

Sistema de Recomendação de Serviços Baseado em uma Arquitetura Aberta para um Ecossistema de Software

Leandro Simões

Universidade Federal Juiz de Fora
Rua José Lourenço Kelmer, S/n
Juiz de Fora - MG, 36036-330
leandro@ice.ufjf.br

Fernanda Campos

Universidade Federal Juiz de Fora
Rua José Lourenço Kelmer, S/n
Juiz de Fora - MG, 36036-330
fernanda.campos@ufjf.edu.br

Victor Ströele

Universidade Federal Juiz de Fora
Rua José Lourenço Kelmer, S/n
Juiz de Fora - MG, 36036-330
victor.stroele@ice.ufjf.br

Regina Braga

Universidade Federal Juiz de Fora
Rua José Lourenço Kelmer, S/n
Juiz de Fora - MG, 36036-330
regina.braga@ufjf.edu.br

RESUMO

Esse artigo descreve uma arquitetura aberta para sistemas de recomendação, com etapas, aspectos de perfil e contexto, técnicas e métodos, que pode ser adaptada a diversos domínios. Essa abordagem permite a redução de custos, amplia a confiabilidade e funcionalidade do sistema, e agiliza o processo de desenvolvimento do software para atender requisitos específicos. A proposta foi desenvolvida a partir de um mapeamento sistemático da literatura e experiências do Grupo de Pesquisa. Para avaliar a arquitetura é apresentado um sistema de recomendação para serviços na área de e-Science, sob a abordagem de ecossistema de software. A avaliação da proposta é composta de um estudo de caso e os resultados preliminares apontam para a viabilidade do modelo.

Palavras-Chave

Sistemas de recomendação, arquitetura aberta, serviço, ecossistema

ABSTRACT

This article describes an open architecture for recommender systems, with steps, profile and context aspects, techniques and methods, that can be adapted to several domains. This approach enables cost saving, enhances system reliability and functionality, and streamlines the software development process to meet specific requirements. The proposal was developed from a systematic mapping of the literature and the Research Group experiences. To validate the proposal, a recommendation system for e-Science services is presented, under ecosystem approach. The evaluation of the proposal is composed of a case study and the preliminary results point to the feasibility of the model.

CCS Concepts

• Information systems→Recommender systems • Information systems→Information systems applications

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SBSI 2017, June 5th–8th, 2017, Lavras, Minas Gerais, Brazil.
Copyright SBC 2017.

Keywords

Recommender Systems, open architecture, service, ecosystem

1. INTRODUÇÃO

Existem diferentes estratégias para o desenvolvimento intensivo de software, uma dessas abordagens é o desenvolvimento usando arquitetura aberta [2]. Essa abordagem de reuso permite a redução de custos, amplia a confiabilidade e funcionalidade do sistema e agiliza o processo de desenvolvimento de software para atender requisitos específicos.

“Arquitetura aberta é uma técnica de customização introduzida por Oreizy, 2000 apud Alspaugh et al 2013 [11, 2] que permite que terceiros modifiquem o sistema de software através de uma arquitetura pré-definida, evoluindo o sistema através da recolocação de seus componentes”.

Um Sistema de Recomendação (SR) sugere recursos ao usuário baseados em suas preferências e expectativas [6]. Esses sistemas são capazes de recomendar produtos, serviços e objetos alinhados ao perfil e contexto do usuário ou grupo, sendo possível aplicá-los a diversos domínios, entre eles e-commerce, e-learning, turismo e eScience.

Na literatura identificamos [1, 15] diversos fatores que combinados definem uma sistema de recomendação personalizado: tipos de recursos a serem recomendados, algoritmos de filtragem, métodos, técnicas, modelos de extração e enriquecimento de perfil e contexto, desempenho e qualidade dos resultados para o processo de recomendação personalizado.

De maneira simplificada é possível dividir os sistemas de informação que utilizam recomendação em dois grupos, aqueles que utilizam a recomendação com fim, ou seja, possuem como objetivo oferecer itens possivelmente aderentes ao perfil do usuário ou grupo, e sistemas que utilizam a recomendação como apoio às atividades dos sistemas, ou seja, utilizam a recomendação como meio.

A proposta é uma arquitetura de alto nível que inclui os principais módulos e elementos de software, e que pode ser instanciada para diferentes configurações de sistemas de recomendação.

A metodologia de pesquisa incluiu o mapeamento sistemático realizado pelos autores [9], a proposta de uma arquitetura aberta e a avaliação da mesma em diferentes domínios.

Outra motivação é avançar as pesquisas relacionadas ao projeto BROAD, no desenvolvimento de ambientes de recomendação personalizada [12, 14]. Nosso foco são recomendações individuais ou em grupo, usando repositórios locais, dados ligados ou outras mídias disponíveis.

Para avaliar a arquitetura proposta esse artigo também descreve um sistema de recomendação para a escolha de serviços aderentes ao perfil do usuário de um ecossistema de software científico E-SECO [8]. O E-SECO auxilia o processo de prototipação de experimentos científicos, sendo possível utilizar workflows presentes em comunidades como o myExperiment. Sob essa ótica durante o processo de recomendação os workflows são tratados como serviços, sendo possível definir e implementar métricas de avaliação de qualidade de serviços utilizadas em web-services.

O artigo está assim organizado: os trabalhos relacionados estão na seção 2, os componentes de um modelo de recomendação e a proposta da arquitetura estão na seção 3. Na seção 4 é apresentada a proposta de recomendação para o projeto E-SECO, os parâmetros de avaliação de serviços e o modelo de recomendação implementado e a avaliação e resultados do modelo de recomendação. Por fim as considerações finais e trabalhos futuros são apresentados na seção 5.

2. TRABALHOS RELACIONADOS

No mapeamento sistemático realizado pelos autores [1, 15] foram identificados 128 artigos que reúnem, categorizam ou descrevem temas relacionados à recomendação adaptável e expansível a diferentes domínios. A quantidade crescente de publicações pode ser um indício de que ainda é um tema atual e de interesse na comunidade científica. Também foi possível perceber que aplicações em redes sociais ainda demandam muitas pesquisas na área.

Outros artigos sobre o tema foram identificados no mapeamento, entretanto, abordavam temas específicos (como partida fria) ou domínios restritos e foram descartadas por não oferecerem uma descrição expansível a múltiplos domínios. Nesse artigo destacamos dois trabalhos.

Em [3] os autores realizam uma revisão a partir da extração dos termos mais relevantes em sistemas de recomendação com base em 300 artigos publicados em periódicos e congressos. O trabalho identifica os principais modelos e algoritmos utilizados e descreve taxonomias da área. Também apresenta métricas utilizadas para avaliação da recomendação, abordando inclusive possíveis tendências na área como a integração de diferentes fontes de dados, além da utilização do conceito de *Internet of Things* e sensibilidade ao contexto.

No trabalho [10] é apresentada uma revisão para sistemas de recomendação dividida em quatro dimensões: métodos e técnicas de recomendação, softwares que utilizam recomendação, domínios de aplicação e plataformas de aplicação. O processo é feito com base na seleção de palavras-chaves, através da busca em bases digitais indexadas. O trabalho também descreve e agrupa diferentes domínios de aplicação e de recursos recomendáveis como turismo, e-learning, músicas e vídeos, documentos, redes sociais, entre outros, além de abordar a recomendação para grupos de usuários. Por fim, é apresentado um guia para aplicar e desenvolver sistemas que utilizam recomendação.

A principal diferença é que o objetivo do presente trabalho não é propor modelos de recomendação específicos e sim uma

arquitetura que possa auxiliar na criação de SRs instanciados, baseados em diferentes estratégias, domínios, restrições e interações.

3. MMRecomender – ARQUITETURA PARA SISTEMAS DE RECOMENDAÇÃO

A partir do mapeamento sistemático [1, 15] e das propostas anteriores do grupo [8, 13, 14], este artigo apresenta um modelo de arquitetura aberta com os componentes necessários para construção de Sistemas de Recomendação, denominado MMRecomender.

As vantagens de uma arquitetura aberta para SRs podem assim serem resumidas:

- capacidade de atender à flexibilidade exigida pelos requisitos dos diferentes domínios dos sistemas de recomendação;
- adoção de modelos que garantam a interoperabilidade dos diferentes componentes;
- disponibilidade de modelos, técnicas e algoritmos que permitem acesso a soluções já testadas em sistemas semelhantes;
- contribuir para que sejam desenvolvidos sistemas de recomendação de alta qualidade para os usuários finais.

3.1 Componentes de um sistema de recomendação

O processo de desenvolvimento do MMRecommender teve início com o mapeamento sistemático, anteriormente citado, para identificação de termos, características, etapas e atividades dos sistemas de recomendação.

Os principais termos identificados foram: *Collaborative Filtering*, *Content-Based*, *Demographic*, *Social* e *Hybrid* que representam técnicas de filtragem ou agrupamento dos objetos recomendados. *Trust model*, *Fuzzy model*, *Clustering*, *Probabilistic*, *Bayesian networks*, *Matrix factorization*, *Knn*, *Markov model*, *LDA Model*, *HRM*, *Heuristic*, *User-Item-Feature-based*, *CSFS*, *Random Forest*, *Genetic*, *Context-aware*, *Slope One*, *TS-PMF*, *Similarity Metrics*, *Machine Learning*, *Natural Language Processing* como métodos que relacionam o perfil e contexto do usuário com os objetos recomendados.

Também foram identificados termos relacionados à forma de extração dos dados como: *Implicit Feedback* que são dados não informados diretamente pelo usuário, como por exemplo as atividades executadas pelo usuário dentro do sistema; *Explicit Feedback* que são os dados informados diretamente ao sistema, como avaliações de conteúdo, além de *Context*, *Profile*, *User*, *Groups* e *Top-N* que podem indicar a forma com que os objetos serão recomendados, por exemplo se a recomendação será feita para os usuários de forma individual ou em grupo.

Em seguida os termos identificados foram agrupados de acordo com as funcionalidades e etapas. Nessa fase também foram observados termos similares em funcionalidades e criados subgrupos. A partir dos grupos e subgrupos formados foi gerado um primeiro modelo de representação da arquitetura.

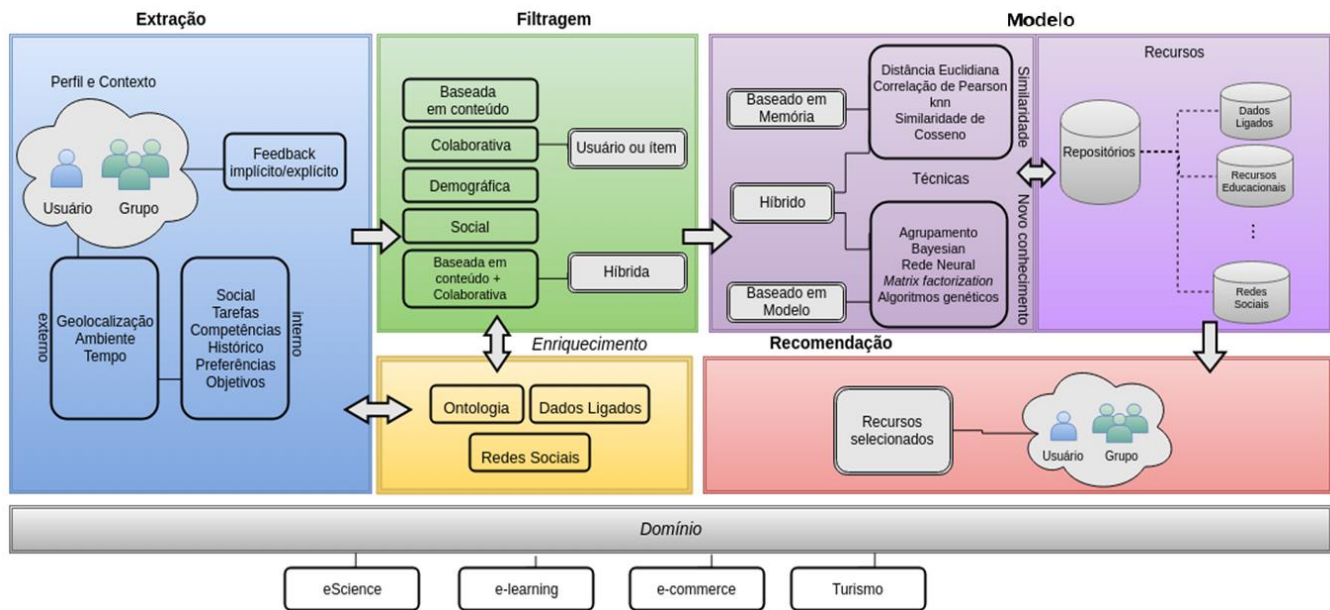


Figura 1 - Arquitetura MMRecomender.

A avaliação da primeira versão foi feita através de um estudo de caso com o BROAD-GRS, um Sistema de Recomendação para Grupos, desenvolvido no contexto do projeto BROAD no domínio educacional [5]. Esse sistema é capaz de extrair informações, definir o perfil educacional de um grupo e recomendar recursos educacionais aos seus membros, utilizando a rede social Facebook. A proposta desse sistema de recomendação evoluiu e inovou as propostas de [12] e [13] e teve suas características identificadas em [1].

3.2 Arquitetura para Sistemas de Recomendação

Para cada aplicação instanciada da arquitetura MMRecomender é necessário definir as etapas Extração, Filtragem, Modelo e Recomendação, além da sub-etapa Enriquecimento e especificar os itens que as compõem. A arquitetura está representada na Figura 1.

Extração: é a etapa onde as informações que irão compor o perfil do usuário ou do grupo serão extraídas, seja de forma implícita ou explícita. O contexto é o fator responsável pela forma de utilização do sistema e construção do perfil do usuário ou grupo. Foi utilizada a categorização proposta por [7]: interno e externo. A sub-etapa Enriquecimento é responsável pelo aprimoramento do perfil extraído através de informações contidas nas redes sociais [7], em dados ligados e ou em ontologias.

Filtragem: nessa etapa o algoritmo de filtragem é aplicado. Segundo Bobadilla et al. [3], uma categorização bem aceita na área divide os algoritmos em cinco tipos: Baseado em Conteúdo (*Content-based*) realiza as recomendações com base nas escolhas já feitas pelo usuário [4]; Colaborativa (*Collaborative*) recomenda com base nas escolhas de outros usuários com perfil e preferências similares (existem dois subtipos *user-based* e *item-based*); Demográfica (*Demographic*) que divide os usuários em grupos com preferências similares; Social baseado na recomendação em redes sociais e grupos de usuários, formados de

modo explícito ou implícito e Híbrida (*Hybrid*) que é a combinação de duas ou mais técnicas.

Modelo: é a etapa em que a estratégia de recomendação é aplicada, sendo as mais comuns: *Model-based*, que cria um modelo que define o perfil e preferências do usuário ou grupos de usuários [3]; *Memory-based*, também conhecido como *similarity-based*, podendo ser subdividido em *user-similarity* ou *item-similarity* e *Hybrid*, que é a combinação das estratégias anteriores. Os algoritmos de recomendação utilizam técnicas estatísticas, de mineração de dados, inteligência artificial, entre outras. A partir da aplicação do modelo os itens são recomendados.

Recomendação: é a etapa final do processo de recomendação, onde os recursos são selecionados para serem apresentados aos usuários ou grupo de usuários. Os SR podem adotar estratégias diferentes para apresentar as recomendações dependendo do domínio de aplicação: listas de itens, ordenação pela aderência ao perfil ou por avaliações de outros usuários, por associação entre preferências de usuários semelhantes ou por objetos recomendados com características similares. Essa etapa está diretamente relacionada aos repositórios de recursos, de onde sairão os itens que serão selecionados e apresentados aos usuários.

4. AVALIAÇÃO DA PROPOSTA NUM DE SISTEMA DE RECOMENDAÇÃO PARA ECOSSISTEMA

Workflow científico é um recurso utilizado no domínio de e-Science e está relacionado à divisão em processos sequenciais de um experimento científico. Geralmente para a implementação de um workflow são necessários conhecimentos distintos em computação e no domínio de atuação do workflow. Não raramente um experimento científico pode envolver diversas áreas do conhecimento, por isso a concepção de experimentos científicos não é uma tarefa trivial.

Nesse contexto surgiram comunidades virtuais científicas que pretendem diminuir o esforço de desenvolvimento de workflows e

aumentar os índices de reuso de workflows e colaboração entre grupos de pesquisa. Nessas comunidades cientistas podem disponibilizar publicamente workflows através da internet.

Workflows disponíveis na internet são implementados na forma de web-services, que é uma forma de serviço computacional muito utilizado em sistemas de informação. Características comuns aos web-services também estão presentes em workflows científicos, como requisições HTTP e retorno em formatos JSON e XML. Seguindo o conceito de serviço o processamento e implementação fica abstraído para o usuário [17], sendo necessário apenas saber os dados de entrada e saída do serviço.

4.1 Ecossistema E-SECO

O sistema de recomendação apresentado nesta seção foi desenvolvido e aplicado ao domínio de e-Science utilizando o ecossistema de software científico e colaborativo chamado E-SECO [8]. Este ecossistema pode realizar a prototipação de experimentos, que é a definição de um fluxo de trabalho para cada etapa necessária no experimento utilizando serviços externos provenientes de sites como myExperiment e BioCatalog . O E-SECO (Figura 2) oferece uma interface de busca e visualização de serviços e workflows externos para a composição do experimento, apresentando uma lista ordenada dos serviços de acordo com a comparação sintática do termo buscado e o nome do serviço.

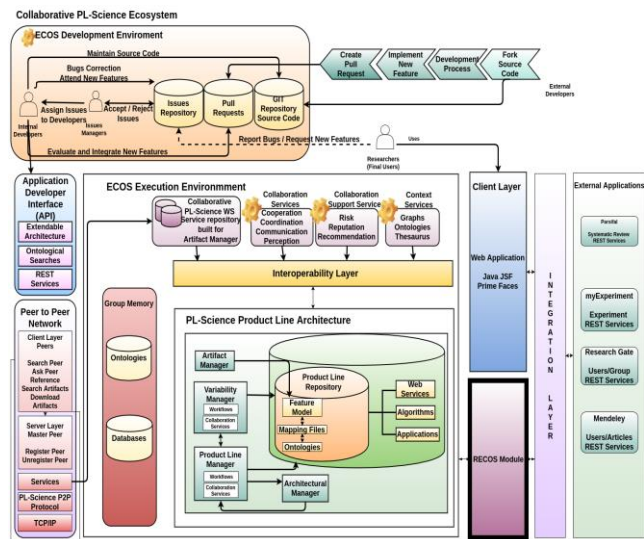


Figura 2 – Arquitetura do E-SECO [8].

O E-SECO possui uma arquitetura complexa que implementa o conceito de linha de produto de software (LPS) ao trabalhar com módulos separados que podem ser combinados na composição final do software. Também possui elementos de colaboração entre desenvolvedores, usuários e grupos de pesquisa. Seus principais módulos são:

- *Development Environment*: Responsável pelo gerenciamento e versionamento do repositório de código fonte, notificações de erros e envio de novas *features* através de *pull-requests*. Através desse ambiente é possível que desenvolvedores contribuam com a evolução do ecossistema implementando novas *features* ou artefatos que podem ser integrados ao E-SECO.

- *Execution Environment*: É o ambiente responsável pela execução dos serviços utilizados na prototipação dos experimentos.
- *External Platforms Third-Party Applications*: É a camada responsável por consultar APIs externas ao E-SECO, como por exemplo, serviços do *myExperiment* e *BioCatalogue*.

O modelo de recomendação proposto neste trabalho utiliza principalmente os módulos *Development* e *External Platforms Third-parties*.

4.2 Sistema de Recomendação E-SECO

O E-SECO pode ser caracterizado com um sistema que utiliza a recomendação como meio, e não como fim, ou seja, seu objetivo principal não é recomendar serviços científicos. O E-SECO utiliza a recomendação para auxiliar no processo de escolha de serviços durante a prototipação dos experimentos.

O modelo apresentado atua durante a busca de serviços externos ao E-SECO, desta forma o algoritmo re-ordena a lista resultante da busca com base em fatores inferidos de *rating* e *time*, que são calculados durante o processo de recomendação para cada serviço utilizado no sistema, logo, a recomendação pode ser definida como baseada no conteúdo [3].

O módulo de recomendação é implementado como um componente que pode ser integrado ao sistema e processa dados de entrada e saída através de APIs. Esta característica é importante para que o módulo de recomendação possa ser acoplado, ou não, ao sistema, mantendo as características de LPS.

A figura 3 mostra como o módulo de recomendação é integrado no E-SECO, onde é possível perceber que os dados de entrada são provenientes da API de memória de grupo, que coleta os dados de utilização (*feedback* implícito) do sistema e também da API de comunicação com serviços externos ao E-SECO.

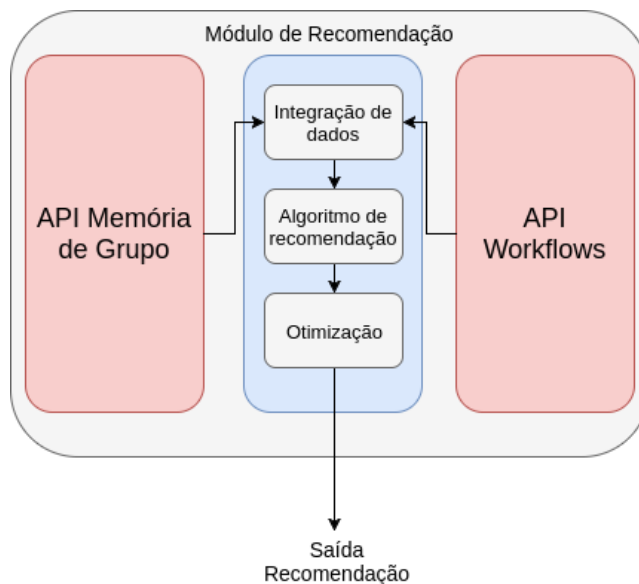


Figura 3 – Módulo de Recomendação do E-SECO.

Como workflows científicos são fornecidos na forma de web-services o termo serviço será utilizado para serviços e workflows.

4.3 Identificação de parâmetros para recomendação de Serviços Científicos

Para identificar quais fatores podem ser considerados importantes na recomendação de serviços científicos foi criado um *dataset* com dois mil serviços originados do repositório myExperiment. Em seguida foram analisadas as relações entre o tempo de vida e a quantidade de downloads a fim de identificar uma possível relação entre o tempo de existência dos serviços e a quantidade de downloads.

A quantidade de downloads do serviço foi analisada pois é possível inferir que em um repositório com muitos serviços, sendo alguns com a mesma finalidade, o serviço com uma quantidade maior de download é o serviço mais popular da comunidade, essa informação pode ser considerada como uma métrica de qualidade do serviço.

Tempo de vida

Para o cálculo do parâmetro tempo de vida foi considerada a diferença, em anos, entre a data de análise e a data de publicação no repositório, em seguida foi calculada a média do tempo de vida de todos os serviços disponíveis nos datasets. O valor resultante foi 5,8 anos. Considerando esse valor os serviços foram ordenados de forma decrescente, de acordo com a quantidade de downloads. Em seguida os duzentos primeiros (10% da amostra) foram avaliados.

Serviços com um tempo de vida maior que a média foram considerados positivos (Bom), e abaixo negativos. Os resultados da análise da amostra são apresentados na tabela 1.

Tabela 1. Comparação do tempo de vida.

Dados	
Média vida (anos)	5,78
Máx. tempo de vida (anos)	8,55
Mín. tempo de vida (anos)	3,75
Desvio Padrão (anos)	1,41
Qtd. "Bom" Tempo de vida (anos)	190

Para alcançar um grau de granularidade maior na avaliação dos serviços foi criada uma escala de pontuação com base no desvio padrão do tempo de vida dos serviços presentes na amostra. A tabela 2 apresenta essa escala criada, assim como o agrupamento dos serviços presentes em cada grupo de pontuação.

Tabela 2. Escala de pontuação com base no desvio padrão.

Escala	Pontuação	Qtd.
+2 Desvio Padrão	3	163
+1 Desvio Padrão	2	27
Média	1	8
-1 Desvio Padrão	0	2

É necessário considerar que, quanto maior o tempo de vida do serviço, maior o tempo em que ele está disponível no repositório, consequentemente maior pode ser a quantidade de downloads, por

isso as avaliações explícitas dos usuários, que são valores inteiros entre 0 e 5, foram estudadas.

Entretanto, a análise das avaliações explícitas apresentou três dificuldades, sendo a primeira a baixa quantidade de avaliações, dos 2000 workflows do dataset apenas 100 foram avaliados, com um total de 128 avaliações. A segunda dificuldade está na subjetividade das avaliações numéricas, ou seja, valores iguais podem ter pesos diferentes para cada usuário. E por último, as avaliações não estão relacionadas com a versão do serviço, ou seja, não considera as atualizações e possíveis melhorias.

A partir dos dados apresentados nas tabelas 1 e 2 é possível inferir que o tempo de vida pode ser usado como um fator positivo de qualidade de serviços no repositório myExperiment. Devido às particularidades nos dados de avaliações explícitas elas foram desconsideradas para a construção da proposta, permanecendo apenas a análise com relação ao tempo de vida e a aceitação da suposição.

Fatores de Similaridade

O fator *rating* (Equação 1) é uma propriedade presente em todos os serviços utilizados no E-SECO e mede, de forma implícita, a avaliação do usuário àquele serviço (S). O *rating* é definido pela razão entre a frequência que um serviço (Si) foi utilizado, e o total de serviços utilizados. A alteração do viés do usuário é trabalhada através deste fator, por exemplo, caso um serviço muito utilizado seja substituído por outro, o fator *rating* será menor, logo, poderá ser menos recomendado com o tempo.

$$rating = \frac{freq(si)}{total(s)} \quad (1)$$

Time é um fator que mede características de todos os serviços, como data de criação, atualização e a data mais recente em que o serviço foi incluído em um experimento no E-SECO. Para o cálculo deste fator a data atual, ou seja, a data em que a recomendação será realizada é utilizada como parâmetro. O fator *time* é definido pela soma dos fatores: *fresh* e *bias*, que medem a atualização e o viés de preferência, respectivamente mostradas nas equações 2 e 3.

O fator *fresh* pode ser considerado um fator de qualidade do serviço, já que serviços que são frequentemente atualizados podem implicar em um nível de qualidade superior em relação a serviços desatualizados [16].

O fator *bias* mede a preferência atual dos usuários em relação à escolha de serviços mais ou menos recentes, logo, se em um determinado período os usuários optarem por utilizar serviços mais novos a recomendação considera essa preferência atual, considerando a alteração do viés de preferências.

$$fresh = \frac{(today - createdAt) + (today - updatedAt)}{today - createdAt} \quad (2)$$

$$bias = \frac{latestUsed - createdAt}{today - createdAt} \quad (3)$$

4.4 Modelo de Recomendação

O modelo de recomendação inicia-se a partir da busca de serviços através do E-SECO por um usuário. Nesse momento são calculados os fatores de todos os serviços utilizados e colocados

no plano cartesiano com coordenadas que representam *rating* (ordenadas) e *time* (abscissas) para cada eixo. A figura 4 mostra um exemplo de serviços no plano cartesiano.

Em seqüência é calculado o valor médio dos serviços através da média geométrica, que equivale à tendência central entre um conjunto de serviços (Figura 5). Este ponto é usado como um modelo de preferência ou tendência de escolha do usuário.

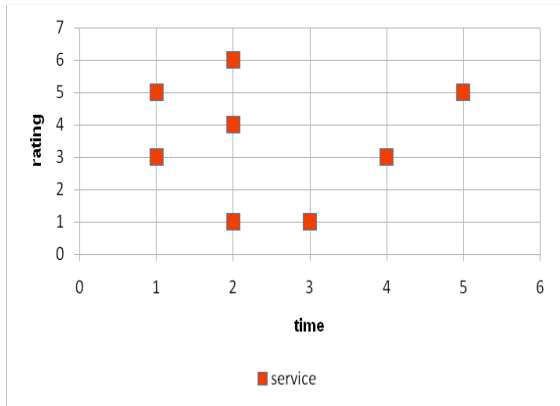


Figura 4 – Serviços utilizados no E-SECO inseridos no plano cartesiano *rating* e *time*.

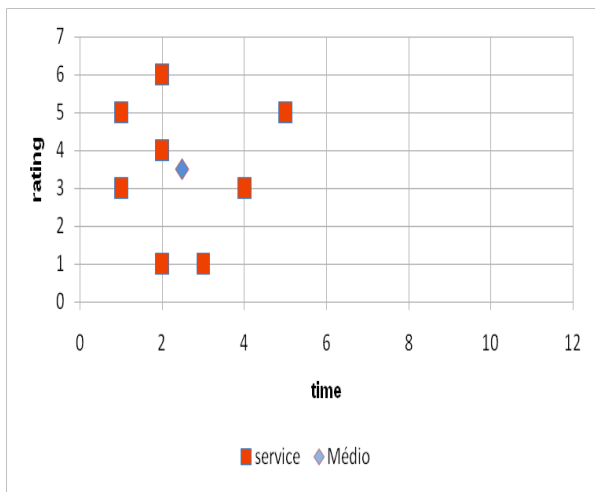


Figura 5 – Ponto geométrico dos serviços utilizados no ESECO.

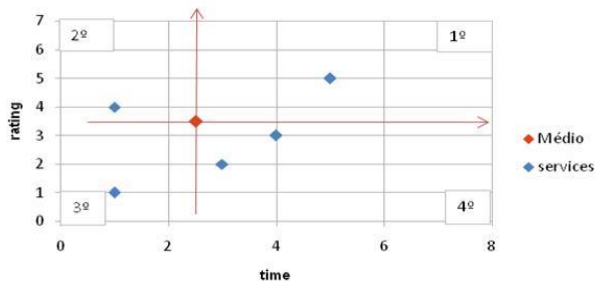


Figura 6 – Novo plano cartesiano com a média geométrica e os serviços retornados na busca.

Por fim é traçado um novo plano cartesiano com origem no ponto médio, então os serviços provenientes da busca do usuário recebem valores de *rating* e *time* e são inseridos no plano cartesiano (*rating* x *time*). A figura 6 mostra a divisão em quatro

quadrantes com origem no ponto médio e os serviços pesquisados pelo usuário.

A distância euclidiana pode ser usada como cálculo de similaridade [3], sendo que a distância entre o modelo de preferência e os serviços indicam o grau de similaridade, ou seja, quanto menor a distância maior a similaridade entre o serviço e a preferência do usuário.

Porém considerar apenas a distância euclidiana pode resultar em recomendações não aderentes ao perfil do usuário, como por exemplo, os serviços localizados no terceiro quadrante do plano cartesiano. Estes serviços possuem valores de *rating* e *time* menores que o ponto médio (preferência do usuário), por outro lado, os serviços localizados no primeiro quadrantes possuem valores maiores que o ponto médio. Sobre esse ponto de vista os quadrantes são priorizados e em seguida cada serviço localizado no quadrante é ordenado de acordo com a distância euclidiana entre o serviço e o ponto médio.

O fluxo completo para realizar a recomendação é apresentado na figura 7.

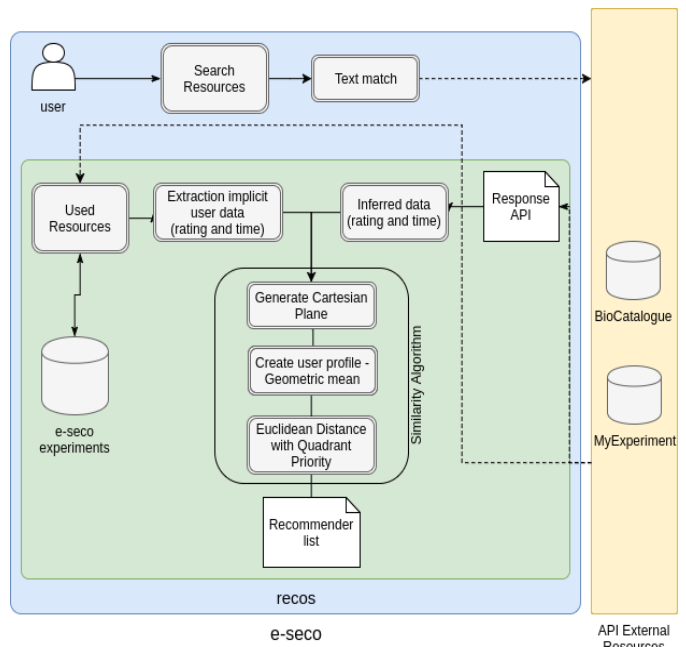


Figura 7 – Fluxo de recomendação do E-SECO.

A partir da arquitetura aberta proposta foi instanciada a arquitetura do modelo de recomendação aplicado ao E-SECO (Figura 8) com as etapas de:

Extração: Para gerar o perfil implícito os dados de prototipação dos experimentos como, serviços utilizados, data de uso e criação, são extraídos do banco de dados do E-SECO que utiliza o conceito de memória de grupo.

Filtragem: Para evitar o problema da partida-fria presente na filtragem colaborativa o modelo utiliza a filtragem baseada em conteúdo onde os valores de *rating* e *time* são gerados para cada serviço utilizado.

Modelo: Responsável por inferir os valores dos serviços retornados na busca do usuário, efetuar o cálculo de similaridade e reordenar a lista de serviços. O método de cálculo da distância euclidiana modificado foi escolhido porque é um método que

pode facilmente ser adaptado a qualquer contexto e possui baixa complexidade, resultando em um método eficiente que não compromete o desempenho do software ao efetuar o cálculo de recomendação.

Recomendação: É o processo final onde os serviços são apresentados. No E-SECO todos os serviços retornados dos repositórios são ordenados de forma decrescente de similaridade, ou seja, do serviço mais recomendado ao menos recomendado.

4.5 Avaliação

Para a avaliação do modelo de recomendação foi necessário desenvolver um estudo de caso baseado no framework sugerido por [18]. Este framework tem como objetivo a definição formal e direcionamento de todo o processo de avaliação, e pode ser escrito como:

“**Analisar** o modelo de recomendação **com o propósito de** oferecer recomendações de serviços mais aderentes ao perfil do usuário, **com relação aos serviços** utilizados com mais frequência e recentemente utilizados, **do ponto de vista** do usuário do E-SECO **no contexto de e-Science.**”

Para o estudo de caso foi criado um novo usuário e um experimento no E-SECO, em seguida alguns serviços do myExperiment foram adicionados ao experimento na etapa de prototipação. Os serviços adicionados foram escolhidos de forma aleatória utilizando como base a palavra ‘protein’ e foram identificados apenas pelo ID presente no myExperiment. Os serviços escolhidos foram: 310, 477 e 703, entre os 200 serviços com maior quantidade de downloads da amostra.

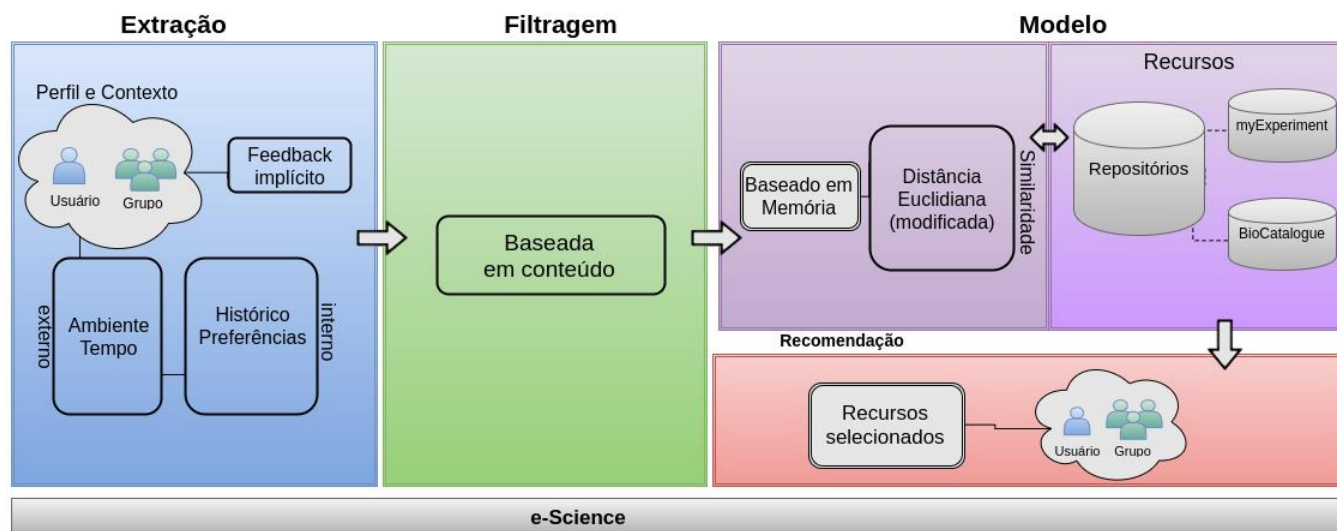


Figura 8 – Arquitetura de recomendação do E-SECO

Após a conclusão da prototipação foi realizada uma busca pelo mesmo termo utilizado na escolha aleatória dos serviços, porém com duas variações no ambiente. Na primeira variação o modelo de recomendação estava desativado, retornando a lista de serviços igual ao myExperiment. Na segunda variação foi ativado o modelo de recomendação, que realizou uma reordenação na lista de acordo com o perfil do usuário do estudo de caso.

A Tabela 3 apresenta, de modo simplificado, os dez primeiros serviços da lista de recomendação, o ID de cada serviço, sua data de criação, posição na lista e os valores de *rating* e a posição na lista sem recomendação, resultante do primeiro ambiente.

Através da tabela apresentada é possível perceber que o modelo de recomendação priorizou serviços com tempo de vida maiores, como os serviços 71, 115 e 124, que subiram mais de 15 posições na lista mesmo com valores de *rating* iguais a zero. O modelo também manteve nas posições iniciais os serviços outrora utilizados, com valores de *rating* maiores. Esse comportamento está aderente à proposta apresentada, onde os serviços que já foram utilizados e os serviços com mais tempo de vida devem ser priorizados.

Tabela 3. Comparação das listas geradas com e sem a recomendação.

Com Recomendação				
Id	Create	Rating	Posição	Posição s/ rec.
310	2008-07-12 10	4	1	1
703	2015-11-16 17	5	2	3
213	2008-06-01 12	1	3	6
71	2007-11-09 10	0	4	18
115	2007-12-10 22	0	5	19
124	2008-01-09 12	0	6	20
368	2008-07-30 16	0	7	8
216	2008-10-26 21	0	8	7
152	2008-11-27 10	0	9	4
160	2008-12-07 20	0	10	5

5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Existem múltiplas aplicações de sistemas de recomendação, o que dificulta a existência de classificações e taxonomias na área. Sob essa ótica foi proposta uma arquitetura denominada MMRecomender, para apoiar o desenvolvimento de SRs em diversos domínios.

O desenvolvimento do módulo de recomendação do E-SECO a partir da arquitetura proposta sugere que o MMRecommender pode auxiliar na concepção de uma arquitetura aderente ao contexto específico e ao sistema já existente.

O estudo de caso realizado apresenta evidências de que o modelo de recomendação implementado no E-SECO é eficiente, porém é ainda necessário realizar experimentos para avaliar se a recomendação pode auxiliar no processo de prototipação de experimentos.

Durante as etapas de prototipação e avaliação da recomendação os serviços não foram avaliados de acordo com a sua funcionalidade, devido a falta de um especialista na área de biologia. Dessa forma foi possível avaliar apenas pelos fatores de *rating* e *time*.

Durante a análise do tempo de vida dos serviços também não foi possível analisar o tempo e quantidade de atualizações. Esta informação poderia sugerir outros aspectos importantes na recomendação dos serviços, porém estes dados não são divulgados pelo myExperiment e BioCatalog.

É possível utilizar as características de colaboração do E-SECO oferecendo uma forma de compartilhamento de serviços entre grupos de pesquisa. Também é possível trabalhar temas inerentes a sistemas de recomendação como o tratamento da partida-fria utilizando um processo de enriquecimento de perfil através de redes sociais científicas.

Como trabalho futuro, pretendemos evoluir o processo de desenvolvimento da arquitetura proposta para definir uma plataforma de ecossistema de software de recomendação, explicitando os fornecedores e caminhos de evolução do mesmo.

6. AGRADECIMENTOS

Agradecimentos: à FAPEMIG, CNPq, UFJF e Capes pelo apoio financeiro ao projeto. Aos desenvolvedores do E-SECO pela colaboração no projeto.

7. REFERÊNCIAS

- [1] Almeida RF, Campos F, Stroele V (2015) Sistemas de Recomendação de Recursos Educacionais para Grupos de Redes Sociais: um Mapeamento Sistemático. Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015) 1022–1031.
- [2] Alsbaugh TA, Asuncion HU, Scacchi W (2013) Chapter 6: The challenge of heterogeneously-licensed systems in open architecture software ecosystems. In: Jansen, S.; Brinkkemper, S. e Cusumano, Softw Ecosyst Anal Manag Bus networks Softw Ind 103–120.
- [3] Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowledge-Based Syst 46:109–132.
- [4] Brunialti LF, Freire V, Peres SM, Lima CAM (2015) Aprendizado De Máquina Em Sistemas De Recomendação Baseados Em Conteúdo Textual Uma Revisão Sistemática. XI Brazilian Symp Inf Syst 203–210.
- [5] Campos F, Braga R, Nery T, Santos N (2012) Rede de Ontologias: apoio semântico a linha de produtos de objetos de aprendizagem. An do Simpósio Bras Informática na Educ 23:26–30.
- [6] Cazella SC, Reategui EB, Machado M, Barbosa JL V. (2009) Recomendação de Objetos de Aprendizagem Empregando Filtragem Colaborativa e Competências. An. do Simpósio Bras. Informática na Educação.
- [7] Fleishmann A, Bastos BR, Pernas H (2012) Sensibilidade à Situação em Sistemas Educacionais na Web. Tese de Doutorado, UFRGS.
- [8] Freitas V, David JMN, Braga R, Campos F (2015) Uma Arquitetura para Ecossistema de Software Científico. Work em Desenvolv Distrib Software, Ecossistemas Softw e Sist 41–48.
- [9] Kitchenham B, Charters S (2007) Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. Engineering.
- [10] Lu J, Wu D, Mao M, Wang W, Zhang G (2015) Recommender system application developments: A survey. Decis Support Syst 74:12–32.
- [11] Oreizy P (2000) Open architecture software: a flexible approach to decentralized software evolution, Doctoral Dissertation in University of California, Irvine, 2000. 9:161.
- [12] Pereira CK, Campos F, Ströele V, David MN, Braga R (2014) Extração de Características de Perfil e de Contexto em Redes Sociais para Recomendação de Recursos Educacionais. XXV Sbie 2014 25:506–515.
- [13] Pereira CK, Campos F, Ströele V, N JM, Braga R (2015) Explorando Dados Ligados através de um Sistema de Recomendação Educacional. An do XXVI Simpósio Bras Informática na Educ.
- [14] Rezende PAA, Pereira CK, Campos F, David JMN, Braga R (2015) PERSONNA: proposta de ontologia de contexto e perfil de alunos para recomendação de objetos de aprendizagem. Revista Brasileira de Informática na Educação 23:70.
- [15] Simões L, Almeida R, Campos F, Stroele V, David JM, Braga R, Guercio H (2016) MMRecommender: Metamodelo de Sistemas de Recomendação Aplicado a Grupos Educacionais. TISE 2016 12:505.
- [16] Tian G, Wang J, He K, Hung PCK, Sun C (2014) Time-aware Web Service Recommendations Using Implicit Feedback. EEE Int Conf Web Serv Time-aware 273–280.
- [17] Veiga W, Campos F, David JM, Braga R (2016) Uma abordagem de Ecossistemas de Software para o domínio de e-Learning. XII Brazilian Symp Inf Syst 574–581.
- [18] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Experimentation in software engineering. Experimentation in Software Engineering. <https://doi.org/10.1007/978-3-642-29044-2>. Springer