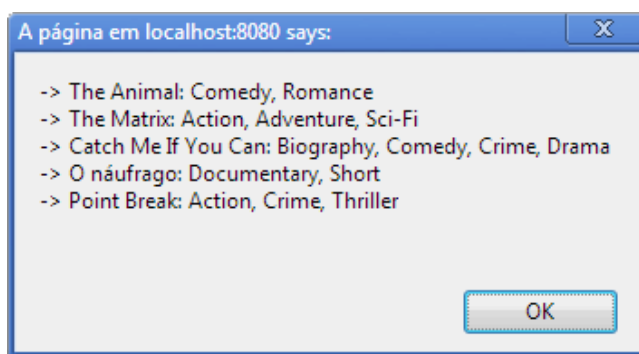


Figura 19. Filmes e gêneros capturados

Após a verificação de captura dos dados da rede social, redirecionamos o usuário para a página de consultas. Caso ele deseje verificar os seus gêneros preferidos capturados novamente, basta acessar a opção “Verifique suas preferências”.



Figura 20. Gêneros preferidos capturados

Na Figura 20, podemos perceber que os gêneros capturados foram automaticamente persistidos e relacionados na ontologia CODI-USER, sem qualquer interação explícita do usuário. Após a persistência, foram mostrados na interface.

Não alteramos o mecanismo de funcionamento de consulta de filmes, ou seja, quando pesquisarmos utilizando o *plugin CODI4In*, o contexto será considerado e os filmes serão ranqueados automaticamente de acordo com a preferência do usuário. As Figuras 21 e 22 ilustram o resultado da consulta de filmes pelo ator “Michael (I) Madsen”.

Comparando as Figuras 21 e 22, observamos que o ranqueamento foi realizado quando utilizado o *plugin*, alterando a ordenação dos resultados de acordo com os gêneros preferidos do usuário, uma vez que os filmes configurados na categoria *crime* e *action*, seguidos de *comedy* foram melhores posicionados. A seguir abordaremos sobre os testes realizados.

Título	Data de Lançamento	Gênero
Bloodrayne	2005-01-01	Action, Fantasy
Blueberry	2004-01-01	Western, Action
Chasing Ghosts	2005-01-01	Crime, Drama
Christmas Cop, A	2004-01-01	Comedy, Action
Distance, The	2005-01-01	Drama, Thriller
Driv3r	2004-01-01	Crime, Thriller
Drop, The	2005-01-01	Action, War
Drummer, The	2004-01-01	Romance, Action
Duplicity	2004-01-01	Thriller
Firedog	2004-01-01	Comedy, Animation
Frankenstein (2004/II)	2004-01-01	Mystery, Action
Hell Ride	2006-01-01	Drama, Thriller
Inglorious Bastards	2006-01-01	Drama, Action
Kill Bill: Vol. 2	2004-01-01	Drama, Romance
Making of 'Kill Bill: Volume 2', The	2004-01-01	Documentary

Figura 21. Resultado da consulta sem o CODI4In

Título	Data de Lançamento	Gênero
Chasing Ghosts	2005-01-01	Crime, Drama
Driv3r	2004-01-01	Crime, Thriller
Bloodrayne	2005-01-01	Action, Fantasy
Blueberry	2004-01-01	Western, Action
Christmas Cop, A	2004-01-01	Action
Drop, The	2005-01-01	Action, War
Drummer, The	2004-01-01	Action
Frankenstein (2004/II)	2004-01-01	Mystery, Action
Hell Ride	2006-01-01	Action, Drama
Inglorious Bastards	2006-01-01	Drama, Action
Kill Bill: Vol. 2	2004-01-01	Action, Drama
Firedog	2004-01-01	Comedy, Animation
Making of 'Kill Bill: Volume 2', The	2004-01-01	Documentary
Distance, The	2005-01-01	Drama, Thriller
Duplicity	2004-01-01	Thriller

Figura 22. Resultado da consulta com o CODI4In

4.4. Testes

Com objetivo de verificarmos se o *plugin* proporciona benefícios nos resultados das consultas, ou seja, se os resultados são ranqueados de acordo os gêneros preferidos do usuário, realizamos testes com os integrantes do projeto *CODI4In*, voluntários de outros projetos de pesquisa, alunos diversos e não alunos do IFPB, totalizando vinte e nove pessoas.

Primeiramente informamos a utilização da rede social como fonte de informações de contexto, em seguida pedimos a todos que curtissem filmes no Facebook (caso nunca tivessem curtido), depois acessassem o MovieShow (logando via Facebook) e realizassem algumas consultas por filmes com e sem o *plugin CODI4In*. Todos puderam comparar os resultados das consultas quando ativado o *plugin* ou não.

No final do teste solicitamos o preenchimento de um questionário online (resposta a algumas perguntas e sugestões, apresentado no Apêndice 1 e também disponível completo na url <http://bit.ly/movieshow>), a fim de avaliar os resultados das consultas. Os gráficos mostrados nas Figuras 23 e 24 ilustram os resultados obtidos.

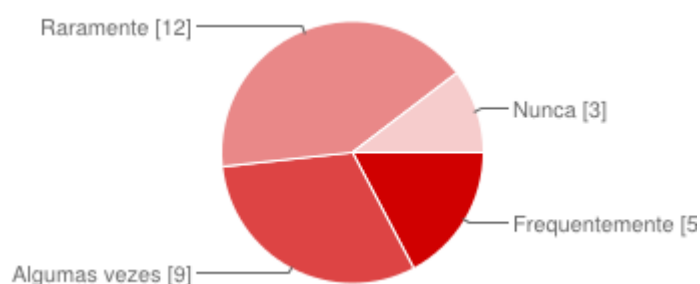


Figura 23. Resposta à pergunta: Frequência em que curte filmes no Facebook

Em relação à frequência em que usuários curtem filmes no Facebook, percebemos que os usuários raramente o fazem. Um dos motivos seria que a maioria deles prefere postar e compartilhar conteúdo (mensagens, fotos e vídeos) ao invés de buscar por páginas de filmes especificamente.

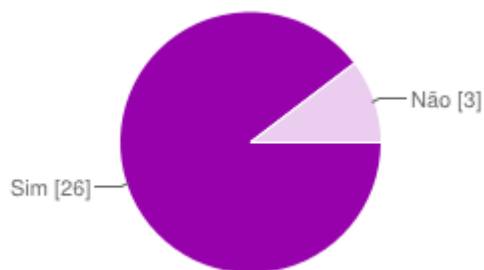


Figura 24. Resposta à pergunta: O uso do *plugin CODI4In* melhorou o resultado da consulta

Como mostra a Figura 24, para a grande maioria dos participantes que testaram a abordagem, quando habilitado o *plugin* para realização das consultas, tiveram uma melhora no resultado das mesmas. Entretanto, eles também relataram uma lentidão quando ativado o *plugin* comparado ao modo desativado, pois aplicamos um algoritmo de ranqueamento para ordenação do resultado da consulta, ao invés de exibirmos o resultado diretamente para o usuário. Em alguns casos, não conseguimos recuperar os gêneros do IMDB *WebService*, isso ocorre quando o Facebook retorna um filme com nome em português (ainda não traduzido). Também sugeriram a utilização de outras redes sociais para captura do contexto.

5. Considerações Finais

Neste trabalho realizamos um estudo sobre personalização de consultas, contexto, redes sociais, captura de dados do perfil do usuário no Facebook, focado no domínio de filmes, onde recuperamos os seus gêneros a partir da fonte de dados do IMDB. O objetivo principal era adicionarmos ao *plugin CODI4In* a captura automática e implícita dos dados e gêneros dos filmes favoritos do usuário, montando assim um perfil baseado no seu contexto que interferisse no ranqueamento dos resultados de consultas, de modo personalizado.

5.1. Contribuições

As principais contribuições deste trabalho foram:

- Levantamento do estado da arte em relação ao *plugin CODI4In* e de trabalhos relacionados.
- Especificação da captura de dados a partir do Facebook e IMDB visando à criação do contexto do usuário.
- Implementação da extensão do *plugin CODI4In* quanto à captura automática dos dados do perfil do usuário, seus filmes e gêneros, criando o perfil do usuário com base nestas informações contextuais.
- Realização de testes de validação e de experimentos com usuários.

5.2. Dificuldades Encontradas

Devido ao curto período de duração do estágio, não foi possível desacoplar totalmente o *plugin CODI4In* da aplicação MovieShow. Além disso, comparado ao mundo real, o banco de dados da aplicação MovieShow é pouco populoso, uma vez que existem diversos títulos dentro do panorama de filmes que ainda não estão nesta base de dados. Através de consultas na base de dados descobrimos quais atores participaram de mais filmes, ou seja, quais atores retornavam mais resultados, sendo os seguintes: “Michael (I) Madsen”, “Johnny Depp”, “Sean Astin” e “The Rock”, sendo assim, nos testes realizados instruímos os usuários que consultassem por esses nomes apresentados, uma vez que retornavam mais filmes de que participavam. Consequentemente teríamos mais categorias, facilitando assim a visualização do ranqueamento quando ativado o *plugin*.

5.3. Trabalhos Futuros

Em paralelo a esta implementação, está sendo realizado um estudo com a API Jena²⁴ e OWL API²⁵, para aprimoramento da manipulação existente da ontologia CODI-User.

Estamos planejando o desenvolvimento de uma nova aplicação web para maior abrangência e teste do *plugin*, visto que neste momento só o testamos no MovieShow. Outras aplicações serão desenvolvidas, capturando diferentes elementos contextuais através de outras redes sociais com o objetivo de prover personalização dos resultados das consultas e aprimorar o uso do *CODI4In*.

²⁴ <http://jena.sourceforge.net/ontology>

²⁵ <http://owlapi.sourceforge.net/>

Referências

- Aguiar S. (2007) “**Redes sociais na internet: desafios à pesquisa**”, Em Anais do Congresso Brasileiro de Ciências da Comunicação (Intercom), Santos, p. 1.
- An Y., Mylopoulos, J. and Borgida A. (2006) “**Building Semantic Mappings from Databases to Ontologies**”, In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06) Nectar Track, Boston, MA, 2006.
- Arruda, T., Souza, D. and Salgado, A. C. (2010) “**PSemRef: Personalized Query Reformulation based on User Preferences**”, In Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS2010), 2010, Paris. iiWAS2010 Proceedings. New York : ACM, 2010. p. 681-684.
- Dey, A. (2001) “**Understanding and Using Context**”, Personal and Ubiquitous Computing Journal, Volume 5, p. 4-7, London, UK.
- Eynard D. (2008) “**Using semantics and user participation to customize personalization**”, HP Labs Technical Report HPL-2008-197. HP Labs, Palo Alto, CA, p. 3.
- Freitas M., Silva J., Bandeira D., Mendonça A., Souza D. (2012a) “**CODI4In – Um Plugin para a Persistência de Informações Contextuais do Usuário em Ambientes de Consultas**”, Em Anais do Connepi 2012, Palmas.
- Freitas M., Silva J., Bandeira D., Mendonça A., Salgado A.C., Souza D. (2012b) “**A User Context Management Approach for Query Personalization Settings**”. In Proceedings of the 6th International Conference on Semantic Computing (ICSC), 2012, Palermo, Italy.
- Karapantelakis A. and Maguire Jr. G. (2010) “**Utilizing social context for providing personalized services to mobile users**”, In Proceedings of the EuroSSC 2010, Royal Institute of Technology, Sweden.
- Kostadinov, D., Bouzeghoub, M. and Lopes S. (2007) “**Query Rewriting based on User’s Profile Knowledge**”, In: Actes des 23emes Journées Bases de Données Avancées, Marseille, France.

Koutrika G., Ioannidis Y. (2004): “**Personalization of Queries in Database Systems**”, In Proceedings of the 20th International Conference on Data Engineering (ICDE’04), Boston, Massachusetts, USA.

Koutrika, G., Ioannidis, Y. (2005) “**Personalized Queries under a Generalized Preference Model**”, In Proceedings of the 21st Intl. Conf. On Data Engineering (ICDE), p. 841 – 852. Tokyo.

Levandowski, J., Khalefa, M.M. (2011) “**The CareDB Context and Preference-Aware Database System**”, In Proceedings of the International Workshop on Personalized Access, Profile Management, and Context Awareness in Databases. PersDB, Seattle.

Newsroom, Facebook (2012). Disponível em <http://newsroom.fb.com/content/default.aspx?NewsAreaId=22>. Acesso em junho 2012.

Santoro, F. M., Brézillon, P. and Araújo, R. M. (2006) “**Context Dynamics in Software Engineering Process**”, In Proceedings of the Computer Supported Cooperative Work in Design III, 10th International Conference (CSCWD), Nanjing, China, p. 377-388.

Souza D., Pires, C. E., Kedad, Z., Tedesco, P., Salgado, A.C. (2011) “**A Semantic-based Approach for Data Management in a P2P System**”, In LNCS Transactions on Large-Scale Data- and Knowledge-Centered Systems.

Souza, D., Belian, R., Salgado, A. C. and Tedesco P. (2008) “**Towards a Context Ontology to Enhance Data Integration Processes**”, In Proceedings of the 4th Workshop on Ontologies-based Techniques for Databases in Information Systems and Knowledge Systems (ODBIS), VLDB, p. 24-30, New Zealand, August.

Souza D. (2009) “**Using Semantics to Enhance Query Reformulation in Dynamic Distributed Environments**”, PhD Thesis, Universidade Federal de Pernambuco (UFPE), Recife.

Stefanidis, K., Drosou, M. and Pitoura, E. (2009) “**You May Also Like Results in Relational Databases**”, In Proceedings of the 3rd International Workshop on Personalized Access, Profile Management and Context Awareness in Databases (PersDB), France.

Tanca, L, Bolchini C., Quitarelli E., Schreiber F. (2011) “**Problems and Opportunities in Context-Based Personalization**”, In Proceedings of PersDL, VLDB Workshops, Seattle, WA, EUA.

Vieira, V., Souza, D., Salgado, A. C. and Tedesco, P. (2006) “**Uso e Representação de Contexto em Sistemas Computacionais**”, Tópicos em Sistemas Interativos e Colaborativos, C. A. C. Teixeira, C. R. G. de Farias, J. C. Leite and R. O. Prates, São Carlos, UFSCAR, p. 127-166.

Apêndices

Apêndice 1- Questionário de avaliação

Seu perfil

Qual o seu nome?

Sexo: *

- ☐ Masculino
☐ Feminino

Qual a sua idade? *

- ☐ 18 a 30
☐ 31 a 50
☐ 50+

Frequência em que curte filmes no Facebook: *

Ex: Curtir Os Vingadores ou Harry Potter, uma página do Facebook que esta na categoria de Filme.

- ☒ Frequentemente
☐ Algumas vezes
☐ Raramente
☐ Nunca

Consultas SEM contexto

Considerando as consultas sem o uso de contexto (sem o CODI4In), responda a pergunta a seguir:

Os resultados foram ordenados de acordo com seus gêneros preferidos? *

- ☐ Sim
☐ Não

Consultas COM contexto

Considerando as consultas com o uso de contexto (com o CODI4In), pontue os seguintes aspectos relacionados ao uso e obtenção de resultados da aplicação MOVIESHOW:

Relevância das respostas obtidas com o CODI4In *

Verifique se os resultados foram ordenados de acordo com seus gêneros preferidos.

- ☒ Excelente
☐ Bom
☐ Regular
☐ Ruim
☐ Péssimo

O uso do plugin CODI4In melhorou o resultado da consulta? *

- ☐ Sim
☐ Não

Sugestões ou reclamações:

Enviar