

RecS-DL



LINCAMM



Instituto de
Computação

**Plataforma de Serviços de Recomendação
para Bibliotecas Digitais**

**Instituto de Computação
Universidade Estadual de Campinas**

Índice

Introdução	3
Objetivos	3
Visão Geral	3
Arquitetura da Plataforma	4
Instalação da Plataforma	4
Pré-requisitos	4
Instalação	5
Instalação da Interface Web	6
Ferramenta de Instalação	6
Ferramenta de Controle de Execução	7
A Plataforma RecS-DL	8
WebService	8
Interface	9
Bibliotecas Digitais	10
Criação de uma biblioteca digital	10
Configuração uma biblioteca digital existente	11
Manutenção de Dados	12
Importação de Dados	12
Inserção de Dados	12
Engines de Recomendação	13
Criação de um engine de recomendação	13
Instalação de um engine de recomendação	14
Ativação de engine de recomendação	15
Configuração de engine de recomendação	15
Recomendação	16
Atualização de modelos de recomendação	16
Gerando recomendações por Item	16
Gerando recomendações por Usuário	17
Anexo I – WebServices - Arquivo services.xml	18
Anexo II – Comandos de criação do Banco de Dados	20

Introdução

Objetivos

Dado o crescimento vertiginoso apresentado pelas Bibliotecas Digitais, seja quanto ao volume de dados gerenciados, seja em abrangência de domínios de utilização, ferramentas para a localização e escolha do conteúdo desejado tornaram-se cada vez mais relevantes. Nesse sentido, as técnicas de recomendação, hoje já bastante difundidas e utilizadas, procuram reduzir ou amenizar os efeitos dessa sobrecarga de informações.

Há um grande número de ferramentas de recomendação disponíveis para utilização, diferentes sobre os mais diversos aspectos. Todavia, vários aspectos estruturais das ferramentas disponíveis geralmente limitam seu horizonte de aplicação e reuso, como:

- **Domínio de aplicação:** as ferramentas são projetadas tendo em vista uma Biblioteca Digital ou domínio específico, impossibilitando ou limitando sua reutilização;
- **Técnicas de recomendação:** os pacotes disponíveis são, em sua maioria, sistemas monolíticos. Os serviços oferecidos não são extensíveis nem configuráveis e geralmente baseiam-se em uma única técnica de recomendação;
- **Tecnologia:** a implementação das ferramentas é realizada em linguagem, plataforma ou interface específica, limitando o acesso de aplicações clientes.

A Plataforma RecS-DL procura realizar uma abordagem flexível o bastante para tornar-se independente de domínio de aplicação, de tecnologias utilizadas e extensível sob o ponto de vista de técnicas de recomendação.

Visão Geral

A **independência de domínio de aplicação** foi conseguida por meio da utilização de termos genéricos e abrangentes para designar as informações armazenadas, utilizando o padrão de metadados Dublin Core. Dessa forma, um item da Biblioteca Digital pode ser um livro, uma tese ou um filme. A plataforma é capaz de armazenar, além da descrição dos itens, conteúdo textual e binário, codificado em base64. Assim, pode-se armazenar imagens, vídeos, etc.

A **extensibilidade da plataforma**, que possibilita a utilização de técnicas de recomendação diversas foi dada por meio de uma estrutura de *engines*, semelhante ao conceito de *plugins*. Os *engines* podem ser desenvolvidos e instalados de maneira uniforme na plataforma, mas com a possibilidade de utilização técnicas, ou até mesmo dados (textuais ou binários) distintos.

Sob o ponto de vista tecnológico, utilizou-se WebServices. Assim, os Serviços de Recomendação tornam-se portáteis e acessíveis a partir de linguagens e sistemas operacionais distintos. Foi considerada também a independência em relação a servidores de aplicação e banco de dados.

Arquitetura da Plataforma

Definida a interface, o modelo arquitetural proposto divide a plataforma internamente em dois grandes módulos: Módulo de Aquisição e Engines de Recomendação, ilustrados respectivamente nas partes (B) e (D) da figura a seguir.

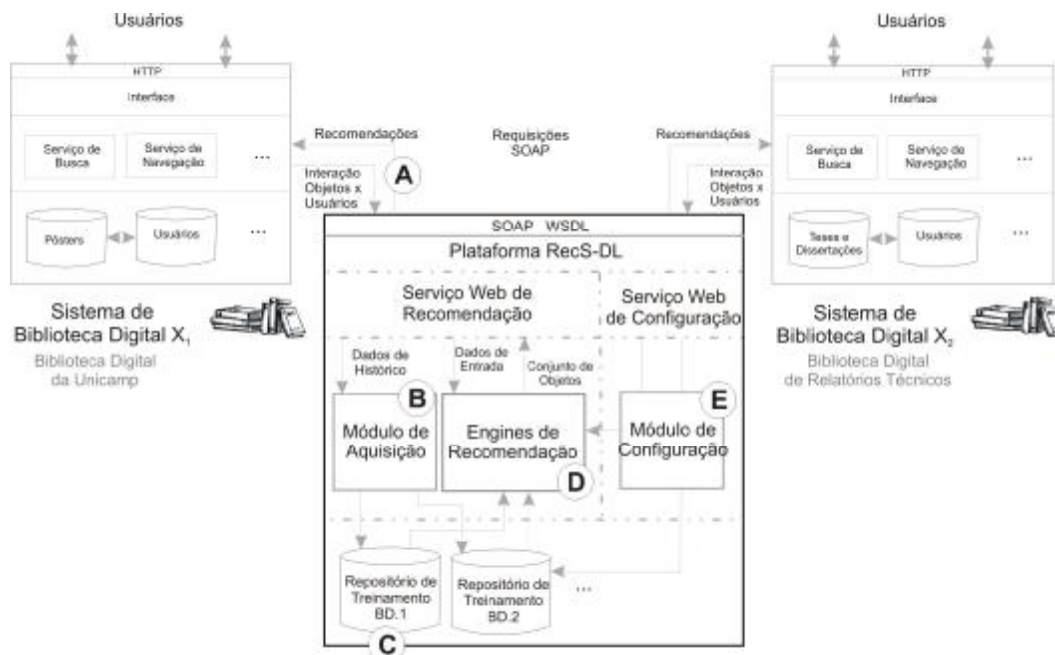


Figura 1. Arquitetura da Plataforma

Vale salientar que, a plataforma RecS-DL, instalada num determinado servidor pode prover serviços de recomendação para diversas bibliotecas digitais distintas. Além disso, os engines de recomendação (D) podem ser instalados com a plataforma em funcionamento.

Visando a facilidade de operação da plataforma, foi criado também um Webservice para configuração de engines, bibliotecas digitais e da plataforma de maneira geral.

Instalação da Plataforma

Pré-requisitos

São pré-requisitos para a instalação da plataforma:

- Java JDK – J2SE
- Servidor de aplicação Java
(foram testados os servidores Apache Tomcat e JBoss)
- Servidor de Banco de Dados
(foram testados os servidores PostgreSQL e MySQL)

- Apache Axis 2 (instalado no servidor de aplicação)
- Apache Ant
- Variável de ambiente JAVA_HOME definida para o JDK

São pré-requisitos para a instalação da interface web da plataforma:

- Apache
- PHP

Instalação

A instalação da plataforma consiste em duas etapas:

1. Instalação dos softwares servidores;
2. Instalação da plataforma, propriamente dita.

Após a instalação dos softwares servidores, a instalação da plataforma dá-se pelos seguintes passos:

1. Download

Baixar a distribuição compactada (.zip) da plataforma.

2. Arquivos de Propriedades

- Configurar o arquivo `recommenderws-config.properties`, que encontra-se na raiz do arquivo .zip. A seguir é ilustrado um exemplo desse arquivo configurado, onde a plataforma foi configurada em um sistema Windows (execução do arquivo de lote `ant.bat`), descompactada para o diretório `D:/RecommenderKit/RecommenderWS` e configurada para um banco de dados PostgreSQL.

```
#Pasta do RecommenderWS
dir.recommender-ws=D:/RecommenderKit/RecommenderWS

#Pasta de deploy de WebServices do Apache Axis 2.0
dir.webservices.apache-axis=D:/RecommenderKit/Servers/apache-tomcat-
5.5.23/webapps/axis2/WEB-INF/services

#Autenticação para Instalação de Engines
admin.user=recadmin
admin.passwrod=icpass

#Banco de Dados
sqlDriver=org.postgresql.Driver
dbUrl=jdbc:postgresql://127.0.0.1/postgres
dbUser=postgres
dbPassword=passrecsdl
logfile=dbrecommender.log.txt
ratingTable=rating

#Pasta de instalação do Ant
ant.home=D:/RecommenderKit/Servers/apache-ant-1.7.0/bin
ant.run=ant.bat
```

- Configurar o arquivo `/nbproject/private/private.properties`, para que as bibliotecas (.jar) apontem para o local onde a plataforma está instalada.

3. Build da Plataforma

Executar o script `Ant build.xml`, localizado na raiz da plataforma. Para isso basta executar o comando “ant”, com o software Apache Ant definido na variável Path.

Esse script irá compilar a plataforma e copiar a sua distribuição (.jar) para a pasta “services” do Apache Axis, localizado no servidor de aplicação configurado no arquivo de propriedades.

4. Iniciando

Ao término desse processo, basta instanciar o servidor de aplicação e o Webservice estará ativo.

Instalação da Interface Web

A instalação da interface web requer o servidor Apache com PHP ativo. Feito isso basta descompactar a distribuição da interface para o diretório Htdocs do Apache e verificar se o arquivo /includes/RecommenderConstants.class.php está corretamente configurado. Esse arquivo contém a URL do Webservice da plataforma.

Ferramenta de Instalação

Como pode ser verificado nas seções anteriores, os procedimentos que demandam maiores esforços durante a instalação da plataforma são aqueles que se referem à instalação e configuração dos softwares servidores.

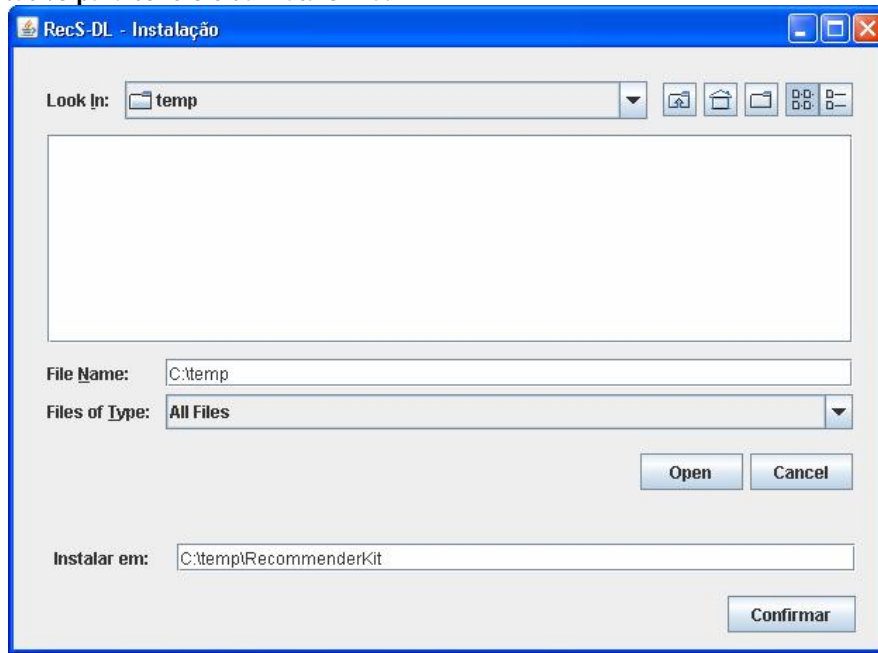
Em vista disso, foi criada uma ferramenta de instalação, em cuja distribuição estão contidos todos os softwares servidores pré-configurados. Assim, para instalar a plataforma através da Ferramenta de Instalação, os únicos pré-requisitos são:

- Java JDK – J2SE
- Variável de ambiente JAVA_HOME definida para o JDK

Para executar a ferramenta, basta “java -jar RecommenderInstall.jar” ou simples clique duplo sobre o arquivo jar em alguns ambientes. (a ferramenta de instalação ainda é específica para o SO Windows, mas há pretensões de criar uma versão Unix).



A ferramenta de instalação questiona apenas sobre o local onde deverá ser instalada. Feito isso, é realizada a instalação e configuração dos softwares servidores, da plataforma e de um aplicativo para controle da Plataforma.



Os softwares instalados e a estrutura de arquivos criada é ilustrada a seguir. A pasta RecommenderWS corresponde à plataforma e a pasta RecommenderWSCLI à interface Web. A pasta Servers contém os servidores e respectivas versões necessários à plataforma e à interface.

- + RecmenderWS
- + RecmenderWSCLI
- Servers
 - + apache-2.2.4
 - + apache-ant-1.7.0
 - + apache-tomcat-5.5.23
 - + php-5.2.3
 - + postgresql-8.2.4-1

* **Atenção para permissões de usuários:** em sistemas Win2000, o Apache deve ser instalado por um usuário com as devidas permissões.

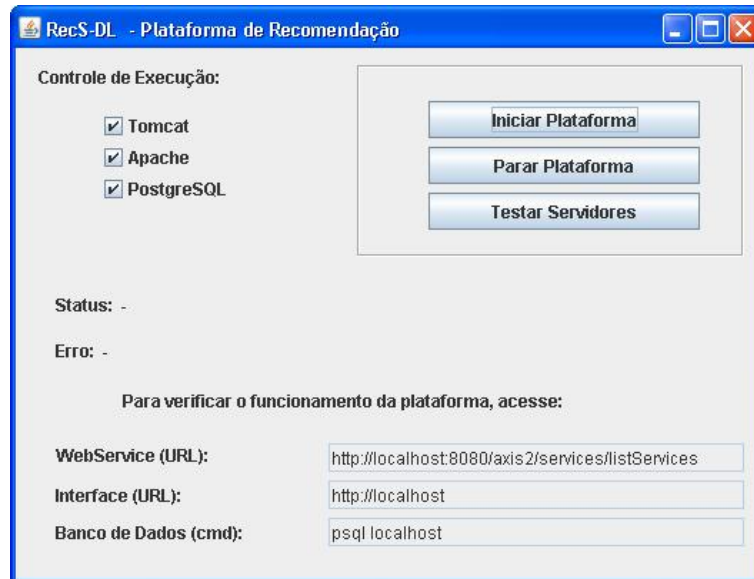
** **Atenção para outros Servidores:** caso existam outros servidores instalados na máquina (nas portas 80 – Apache, 8080 – Tomcat e 5432 – PostgreSQL), podem ocorrer conflitos durante a execução da Plataforma. Para evitar tais problemas, pare quaisquer servidores nas portas citadas antes da instalação.

Ferramenta de Controle de Execução

Completada a instalação, é necessário instanciar os servidores para testar e utilizar a plataforma. Embora seja uma tarefa simples, exige a manipulação de vários comandos e para cada servidor.

Em vista disso, foi criada uma ferramenta, ilustrada a seguir, visando fornecer uma interface gráfica para controle de execução da plataforma. A ferramenta permite que se pare ou instancie de forma visual todos os softwares servidores instalados. A ferramenta permite também testar se os servidores estão efetivamente ativos. Vale lembrar que erros durante a instalação da plataforma podem causar a não criação dessa ferramenta no diretório esperado.

* **Atenção para permissões de usuários:** em sistemas Win2000, o PostgreSQL deve ser instanciado por um usuário não administrador. Para tal procedimento pode-se executar o script runAsPostgresW2K.bat que deve ser previamente editado e configurado. As informações que devem ser configuradas nesse arquivo são: o nome da máquina onde será executado e um usuário não administrador nessa máquina.



A Plataforma RecS-DL

WebService

A plataforma utiliza a biblioteca Apache Axis 2 para implementação do Webservice. Ao instanciar o servidor de aplicação, o Webservice está pronto para ser executado. A figura seguinte exibe uma interface do Apache Axis, listando os serviços e respectivos métodos disponíveis. Ao clicar no link do serviço, o WDSL que o define é exibido.



O funcionamento dos métodos listados serão abordados a seguir. Definições mais detalhadas de métodos e parâmetros, contida no arquivo services.xml pode ser encontrada no Anexo 1 deste documento.

Interface

A interface web distribuída com a plataforma consiste em uma aplicação PHP, que opera como cliente dos WebServices disponíveis. A interface possibilita um acesso visual e rápido aos serviços oferecidos pela plataforma.

Assim, há duas opções ao utilizar a plataforma: (i) os softwares de bibliotecas digitais podem acessar diretamente os WebServices e; (ii) um usuário final, operando para uma determinada biblioteca digital pode utilizar e configurar seus serviços através da interface oferecida.

A figura abaixo ilustra a tela inicial da interface, que possibilita o acesso aos seus dois grandes módulos: (i) RecommenderWS: serviço de recomendação e; (ii) RecommenderWSConfig que opera a configuração da plataforma.

RecommenderWS: Serviço de Recomendação e Configuração

■ [RecommenderWS](#)

Interface Web do Serviço de Recomendação.
Módulos de Aquisição e Recomendação.

■ [RecommenderWSConfig](#)

Configuração da Plataforma de Recomendação.
Criação e configuração de bibliotecas digitais, configuração de engines.



* Vale lembrar que todas as operações realizadas sobre a interface RecommenderWS referem-se a uma "Biblioteca Digital ativa". A ativação pode ser realizada na página superior de sse módulo

Bibliotecas Digitais

Criação de uma biblioteca digital

A criação de uma biblioteca digital na plataforma consiste basicamente em criar um espaço de armazenamento de dados. Durante o processo, a plataforma cria um banco de dados no servidor configurado para a plataforma, assim como um espaço no sistema de arquivos.

O banco de dados criado será utilizado para armazenar as informações da biblioteca digital, como itens, usuários, ratings e configuração de engines de recomendação. As informações dos engines podem ser armazenadas tanto no banco de dados quanto no sistema de arquivos, dependendo das configurações definidas para os engines.

RecommenderWSConfig

The screenshot displays the 'RecommenderWSConfig' application interface, divided into three main sections:

- Instalação & Configuração de Engines no Webservice:**
 - Instalar novo Engine:** Fields for 'Nome do Engine', 'Arquivo de Biblioteca (.jar)', and 'Descrição (.xml)', with 'Arquivo...' buttons and 'Instalar'/'Limpar' buttons.
 - Engines Instalados:** A table listing installed engines:

Engine Name	Details	Parameters
ICHybridRecommender	Detalhes	Parâmetros
ICLuceneRecommender	Detalhes	Parâmetros
ICMultilensRecommender	Detalhes	Parâmetros
- Configuração de Engines nas Bibliotecas Digitais:**
 - Ativar & Configurar Engine em Bib. Digital:** Fields for 'Nome da Biblioteca Digital' and 'Nome do Engine', with 'Ativar' and 'Configurar' buttons.
 - Configurar Engines em Bib. Digital:** 'Nome da Biblioteca Digital' field, 'Listar Engines Ativos', and 'Consultar Engine Default' buttons.
- Criação e Configuração de Bibliotecas Digitais:**
 - Criar Biblioteca Digital (utilizando BD padrão):** 'Nome da Biblioteca Digital' field (value: bibdig1) and 'Criar' button.
 - Configurar Biblioteca Digital (utilizando BD existente):** Fields for 'Nome da Biblioteca', 'Driver de BD', 'URL JDBC', 'Usuário', and 'Senha', with a 'Configurar' button.

At the bottom, there is a logo and text: '2006 - 2007 Universidade Estadual de Campinas, Cidade Universitária "Zelmaro Vaz" - Br. Geraldo - Campinas/SP'.

Configuração de uma biblioteca digital existente

Além da possibilidade de utilizar o servidor de banco de dados pré-configurado, é possível também configurar a plataforma para acessar outros servidores de banco de dados.

Para isso será necessário criar uma estrutura de tabelas (ou views) conforme a definida no Anexo 2 deste documento. Após a criação das estruturas será necessário configurar as informações de conexão ao banco de dados: Driver JDBC, URL de conexão JDBC, usuário, senha e um identificador para a criação da biblioteca digital. Um exemplo de configuração é ilustrado na figura abaixo.

This screenshot shows the 'Configurar Biblioteca Digital (utilizando BD existente)' section of the application, with a large circle highlighting the configuration fields:

- Nome da Biblioteca:** bibdig2
- Driver de BD:** com.mysql.jdbc.D
- URL JDBC:** jdbc:mysql://loca
- Usuário:** root
- Senha:** passrecdl

The 'Configurar' button is also visible next to the fields.

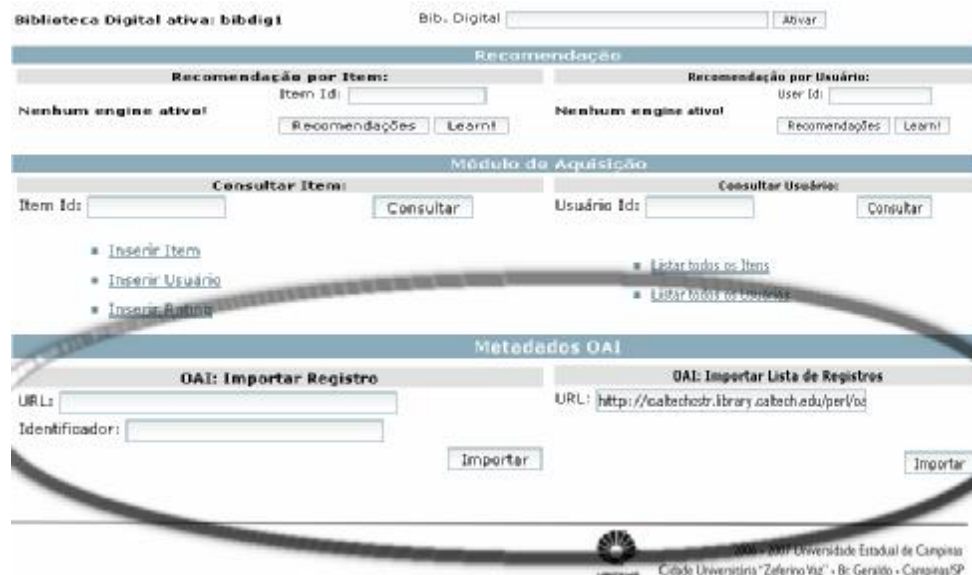
At the bottom, there is a logo and text: '2006 - 2007 Universidade Estadual de Campinas, Cidade Universitária "Zelmaro Vaz" - Br. Geraldo - Campinas/SP'.

Manutenção de Dados

Importação de Dados

A Plataforma RecS-DL dispõe de um módulo para importação de metadados por meio do protocolo OAI (Open Archives Initiative). Dada uma Biblioteca Digital previamente criada (ou configurada) na plataforma, este recurso pode ser utilizado para: (i) importar um registro ou; (ii) importar um conjunto completo de metadados de uma biblioteca digital que seja compatível com esse protocolo.

O processo de importação para grandes quantidades de registros é realizado em segundo plano no servidor de aplicações, já que pode levar vários minutos para sua conclusão. Dessa forma, o término da requisição http da interface web da plataforma não implica que o processo de importação também tenha terminado.



Inserção de Dados

Além das possibilidades de acesso à informações já existentes, como configuração de banco de dados e importação de dados OAI, o módulo de aquisição da plataforma dispõe também de funcionalidades para inserção pontual de dados.

Dessa forma, uma biblioteca digital pode ser inserida sob demanda na plataforma. E essa carga de dados pode ser realizada de duas maneiras: através do software da biblioteca digital que pode acessar diretamente os WebServices da plataforma ou através da interface disponibilizada.

A figura abaixo ilustra uma das telas que permite a inserção de dados. Pode-se verificar que é possível inserir, tanto metadados, quanto conteúdo propriamente dito, seja ele textual ou binário.

Insert Item

Titulo:	
Item Id:	<input type="text"/>
Identificador:	<input type="text"/>
Título:	<input type="text"/>
Autor:	<input type="text"/>
Descrição:	<input type="text"/>
Assunto:	<input type="text"/>
Relações:	<input type="text"/>
Linguagens:	<input type="text"/>
Tipos:	<input type="text"/>
Formato:	<input type="text"/>
...	
Conteúdo Textual:	<input type="text"/>
Conteúdo Binário:	<input type="text"/> <input type="button" value="Carregar"/>
<input type="button" value="Inserir"/> <input type="button" value="Limpar"/>	

Engines de Recomendação

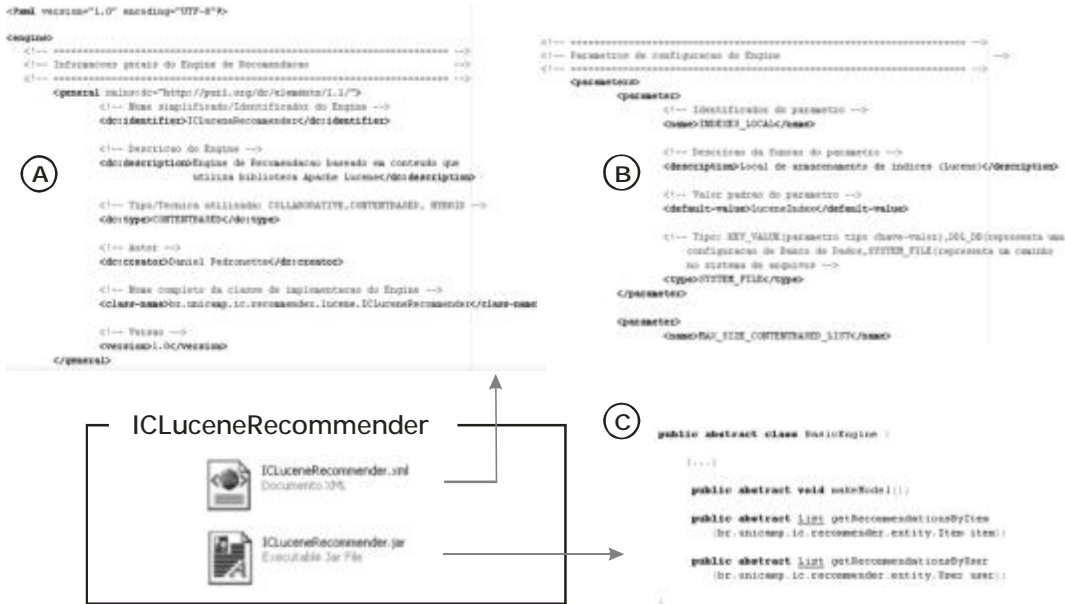
Criação de um engine de recomendação

Os engines consistem em mecanismos de recomendação, desenvolvidos segundo padrões e interfaces definidas pela plataforma. O funcionamento desses elementos assemelha-se ao conceito de plugin, nos quais módulos de software podem ser desenvolvidos e facilmente instalados sobre uma aplicação principal.

Tal conceito torna a plataforma extensível sem, no entanto, afetar a forma de comunicação e requisição de serviços junto às Bibliotecas Digitais clientes.

Os engines de recomendação são compostos basicamente por dois elementos: um pacote de implementação e um conjunto de metadados. A figura a seguir ilustra os elementos que compõem um engine, exibindo parte de um arquivo de metadados e os métodos abstratos que devem ser implementados pela classe de implementação principal.

* Maiores detalhes para a construção de engines são encontradas no documento "Criação de Engines".



Instalação de um engine de recomendação

A partir da implementação de um engine de recomendação, dada por um módulo Java empacotado (.jar) e um descritor de metadados em formato XML, esse engine deve ser instalado na plataforma.

A instalação pode ser feita pela própria plataforma, de duas maneiras: (i) acesso direto ao Webservice de configuração, por meio de um software, onde devem ser submetidos os arquivos .jar e .xml, codificados em base64 ou; (ii) o engine também pode ser instalado pela interface da plataforma, apenas indicando a localização dos arquivos em disco.

O administrador da plataforma pode realizar a instalação manipulando diretamente os arquivos que compõem o engine no diretório onde está instalada a plataforma. Há um diretório para os arquivos .jar, outro para os descritores .xml e um terceiro para as bibliotecas utilizadas pelos engines. Após a cópia dos arquivos deve-se refazer o build da plataforma, por meio do script ant que se encontra no diretório raiz.

A figura abaixo exibe o processo de instalação de um engine através da plataforma e a árvore de diretórios onde são gravados os arquivos dos engines na plataforma.

RecommenderWSConfig

Instalação & Configuração de Engines no Webservice

Instalar novo Engine:

Nome do Engine:

Arquivo de Biblioteca (.jar):

Descritor (.xml):

- [-] RecommenderWS
 - [+] build
 - [+] dist
 - [+] dl-files
 - [-] engines
 - [+] descriptors
 - [+] descriptors-dtd
 - [+] engine-impl
 - [+] engine-jars
 - [+] engine-libs
 - [+] lib
 - [+] nbproject
 - [+] src
 - [+] test

Ativação de engine de recomendação

Para a utilização efetiva de um engine de recomendação é necessário, além da instalação na plataforma, a ativação desse engine na biblioteca digital que deseja utilizá-lo.

O processo de ativação refere-se à cópia das informações e parâmetros do engine selecionado para a biblioteca digital vigente, tornando-o apto para a utilização. A relevância desse procedimento para a plataforma, consiste na possibilidade de customização de cada engine, de acordo com as necessidades e preferências de uma determinada biblioteca digital.

Configuração de Engines nas Bibliotecas Digitais

Ativar & Configurar Engine em Bib. Digital:

Nome do Biblioteca Digital:

Nome do Engine:

Configuração de engine de recomendação

Após a ativação de um engine em uma biblioteca digital, ele pode ser configurado e customizado de acordo com as necessidades dessa biblioteca digital. Cada engine, apresenta seus próprios parâmetros de configuração, que são divididos em três categorias:

- (i) **KEY_VALUE**: indicam apenas um valor armazenado ;
- (ii) **SYSTEM_FILE**: indicam um local no sistema de arquivos e;
- (iii) **DDL_DB**: armazenam um comando DDL de criação de estruturas de banco, a ser executado no momento da ativação do engine.

Dessas três categorias, apenas a primeira tem efeito ao ser customizada após a ativação. A figura a seguir exibe um exemplo de configuração do engine distribuído com a plataforma ICLuceneRecommender.

Biblioteca Digital: bibdig1; Engine: ICLuceneRecommender

Parâmetros

Parâmetro:	INDEXES_LOCAL
Descrição:	
Tipo:	SYSTEM_FILE
Valor:	<input type="text" value="LuceneIndex"/> <input type="button" value="Alterar"/>
Parâmetro:	MAX_ITEMS_RATED_BY_USER
Descrição:	
Tipo:	KEY_VALUE
Valor:	<input type="text" value="5"/> <input type="button" value="Alterar"/>
Parâmetro:	MAX_SIZE_CONTENTBASED_LIST
Descrição:	
Tipo:	KEY_VALUE
Valor:	<input type="text" value="5"/> <input type="button" value="Alterar"/>
Parâmetro:	MIN_DOCUMENT_FREQ_CONTENT_BY_ITEM
Descrição:	
Tipo:	KEY_VALUE
Valor:	<input type="text" value="2"/> <input type="button" value="Alterar"/>

...

Recomendação

Atualização de modelos de recomendação

Os engines geram recomendações baseados em alguma forma de “aprendizado” armazenado em modelos. Esses modelos variam de acordo com o engine, e pode representar a criação ou atualização de índices, o cálculo de similaridade entre itens ou usuários.

De qualquer forma, para que os engines criem recomendações com desempenho satisfatório é necessário manter os modelos atualizados. Isso pode ser feito através de chamada ao método *makeModel()* do Webservice RecommenderWS ou através da função *Learn* na interface oferecida pela plataforma.

Gerando recomendações por Item

As recomendações por item utilizam técnicas de recuperação baseada no conteúdo para geração de recomendações. Tais técnicas avaliam a similaridade entre itens, por meio do conteúdo de cada item. O conteúdo analisado pode ser textual ou binário (imagens, vídeos, etc), dependendo das funcionalidades do engine.

A figura abaixo apresenta um detalhe de item da biblioteca digital de filmes MovieLens. Apresenta também os resultados da requisição de recomendação por item realizada ao engine ICLuceneRecommender, que trabalha com recuperação de informações em conteúdo textual. Pode-se perceber que as recomendações são significativamente correlacionadas quanto ao conteúdo, em relação ao item analisado.

Consulta Item

Item:

Item Id: 29

Identificador: oai_dc:29

Título: Batman Forever (1995)

Autor:

Descrição: The Dark Knight of Gotham City confronts a dastardly duo: Two-Face and the Riddler. Formerly District Attorney Harvey Dent, Two-Face incorrectly believes Batman caused the courtroom accident which left him disfigured on one side; he has unleashed a reign of terror on the good people of Gotham. Edward Nygma, computer-genius and former employee of millionaire Bruce Wayne, is out to get the philanthropist; as The Riddler he perfects a device for draining information from all the brains in Gotham, including Bruce Wayne's knowledge of his other identity. Batman/Wayne is/are the love focus of Dr. Chase Meridian. Former circus acrobat Dick Grayson, his family killed by Two-Face, becomes Wayne's ward and Batman's new partner Robin the Boy Wonder.

Assunto:

Relações:

Linguagem:



Recomendação por Item:

Item Id:

ICMultilensRecommender

ICLuceneRecommender

Recomendações:

Item Id	Identificador	Título	Autor	Detalhes
231	oai_dc:231	Batman Returns (1992)		Detalhes
403	oai_dc:403	Batman (1989)		Detalhes
254	oai_dc:254	Batman & Robin (1997)		Detalhes
1184	oai_dc:1184	Endless Summer 2, The (1994)		Detalhes
392	oai_dc:392	Man Without a Face, The (1993)		Detalhes
1393	oai_dc:1393	Stag (1997)		Detalhes

Gerando recomendações por Usuário

As recomendações por usuário utilizam técnicas colaborativas para geração de recomendações. Tais técnicas necessitam de dados de histórico de usuário, que relacionam usuário e item. Esses dados podem ser do tipo binário, indicando uma visualização de um determinado item ou, ainda podem indicar uma avaliação numérica de um item por um usuário.

A figura abaixo ilustra a interface da plataforma exibindo um conjunto de itens avaliados por um determinado usuário, e um exemplo de requisição de recomendação, realizada ao engine ICMultilensRecommender.

Itens em Onda em de Rating (User Id: 100)

Item Id	Identificador	Título	Autor	Detalhes
515	oai_dc:515	Apocalypse (2008)		Detalhes
219	oai_dc:219	El Mismo Día (1971)		Detalhes
268	oai_dc:268	Uchaco (2007)		Detalhes
272	oai_dc:272	Grand Prix (1964)		Detalhes
294	oai_dc:294	Die Witz (1957)		Detalhes
370	oai_dc:370	Big Fish (2003)		Detalhes
328	oai_dc:328	Dispirado (1997)		Detalhes
147	oai_dc:147	Mac the Dog (2007)		Detalhes
870	oai_dc:870	Seven Years in Tibet (1997)		Detalhes
752	oai_dc:752	Turmeric Paper Dais (1997)		Detalhes
179	oai_dc:179	Headmaster, The (1997)		Detalhes
878	oai_dc:878	Postman, The (1997)		Detalhes
900	oai_dc:900	Kur Jun (1997)		Detalhes
1231	oai_dc:1231	Big Bang Theory, The (1994)		Detalhes
849	oai_dc:849	Hard Rain (1998)		Detalhes
678	oai_dc:678	Volcano (1997)		Detalhes



Recomendação por Usuário:

User Id:

ICMultilensRecommender

ICLuceneRecommender

Recomendações:

Item Id	Identificador	Título	Autor	Detalhes
28	oai_dc:28	Apocalypse (1995)		Detalhes
115	oai_dc:115	Spidey and the (1990)		Detalhes
507	oai_dc:507	Frank (1994)		Detalhes
1226	oai_dc:1226	Multiple Man (1997)		Detalhes
901	oai_dc:901	Frank (1994)		Detalhes

Anexo I – WebServices - Arquivo services.xml

Este anexo apresenta uma trecho do arquivo services.xml, que descreve os métodos disponíveis nos WebServices da plataforma.

```

=<serviceGroup>
=<service name="RecommenderWS">
<description>Serviço Web de Recomendação Desenvolvido no Instituto de Computação (IC) da
  Universidade Estadual de Campinas (Unicamp)</description>
<parameter name="ServiceClass"
  locked="false">br.unicamp.ic.recommender.webservice.RecommenderWS</parameter>
- <!-- =====>
- <!-- Métodos de Acesso a Dados -->
- <!-- =====>
- <operation name="insertUser">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="user" locked="false" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="insertItem">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="item" locked="false" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="insertSerialItem">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="item" locked="false" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="insertRating">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="rating" locked="false" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="getUserById">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="userId" locked="false" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="getAllUsers">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="getItemById">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="itemId" locked="false" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="getAllItems">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="getRatedItems">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="userId" locked="false" />
<parameter name="dbName" locked="false" />
</operation>
- <!-- =====>
- <!-- Métodos de Acesso aos Engines -->
- <!-- =====>
=<operation name="makeModel">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="engineClassName" locked="false" />

```

```

<parameter name="dbName" locked="false" />
  </operation>
- <operation name="getRecommendationByUser">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="user" locked="false" />
  <parameter name="dbName" locked="false" />
  <parameter name="engineClassName" locked="false" />
  </operation>
- <operation name="getRecommendationByItem">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="item" locked="false" />
  <parameter name="dbName" locked="false" />
  <parameter name="engineClassName" locked="false" />
  </operation>
- <operation name="getDefaultRecommendationByUser">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="user" locked="false" />
  <parameter name="dbName" locked="false" />
  </operation>
- <operation name="getDefaultRecommendationByItem">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="item" locked="false" />
  <parameter name="dbName" locked="false" />
  </operation>
- <!-- =====>
- <!-- OAI: Métodos de importação de Metadados -->
- <!-- =====>
- <operation name="importRecordList">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="dbName" locked="false" />
  <parameter name="url" locked="false" />
  </operation>
- <operation name="importRecord">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="dbName" locked="false" />
  <parameter name="url" locked="false" />
  <parameter name="identifier" locked="false" />
  </operation>
</service>
- <service name="RecommenderWSConfig">
  <description>Configuração do Serviço Web de Recomendação Desenvolvido no Instituto de
  Computação (IC) da Universidade Estadual de Campinas (Unicamp)</description>
  <parameter name="ServiceClass"
    locked="false">br.unicamp.ic.recommender.webservice.RecommenderWSConfig</parameter>
  - <!-- =====>
  - <!-- Recursos de Instalação dos Engines no RecommenderWS -->
  - <!-- =====>
- <operation name="installEngine">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="engineName" locked="false" />
  <parameter name="xmlFile" locked="false" />
  <parameter name="jarEncodedFile" locked="false" />
  </operation>
- <operation name="getParametersList">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="engineName" locked="false" />
  </operation>
- <operation name="installedEngines">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  </operation>
- <operation name="getEngineInformation">
  <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
  <parameter name="engineName" locked="false" />
  </operation>
- <!-- =====>

```

```

- <!-- Recursos de Configuração dos Engines nas Bibliotecas Digitais -->
- <!-- =====>
=<operation name="activateEngine">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="engineName" locked="false" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="getDefaultEngine">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="getActivatedEngines">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="setDefaultEngine">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
<parameter name="engineClassName" locked="false" />
</operation>
=<operation name="getParameter">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
<parameter name="engineClassName" locked="false" />
<parameter name="parameter" locked="false" />
</operation>
=<operation name="setParameter">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
<parameter name="engineClassName" locked="false" />
<parameter name="parameter" locked="false" />
<parameter name="value" locked="false" />
</operation>
=<operation name="getParameterValues">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
<parameter name="engineClassName" locked="false" />
</operation>
- <!-- =====>
- <!-- Recursos de criação de Bibliotecas Digitais no RecommenderWS -->
- <!-- =====>
=<operation name="createDigitalLibrary">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
</operation>
=<operation name="configureDigitalLibrary">
<messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
<parameter name="dbName" locked="false" />
<parameter name="uriDB" locked="false" />
<parameter name="driver" locked="false" />
<parameter name="user" locked="false" />
<parameter name="password" locked="false" />
</operation>
</service>
</serviceGroup>

```

Anexo II – Comandos de criação do Banco de Dados

```
CREATE TABLE item (itemid serial NOT NULL, identifier varchar(250) NOT NULL,  
title text, creator text, description text, base64bincontents text, filename text, date text, subject  
text, language text, format text, type text, contributor text, coverage text, publisher text, relation  
text, rights text, source text, textcontents text, CONSTRAINT itempk PRIMARY KEY  
(identifier), CONSTRAINT UniqueItemId UNIQUE (itemid) )
```

```
CREATE TABLE rating (userid integer NOT NULL DEFAULT 0, itemid integer NOT  
NULL DEFAULT 0, rating double precision NOT NULL DEFAULT 0, ratingdate text,  
CONSTRAINT u_i_pair PRIMARY KEY (userid, itemid))
```

```
CREATE TABLE userdb (userid integer NOT NULL DEFAULT 0, username text, name  
text, mail text, age integer DEFAULT 0, profession text, education text, region text, country  
text, interests text, CONSTRAINT userpk PRIMARY KEY (userid))
```

```
CREATE TABLE engine (classname text NOT NULL, alias varchar(100) NOT NULL,  
isdefault boolean, CONSTRAINT engine_pk PRIMARY KEY (alias))
```

```
CREATE TABLE parameter (classname varchar(200) NOT NULL, paramname  
varchar(200) NOT NULL, paramtype text, value text, CONSTRAINT parameter_pk PRIMARY  
KEY (classname, paramname))
```