UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

DÂMARA DAIANE VIEGAS DA SILVA

Recomendação de Web Services para Auxiliar a Composição

Projeto de Diplomação

Prof. Dr. Leandro Krug Wives
Orientador

Porto Alegre, dezembro de 2009

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitora de Graduação: Prof[a]. Valquiria Link Bassani
Diretor do Instituto de Informática: Prof. Flávio Rech Wagner
Coordenador do CIC: Prof. João César Netto
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“Uma vida sem desafios não vale a pena ser vivida.”
— SÓCRATES

AGRADECIMENTOS

Agradeço primeiramente à minha mãe, por ter me dado toda atenção e carinho em todos os momentos da minha vida. Obrigada mãe, por ser sempre a melhor mãe desse mundo, por ser a minha inspiração e motivo de orgulho.

Agradeço ao meu pai, que mesmo distante ajudou muito na minha formação.

Ao Silvio, meu segundo pai, que esteve sempre presente me apoiando.

À minha irmã Gabrielly, por todo amor que ela sempre demonstrou.

Ao meu irmão Tiago, pelas ajudas quando foi preciso.

Agradeço também às minhas amigas Mariane e Denise pela compreensão e amizade durante meus anos de graduação.

À Jamile e o Rodrigo, meus afilhados queridos, que me apoiaram muito nessa fase da minha vida.

Às minhas amigas Gabi e Letícia, pelos ótimos momentos e pelas palavras amigas.

Ao Bridi, meu grande amigo, que me ensinou muita coisa sobre tecnologias e também sobre a vida.

Ao Caçula, meu amigo e colega de trabalho, que me ajudou muito nos momentos de desespero com o trabalho de conclusão.

À minha prima Lidiane, grande amiga e parceira para todos os momentos.

Ao meu orientador, Leandro Wives, que esteve sempre presente me ajudando.

Muito obrigada a todos que participaram desta etapa da minha vida. Eu não teria conseguido sem vocês.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

BPEL	Business Process Execution Language
WSDL	Web Services Description Language
XML	Extensible Markup Language
XLANG	XML business process language
IBM	International Business Machines
WSFL	Web Services Flow Language
JWS	Java Web Services
SOAP	Simple Object Access Protocol
CWS	Composite Web Services
W3C	World Wide Web Consortium
XLANG	XML business process language
WSFL	Web Services Flow Language

LISTA DE FIGURAS

RESUMO

Recomendação de serviços é uma técnica bastante utilizada para ajuda nas tomadas de decisões sobre produtos, informações ou até mesmo serviços que o usuário deseja utilizar. Tal técnica é muito utilizada em sites e sistemas comerciais, pois facilita o uso de serviços que tenham a finalidade procurada pelo cliente.

O objetivo deste trabalho é propor uma ferramenta para recomendar serviços Web, auxiliando a sua composição ou a gerência de composições. Por exemplo, quando um serviço falha ou quando são necessários mais serviços para formar uma composição, é preciso pesquisar algum outro serviço com a mesma finalidade para substituí-lo. Através da ferramenta proposta, a partir de um serviço informado pelo usuário / cliente, são recomendados outros serviços semelhantes, possibilitando substituições.

Palavras-chave: Recomendação, serviços web, serviços web compostos.

Web Services Recommendation to Help Composition

ABSTRACT

Service recommendation is a widely used technique to help in the decision process about products, information or even services that user wants to use. This technique is often found in commercial sites and systems, because it makes easy the use of services that has the same goal needed by the client.

The goal of this work is to propose a Web service recommendation tool to aid their composition and to easy composition management. For instance, when a service fails or when more services are needed to make a composition, one have to search for another service with the same goal to replace the failing service. With the proposed tool, it is possible to recommend services that are similar to the one that was informed by the user/customer to perform the replacement.

Palavras-chave: recomendation, web services, composite web services.

1 INTRODUÇÃO

Web Services (serviços Web) são uma tecnologia muito utilizada para a integração de sistemas que está sendo empregada com muita freqüência para a integração de softwares empresariais, já que permite a interação entre sistemas utilizados em diferentes organizações. Web Services têm se mostrado eficientes para a interconexão de sistemas através da rede. Com eles, toda comunicação passa a ser dinâmica e principalmente segura, pois não há intervenção humana e, além disso, fazem com que os recursos da aplicação do software estejam disponíveis sobre a rede.

Nos atuais mercados, altamente competitivos, as companhias são pressionadas a alcançar desempenho elevado por meio de produtividade, tempo rápido de resposta ao mercado, flexibilidade e força de trabalho eficiente. Isso deve ser realizado mesmo com vários produtos em diversas plataformas. SOA (Service Oriented Architecture) é uma abordagem baseada em atividades para projetar e desenvolver soluções flexíveis que podem ser facilmente combinadas com novas tecnologias. SOA permite que componentes de processos de negócios sejam montados e coordenados de maneira rápida e eficiente, dando aos negócios a agilidade necessária para responder às mudanças e garantindo serviços de alta qualidade. Um serviço, do ponto de vista da arquitetura SOA, é uma função de um sistema computacional que é disponibilizado para outro sistema, que deve funcionar de forma independente do estado de outros serviços, exceto nos casos de serviços compostos, e deve possuir uma interface bem definida. Normalmente, a comunicação entre o sistema cliente e aquele que disponibiliza o serviço é realizada através de Web Services (CALLOWAY, 2008). Se um único serviço Web não pode satisfazer as requisições do usuário, deve haver uma possibilidade de combinar os serviços existentes em conjunto para atender tal solicitação. Esta tendência tem provocado um número considerável de trabalhos de investigação sobre a composição de serviços Web, que serão mostrados no capítulo 3.

Os serviços compostos, que são a combinação de vários outros serviços, têm a vantagem de possuir um conjunto de serviços agrupados num só. Por exemplo, uma pessoa que quer viajar pode usufruir de um serviço combinado com outros. Pegando como exemplo um serviço que oferece o vôo, a hospedagem em um hotel, o aluguel de um carro e uma rota com passeios turísticos, tudo incluso em um pacote oferecido por uma empresa. A empresa oferece todos esses serviços ao turista por um preço, facilitando assim o processo, sem que o turista precise buscar tudo isso em separado. Dessa maneira, ele acaba optando pelo pacote oferecido pela empresa de viagens. Esses serviços são cada vez mais oferecidos aos usuários da internet, pois facilitam muito usufruir de um conjunto de opções combinados com a mesma finalidade (MAY; MARKOVSKY, 2006).

Apesar de muitos esforços, a composição de serviços Web ainda é uma tarefa muito complexa e é complicado para o usuário lidar com todo o processo manualmente. Os

motivos para tal complexidade são: o número de serviços disponíveis na web aumenta drasticamente e o repositório para pesquisa é muito grande; os serviços Web podem ser criados e atualizados em tempo real, assim o sistema de composição precisa detectar a atualização em tempo de execução e a decisão deve ser feita com base nas informações atualizadas; os serviços da Web podem ser desenvolvidos por diferentes organizações, que utilizam modelos diferentes para descrever os serviços, no entanto, não existe uma linguagem única para definir e avaliar os serviços da Web em uma forma idêntica.

Os usuários da Web possuem acesso a uma grande quantidade de informações, produtos e serviços, muitos deles irrelevantes. Para que os usuários possam executar suas tarefas e tomar decisões corretas, eles necessitam de informações, produtos e serviços corretos. Nesse contexto, os Sistemas de Recomendação surgem como um complemento às ferramentas de busca, pois costumam ser autônomos e proativos, aconselhando o usuário sobre os itens mais adequados, de acordo com o seu perfil. Sendo assim, a aplicação de técnicas de recomendação aos serviços Web é uma atividade importante, que exige muita pesquisa em relação a como desenvolver modelos específicos, que levem em conta os atributos e as características desses elementos.

Esse trabalho está relacionado ao projeto Dodona (WIVES, 2008), que tem como objetivo estudar e propor técnicas de recomendação aplicadas aos serviços web que levem em conta seus diferentes aspectos (tipos de atributos, composição, registro de uso...) e considerem a reputação dos diferentes elementos envolvidos no processo (serviços, clientes, provedores). Também tem como objetivo estudar e aplicar técnicas de análise de similaridade de serviços Web simples e compostos, visando identificar serviços complementares e alternativos; estudar e desenvolver técnicas de recomendação de serviços web simples e compostos; estudar e propor modelos de reputação que possam ser aplicados aos sistemas de recomendação, principalmente aos sistemas de recomendação de serviços web e a divulgação dos resultados obtidos.

Objetivos

O objetivo deste trabalho é a recomendação de serviços Web alternativos a um serviço dado. Para tanto, elaborou-se uma ferramenta que identifica os serviços Web utilizados em cada composição, por categorias. Com isso, podemos identificar e listar esses serviços, que podem servir como alternativas.

Estrutura do Texto

- O capítulo 1, a Introdução, visa dar uma idéia geral ao leitor às nomenclaturas e também fornecer o conhecimento básico necessário de recomendação de serviços.

- O capítulo 2, Serviços Web e Composição de Serviços Web, são descritos os serviços Web, as tecnologias e conceitos relacionados a eles. Também são descritos os serviços Web compostos, seu funcionamento, exemplos práticos e também exemplos de código.

- O capítulo 3, Sistemas de Recomendação aplicada a Serviços Web, mostra alguns trabalhos relacionados, indicando como os sistemas de recomendação já foram utilizados para resolver os problemas de Web Services.

- O capítulo 4, Recomendação aplicada à composição de Web Services, apresenta os detalhes da implementação do trabalho. Tecnologias utilizadas, banco de dados, relatórios.

- O capítulo 5, Considerações Finais, apresenta as conclusões do trabalho, e também as possibilidades de extensões em trabalhos futuros.

2 SERVIÇOS WEB E COMPOSIÇÃO DE SERVIÇOS WEB

Nesta seção são descritos os serviços Web, a linguagem utilizada para representá-lo, sua arquitetura e conceitos relacionados a eles. Também são descritos os serviços Web compostos, seu funcionamento, exemplos práticos, exemplos de código e a linguagem utilizada para descrevê-los.

2.1 Web Services

Web Service é uma solução utilizada para distribuição de serviços em que seus componentes são independentes de plataforma e permitem a interoperabilidade entre aplicações (BOOTH, 2004). Essa independência de plataforma deve-se ao formato utilizado para a construção de mensagens, o XML, mediante o qual é permitido descrever dados de uma maneira organizada, sem amarração a tipos definidos em certa linguagem ou plataforma. Logo, tem-se outra característica importante dos Web Services: a interoperabilidade entre aplicações. Dessa maneira, qualquer aplicação que possa usar dados XML e comunicar-se sobre um protocolo Web, reúne condições para ser cliente de um Web Service.

A arquitetura de um Web Service é basicamente constituída pelos seguintes elementos: provedor de serviço (service provider), cliente do serviço (service requestor) e servidor de registro (service registry), de acordo com (RECKZIEGEL, 2006).

O provedor de serviços é a entidade que cria o Web Service. Ele representa a plataforma que hospeda o Web Service permitindo que os clientes acessem o serviço. Para que isto ocorra, ele precisa descrever o Web Service em um formato padrão, indicando suas funcionalidades (e.g., entradas e saídas) de forma compreensível para qualquer um que precise usar esse serviço. Para tanto se utiliza o WSDL, descrito na próxima seção.

O cliente de serviços é qualquer um que utilize um Web Service criado por um provedor de serviços. É a aplicação que está procurando, invocando ou iniciando uma interação com o Web Service. Esse conhece a funcionalidade do Web Service, a partir da descrição disponibilizada pelo provedor de serviços, recuperando os seus detalhes através de uma pesquisa sobre o registro publicado. Através desta pesquisa, também o consumidor de serviços pode obter o mecanismo para ligação com este Web Service. Um servidor de registro é a localização central onde o provedor de serviços pode relacionar seus Web Services, e no qual um consumidor de serviços pode pesquisá-los. O registro dos serviços contém informações como detalhes de uma empresa, quais os serviços que ela fornece e a descrição técnica de cada um deles. Portanto, o provedor de serviços define a descrição do serviço para o Web Service e publica esta para o consumidor de serviços no registro de serviços. O consumidor de serviços utiliza a descrição do serviço publicada para se ligar ao provedor de serviços e invocar ou interagir com a implementação do Web Service (CUNHA, 2002).

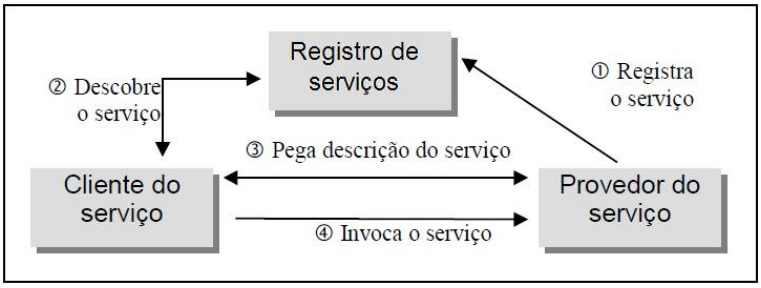


Figura 2.1: Arquitetura Web Services (SONDA; MONTEZ, 2007)

O principal problema da arquitetura mostrada na figura 2.1 é a identificação e pesquisa dos serviços. Nem sempre é fácil pesquisar um serviço, encontrá-lo e após isso utilizá-lo, pois existem diferentes serviços que podem ou não ser relevantes para determinada necessidade. Em serviços compostos acontece a mesma coisa, pois o criador do serviço composto tem que descobrir vários serviços que sirvam para montar a tal composição. E é ainda pior, pois quando a composição está executando e acontece algum problema, tem de haver uma troca dinâmica, e a descoberta (busca) ideal deveria ser automática para que tudo continuasse funcionando.

Sendo assim, a aplicação de técnicas de recomendação aos serviços Web é uma atividade importante, que exige muita pesquisa em relação a como desenvolver modelos específicos, que levem em conta os atributos e as características desses elementos, e que enfrenta novos desafios relacionados com a interoperação e com o uso desses elementos (WIVES, 2008).

2.2 WSDL

O WSDL é uma especificação desenvolvida pelo W3C que permite descrever os Web Services segundo um formato XML (CHINNICI et al., 2002). O WSDL é extensível para permitir a descrição dos serviços e suas mensagens, independentemente dos formatos de mensagem e dos protocolos de rede que sejam usados. O WSDL descreve os serviços disponibilizados à rede através de uma semântica XML, sendo que este providencia a documentação necessária para se chamar um sistema distribuído e o procedimento necessário para que esta comunicação se estabeleça.

A figura 2.2 apresenta o trecho da descrição de um Web Service que combina uma série de serviços de viagens. Nesse trecho, pode-se chamar atenção aos seguintes atributos importantes:

- import namespace é onde são chamados os wsdl relacionados (Employee.wsdl e Airline.wsdl);
- message name é um método que possui 2 parâmetros (employee e flightData);
- portType são usados para acesso a serviços do próprio processo BPEL;
- partnerLinks são partes que interagem com o processo BPEL.
- partnerLinkType caracteriza o PartnerLink que está relacionado a um partnerLinkType.

```
<?xml version="1.0" encoding="utf-8" ?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
             xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
             xmlns:xs="http://www.w3.org/2001/XMLSchema"
             xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
             xmlns:emp="http://packtpub.com/service/employee/"
             xmlns:aln="http://packtpub.com/service/airline/"
             xmlns:tns="http://packtpub.com/bpel/travel/"
             targetNamespace="http://packtpub.com/bpel/travel/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/" >
<import namespace="http://packtpub.com/service/employee/"
        location="./Employee.wsdl"/>
<import namespace="http://packtpub.com/service/airline/"
        location="./Airline.wsdl"/>
<message name="TravelRequestMessage">
  <part name="employee" type="emp:EmployeeType" />
  <part name="flightData" type="aln:FlightRequestType" />
</message>
<portType name="TravelApprovalPT">
  <operation name="TravelApproval">
    <input message="tns:TravelRequestMessage" />
  </operation>
</portType>
<portType name="ClientCallbackPT">
  <operation name="ClientCallback">
    <input message="aln:TravelResponseMessage" />
  </operation>
</portType>
<plnk:partnerLinkType name="travelLT">
  <plnk:role name="travelService">
    <plnk:portType name="tns:TravelApprovalPT" />
  </plnk:role>
  <plnk:role name="travelServiceCustomer">
    <plnk:portType name="tns:ClientCallbackPT" />
  </plnk:role>
</plnk:partnerLinkType>
</definitions>
```

Figura 2.2: Exemplo de um código WSDL (JURIC, 2003)

2.3 Web Services compostos

A composição de Web Services é uma junção de serviços da Web para os usuários que desejam usufruir de vários serviços ao mesmo tempo. Na verdade, o usuário ou cliente não se importa muito com o fato de ser uma composição de vários serviços. Os serviços compostos surgiram porque as vezes uma necessidade não é oferecida por um único serviço, mas sim vários serviços. Isso também vai ao encontro da definição de SOA, onde uma aplicação é construída pela composição de vários serviços diferentes.

Várias linguagens de especificação para compor os Web Services são relatados na literatura, por exemplo, o WS-BPEL (LEYMANN; ROLLER; THATTE, 2003), WS-CDL (BARROS; DUMAS; OAKS, 2005), e XLANG. WS-BPEL é hoje o padrão para compor Web Services, e será a linguagem mostrada neste trabalho.

A infra-estrutura da composição pretende ser um Middleware que resolve problemas, oferecendo um framework que facilite a definição, criação e execução de Web Services focando a sua preocupação na lógica de negócio, e não nos pormenores de baixo nível. A composição de Web Services foca-se na implementação interna das operações dos

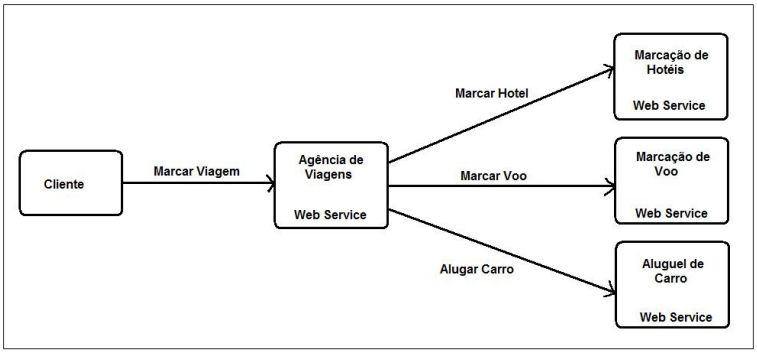


Figura 2.3: Exemplo de composição de Web Services Agência de Viagens (NOGUEIRA, 2006)

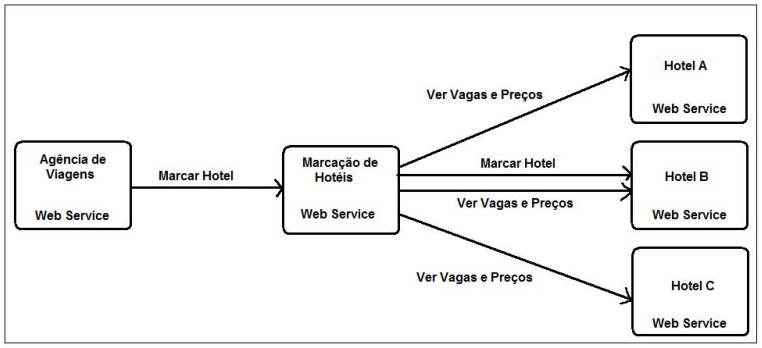


Figura 2.4: Serviço Marcação de Hotéis (NOGUEIRA, 2006)

serviços. A especificação de um serviço composto é interno a uma empresa e é mantido privado apesar de poder usar outros Web Services externos como seus componentes (NOGUEIRA, 2006), como mostra a figura 2.3.

A figura 2.3 mostra um exemplo de Web Service composto, onde um serviço inteiro, Agência de Viagens, possui outros três serviços internos, Marcação de Hotéis, Marcação de Vôos e Aluguel de Carros, cada um com uma função diferente, onde são invocados pelo serviço Agência de Viagens. O cliente se preocupa apenas em marcar sua viagem e a Agência de Viagens é quem vai resolver a escolha do hotel, passagem de avião e aluguel de carro.

A figura 2.4 mostra um exemplo de funcionamento de um dos serviços mostrados na figura 2.3, Marcação de Hotéis.

O serviço Agência de Viagens invoca o serviço Marcação de Hotéis, que por conseqüência invoca outros serviços, para verificação de vagas, preços e confirmar vaga.

Os Web Services compostos muitas vezes podem falhar (um dos serviços que faz parte da composição pode não responder, por exemplo) ou pode ser que não seja apropriado para o usuário naquele momento (serviço não está relacionado com o objetivo do usuário), recorrendo à alternativa de buscar outro serviço que seja similar ao que não pôde ser utilizado. Por isso a preocupação deste trabalho em recomendar serviços que possam ser úteis ao usuário. Caso ele não consiga utilizar o serviço que deseja, ele conseguirá usufruir de um serviço similar. Como exemplo, a figura anterior mostra o serviço de Marcação de Hotéis, e, caso algum serviço de Ver Vagas e Preços falhe (Web Service do Hotel A), ele

parte para uma segunda alternativa, onde é recomendado o serviço que possui a mesma finalidade (Web Service do Hotel B).

Uma forma de indicar como os serviços Web são compostos é através da linguagem BPEL, apresentada na seção seguinte.

2.4 BPEL

BPEL é uma linguagem baseada em XML que permite orquestrar diferentes serviços e que explicita a lógica do fluxo de processos de negócio através de um conjunto de atividades de um processo e do controle de dependências entre essas atividades, de acordo com (MATTEW, 2004). No BPEL, cada composição é um processo de negócio que interage com um conjunto de Web services (chamados de parceiros) para atingir determinado objetivo. A comunicação dos processos com seus parceiros é feita na forma P2P, onde os parceiros oferecem funções a serem usadas pelos processos e os processos somente utilizam estas funções, sem disponibilizar outras (STAL, 2002).

A linguagem BPEL foi desenvolvida através da colaboração entre a Microsoft, IBM e BEA Systems, e combina a XLANG e WSFL, as gerações anteriores de linguagens de processos (MATTEW, 2004). O BPEL representa um passo importante no desenvolvimento de aplicações numa área com processos de negócio flexíveis, com unidades distribuídas, independentes de plataforma. Permite coordenar a comunicação assíncrona entre serviços, relacionar trocas de mensagens entre as partes, implementar o processamento paralelo de atividades, manipular/transformar dados trocados pelos parceiros, suportar atividades e transações de negócios demoradas e também prover tratamento de exceções consistente.

A orquestração permite combinar diversos serviços, onde um processo central controla a execução dos processos envolvidos. Esses processos não sabem quem estão envolvidos em um processo composto. Somente o coordenador, mostrado na figura 2.5, conhece a ordem de invocação dos serviços. Exemplo de atividades de um processo BPEL: uma mensagem é enviada através de uma mensagem invoke; o serviço pode esperar que uma de suas funções seja invocada usando o receive; e o reply envia o resultado da composição ao cliente.

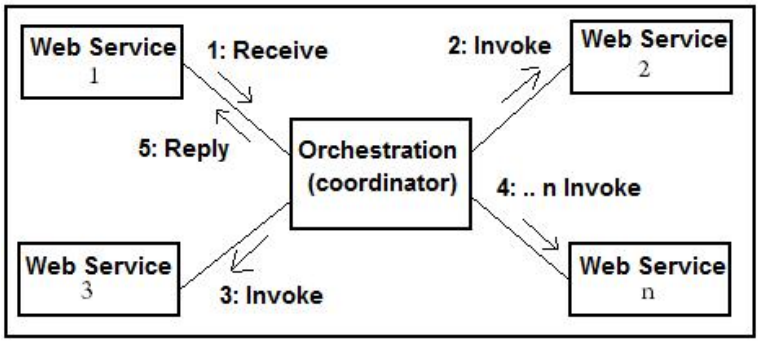


Figura 2.5: Orquestração de Serviços (JURIC, 2003)

A figura 2.6 apresenta o trecho da descrição de BPEL que faz o planejamento de uma viagem. Pode-se verificar o nome do serviço (servicename = BusinessTravelProcess), que está disponibilizado em xmlns:soap = http://schemas.xmlsoap.org/ws/2003/03/business-process/ e as operações por ele disponibilizadas são employeeTravelStatus, AmericanAir-

lines, DeltaAirlines, dentro da tag partnerLink. Dentro da tag sequence são feitas as requisições de serviços pelos clientes, verificações de status dos serviços e são dadas as respostas das requisições aos clientes.

Quando é utilizado BPEL para definir um processo de negócio, é definido um novo serviço Web que é a composição de serviços existentes. A interface do novo serviço Web composto utiliza um conjunto de tipos de porta que fornece operações como qualquer outro serviço da Web. Para invocar esse processo de negócios, é preciso chamar o resultado do serviço Web composto. A figura 2.7 mostra o projeto de negócio de viagens utilizando BPEL.

Como se pode ver na figura 2.7, o serviço possui vários outros serviços incluídos. No início do fluxograma é feita a verificação do status do serviço Employee Travel. Se o serviço está disponível, passa para o próximo nível do fluxograma e faz a verificação dos valores das passagens em duas companhias de viagens, American Airlines e Delta Airlines. Depois da verificação dos valores, é escolhida a companhia que ofereceu menor valor de passagem. Tudo isso é feito com serviços trabalhando junto a outros, como se pode verificar nos módulos ao lado direito do fluxograma.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Asynchrnous BPEL process -->
<process name="BusinessTravelProcess"
         targetNamespace="http://packtpub.com/bpel/travel/"
         xmlns:soap="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
         xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
         xmlns:trv="http://packtpub.com/bpel/travel/"
         xmlns:emp="http://packtpub.com/service/employee/"
         xmlns:aln="http://packtpub.com/service/airline/" >
   <partnerLinks>
      <partnerLink name="client"
                   partnerLinkType="trv:travelLT"
                   myRole="travelService"
                   partnerRole="travelServiceCustomer"/>
      <partnerLink name="employeeTravelStatus"
                   partnerLinkType="emp:employeeLT"
                   partnerRole="employeeTravelStatusService"/>
      <partnerLink name="AmericanAirlines"
                   partnerLinkType="aln:flightLT"
                   myRole="airlineCustomer"
                   partnerRole="airlineService"/>
      <partnerLink name="DeltaAirlines"
                   partnerLinkType="aln:flightLT"
                   myRole="airlineCustomer"
                   partnerRole="airlineService"/>
   </partnerLinks>
   <sequence>
      <!-- Receive the initial request for business travel from client -->
      <receive partnerLink="client"
               portType="trv:TravelApprovalPT"
               operation="TravelApproval"
               variable="TravelRequest"
               createInstance="yes" />
      <!-- Prepare the input for the Employee Travel Status Web Service -->
      <assign>
        <copy>
          <from variable="TravelRequest" part="employee"/>
          <to variable="EmployeeTravelStatusRequest" part="employee"/>
        </copy>
      </assign>
      <!-- Synchronously invoke the Employee Travel Status Web Service -->
      <invoke partnerLink="employeeTravelStatus"
              portType="emp:EmployeeTravelStatusPT"
              operation="EmployeeTravelStatus"
              inputVariable="EmployeeTravelStatusRequest"
              outputVariable="EmployeeTravelStatusResponse" />
<!-- Prepare the input -->
      <assign>
        <copy>
          <from variable="TravelRequest" part="flightData"/>
          <to variable="FlightDetails" part="flightData"/>
        </copy>
        <copy>
          <from variable="EmployeeTravelStatusResponse" part="travelClass"/>
          <to variable="FlightDetails" part="travelClass"/>
        </copy>
      </assign>
      <!-- Make a callback to the client -->
      <invoke partnerLink="client"
              portType="trv:ClientCallbackPT"
              operation="ClientCallback"
              inputVariable="TravelResponse" />
   </sequence>
</process>
```

Figura 2.6: Exemplo de Código da Linguagem de Processos (JURIC, 2003)

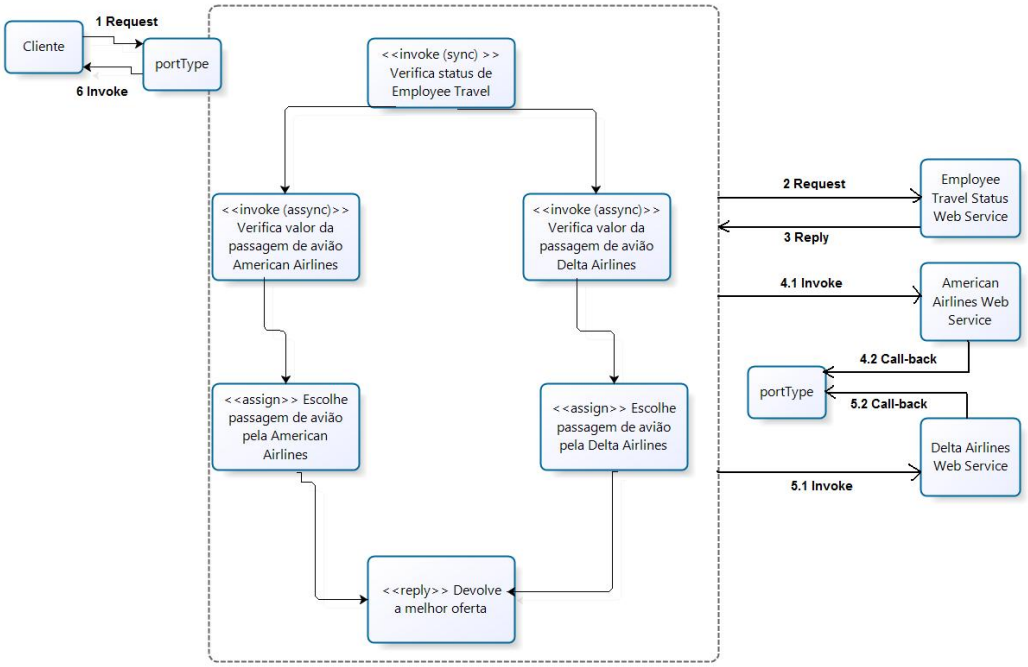


Figura 2.7: Exemplo de Processo de Negócio para Viagens (JURIC, 2003)

3 SISTEMAS DE RECOMENDAÇÃO APLICADOS A SERVIÇOS WEB

Nesta seção são descritos alguns trabalhos que, de alguma forma, mostram estudos para a recomendação de serviços.

3.1 Dodona - Recomendação de Serviços Web

Com a quantidade de informações que se pode encontrar através da Internet, as pessoas, muitas vezes, deparam-se com uma diversidade enorme de opções. Costumeiramente, um indivíduo possui pouca ou praticamente nenhuma experiência para realizar escolhas dentre as várias alternativas que lhe são apresentadas ou fica em dúvida relativamente qual seria o produto ideal. Com o objetivo de equacionar tais questionamentos e necessidades frente à escolha entre alternativas, geralmente os indivíduos confiam nas recomendações que são passadas por outras pessoas, propagandas, notas de jornais, entre outros. Os sistemas de recomendação são utilizados para minimizar esses problemas e ajudam na otimização e aumento da capacidade deste processo de indicação. Um sistema de recomendação é aquele que tenta "adivinhar"o que um usuário está procurando para poder lhe oferecer mais. Um dos grandes desafios deste tipo de sistema é realizar o casamento correto entre os que estão recomendando e aqueles que estão recebendo a recomendação, ou seja, definir e descobrir este relacionamento de interesses.

Muitos sites de venda na internet utilizam recomendação para conquistar usuários, oferecendo produtos da sua preferência, nos quais o sistema mostra para aquele indivíduo itens que possam ser do seu interesse, a partir daquelas informações que já foram passadas anteriormente (páginas consultadas, produtos visualizados, etc.). Alguns exemplos são: Amazon , Submarino , Americanas , entre outros. Os sistemas de recomendação podem ser classificados como: de filtragem colaborativa, baseados em conteúdo e híbridos. Os de filtragem colaborativa são aqueles em que o usuário avalia itens que, posteriormente, são recomendados para usuários de mesmo perfil. Os baseados em conteúdo são aqueles onde a recomendação é feita a partir da similaridade entre objetos e o perfil do usuário. Os híbridos combinam as duas técnicas anteriores (WIVES, 2008).

3.2 Casamento entre Web Services e Sistemas de Recomendação

Em (MAAMAR et al., 2008) é mostrado que Web Services visam desenvolver serviços em larga escala, enquanto sistemas de recomendação indicam qual o conjunto de decisões que devem ser tomadas. O casamento dessas duas tecnologias mostra que juntos podem ser muito mais úteis, pois um oferece vantagens ao outro. A recomendação, por ex-

emplo, é essencial para evitar o uso de serviços ruins, baseando-se em serviços usados anteriormente e ajuda a descobrir Web Services com menos esforço. Os Web Services possuem características que tem impacto sobre os sistemas de recomendação que ajudam a recomendar outros serviços através de técnicas feito por avaliação de itens pelos usuários, similaridade entre objetos e as duas combinadas. O objetivo do artigo é descobrir os serviços Web através de recomendação. Um dos fatores críticos para o sucesso dos serviços Web é o de simplificar a sua descoberta de acordo com requisitos funcionais e não funcionais. Um sistema de recomendação tem a base sobre a qual os serviços Web poderiam, por exemplo, analisar a composição de cenários anteriores em que participaram, para estabelecer relações e classificar os pares que compartilharam com eles estes cenários de composição.

3.3 Mercado Competitivo de Web Services

Em (CHENG et al., 2007) os autores retomam a idéia de tornar os serviços Web compostos diretamente acessíveis aos usuários. Na verdade eles observam que as pesquisas atuais sobre os serviços da Web tendem a olhar para o problema de composição de um único fornecedor. Como resultado, as várias soluções que são apresentadas para resolver este problema não promovem a competitividade, um elemento-chave no mundo de hoje. Vários provedores de serviços da Web em concorrência podem satisfazer o pedido do mesmo serviço com diferentes opções. Soluções consistem em um mercado de serviços, os CWSMarts, e um modelo de três camadas. Essas camadas são processos de negócios, instanciação de serviço e de rede.

3.4 Composição de Serviços Web em uma Arquitetura Orientada a Serviços

Em (JURETA et al., 2007) destaca-se que seu objetivo consiste em reunir os serviços da Web em grupos chamados centros de serviços. Cada centro de serviço é dedicado a um determinado tipo de funcionalidade através de uma variedade de serviços Web compostos que podem ser criadas e utilizar os serviços da Web que residem neste centro de serviço. A identificação desses serviços Web é atribuída a um serviço Web dedicado, conhecido como mediador. Essa identificação é geralmente dependente em alguns requisitos não-funcionais (por exemplo, tempo de resposta, custo de desempenho) que os usuários definiram. Apesar da natureza que caracteriza os centros de serviços e CWSMarts em termos de serviços de hospedagem de sites e serviços Web compostos, respectivamente, o conceito de centros de serviços não difere muito das práticas atuais de compor Web services. Uma dessas práticas é a tela de registros (por exemplo, UDDI) para identificar os Web services necessários. Em (JURETA et al., 2007) a seleção é restrita a centros de serviço e assim, mais focalizado.

3.5 Uma Abordagem para Monitorar o Uso de Web Services e Avaliação da Qualidade de Serviço

Em (CRUZ et al., 2003) desenvolveu-se um data mart para monitorar e avaliar o uso e a qualidade dos serviços Web. Aqui, o data mart é essencialmente voltado para as necessidades do componente de serviços Web e não de serviços Web compostos. Um exemplo da

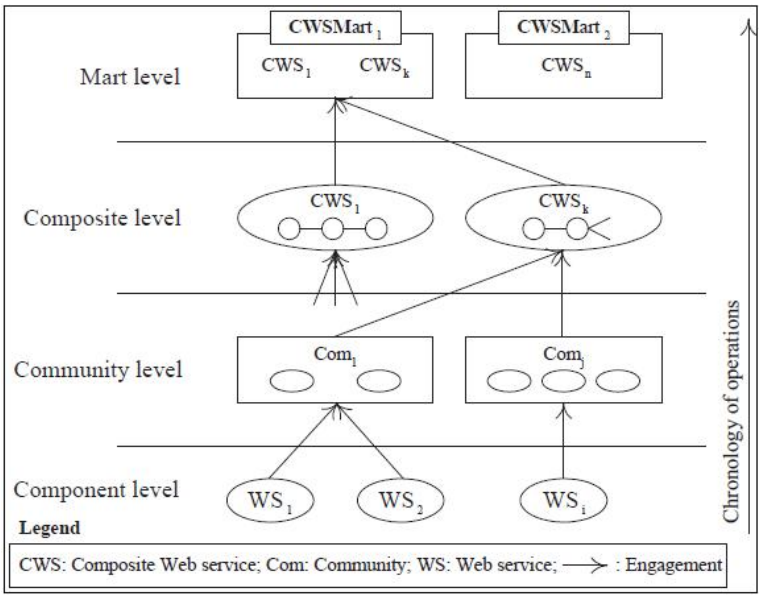


Figura 3.1: Arquitetura que Suporta CWSMarts (MAAMAR et al., 2008)

necessidade de identificar padrões de interação entre usuários e serviços da Web significa uma melhor interação / solução em comparação com os existentes que são oferecidos aos usuários futuramente. O autor sugeriru um serviço, nomeado WSLogA, para capturar os dados necessários para a tomada de uma decisão. Dois esquemas profissionais sobre os dados WSLogA são propostos. Isso permite que os administradores da Web e os comerciantes tirem proveito das informações que os logs de serviços da Web fornecem.

A maioria dos trabalhos citados não aborda a recomendação para a composição de serviços Web. Isso foi proposto no projeto Dodona (MAAMAR e WIVES 2009), no módulo Recommendation, objetivo principal desse trabalho. O módulo de recomendação será futuramente integrado ao projeto, juntamente aos outros módulos da figura 3.2. Em (MAAMAR et al., 2008) os serviços da Web devem ser identificados e, em seguida, ordenados com base em alguns requisitos não-funcionais. Logo em seguida, as questões semânticas entre os serviços identificados devem ser abordadas. Por último, a confiabilidade dos serviços executados na Web deve ser garantida. Os serviços compostos podem ser chamados a qualquer momento sem passar pelas etapas regulares (descoberta, seleção, composição, invocação, e monitoramento), o que poderia reduzir o impacto dos desafios sobre o andamento dessas etapas. A idéia é separar os serviços compostos de CWSMart em um subconjunto dos recursos de dados, geralmente orientados para uma finalidade específica, que pode ser distribuído para apoiar as necessidades de negócios. Os CWSMarts contém dados que ajudam a preparar estratégias de desenvolvimento, analisando as tendências e situações anteriores. A figura 3.1 mostra a arquitetura que suporta CWSMarts.

Na figura 3.1, conforme (MAAMAR et al., 2008), o nível componente é onde ficam os Web Services, e cada um deles oferece um serviço que satisfaz determinadas necessidades. A descrição de um serviço pode ser enriquecida de acordo com alguns requisitos não-funcionais. O cadastro de serviços em comunidades ocorre de acordo com algumas políticas, como a recompensa para esse serviço inscrever-se na comunidade. Outro exemplo pode ser penalizar o serviço que apresenta baixo desempenho para respostas a usuários. O nível comunidade hospeda os Web Services com a mesma funcionalidade.

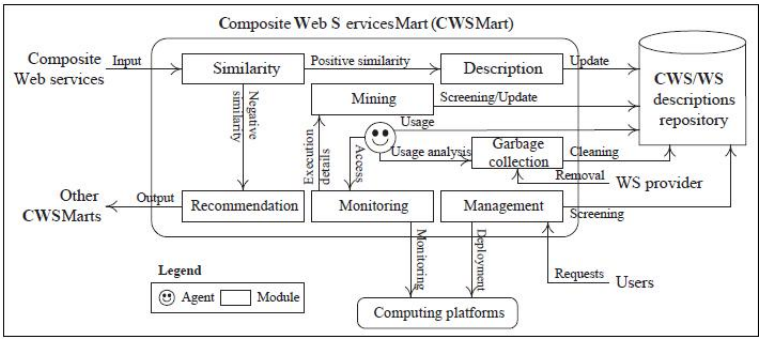


Figura 3.2: Arquitetura de um CWSMart (MAAMAR et al., 2008)

O nível de composição é sustentado pela crescente complexidade das necessidades dos usuários de hoje. Essa complexidade é abordada através da integração de vários serviços Web a partir de diferentes comunidades em serviços Web compostos. O nível de conteúdo da comunidade alimenta o nível da composição com os componentes necessários para que os serviços web compostos sejam formados. O primeiro passo para identificar as comunidades é combinando suas funcionalidades às necessidades de um determinado usuário. A segunda etapa seleciona a partir de comunidades que são identificados na primeira etapa, os serviços da Web que irão implementar essas funcionalidades. Cada comunidade contribui com um serviço da Web para uma composição de Web Service. O nível Mart é a contrapartida do nível da comunidade, com ênfase neste momento na composição. O conteúdo do nível composto alimenta o nível Mart com o serviços compostos da Web necessários que são desenvolvidas quer proactivamente (os serviços Web compostos são colocados juntos, em seguida, armazenados em CWSMarts e, finalmente, executado em caso de seleção de vários serviços compostos Web similares) ou reativamente (os serviços compostos da Web estão juntos, então executado, e finalmente armazenado em CWSMarts.

A figura 3.2 mostra em detalhes a arquitetura de um CWSMart.

O módulo Similarity compara a funcionalidade de um CWS com a funcionalidade de outro CWSMart (que é esperado para participar deste módulo). Em caso de comparação positiva, o módulo aceita o CWS no CWSMart, onde pode ser usado futuramente. O módulo Description consiste em especificar o CWS de acordo com um conjunto de argumentos antes de ter obtido a especificação armazenada no repositório de descrições de CWS. O módulo Management consulta o repositório de descrições CWS e olha para o CWSs adequado. Uma vez que m CWS apropriado é identificado, o módulo de Management implanta este CWS sobre algumas plataformas de computação. O módulo Mining procura alguns padrões recorrentes em comum entre as diferentes especificações de CWSs. O módulo Monitoring supervisiona serviços compostos da Web em tempo de execução após a sua implantação pelo módulo de gestão sobre plataformas de computação e armazena informações necessárias para uso posterior. O objetivo da vigilância é coletar alguns parâmetros de execução em CWSs como tempos de resposta e as exceções levantadas e refletir estes parâmetros no repositório de descrições de CWS. O módulo Agent coleta dados sobre o uso CWSs e WSS. De tempos em tempos, uma análise de uso é realizada para recomendar quais CWS e WSS devem ser removidas do repositório e passar essa informação para o módulo Garbage Collection. O módulo Garbage Collection controla o processo de remoção de CWSs não utilizados que é importante para manter

o desempenho do repositório CWS e WS e qualidade de serviço dos Marts. O módulo Recommendation é o principal objetivo desse trabalho. É nele onde são feitas as recomendações de serviços para uma composição. Os serviços são recomendados pela sua freqüência em BPEL da mesma categoria, onde se conclui que serviços mais utilizados são os melhores para recomendação.

3.6 Seleção Dinâmica de Web Services com Sistema de Recomendação

Em (MANIKRAO; PRABHAKAR., 2005) sugere-se recomendar serviços a partir da avaliação dos usuários. Eles avaliam cada serviço através de tempo de execução, tempo de resposta, etc. O sistema de recomendação usa a abordagem de filtragem colaborativa. Esta abordagem analisa o conjunto de serviços que foi avaliado pelo usuário e computa como eles são semelhantes ao serviço que deve ser previsto. Uma vez que os serviços semelhantes são encontrados, a previsão é calculada tomando uma média ponderada das avaliações do usuário destes serviços semelhantes. Uma vez que se sabe a previsão de votos para cada serviço que satisfaça as necessidades dos usuários, pode-se recomendar serviços em ordem de classificação.

4 RECOMENDAÇÃO APLICADA À COMPOSIÇÃO DE WEB SERVICES

O objetivo deste trabalho é propor e implementar uma técnica de recomendação que possa ser futuramente utilizada dentro do contexto do projeto Dodona, mais especificamente no contexto de Data Marts de Web services compostos (MAAMAR e WIVES 2009), no componente de recomendação descrito na figura 3.2.

O Data Mart de Web services recebe BPELs descrevendo o serviço Web composto que se deseja armazenar nele. Na arquitetura original, descrita por Maamar e Wives (2009), há um componente de análise de similaridades que verifica se este serviço tem alguma similaridade com o datamart em questão, ou seja, se ele se enquadra no contexto de atuação do datamart. Caso se encaixe o BPEL é inserido no catálogo do datamart. Caso contrário, é encaminhado a um componente de recomendação para indicar um datamart alternativo, mais adequado ao contexto da composição em questão.

O trabalho aqui proposto, no entanto, diferencia-se nessa questão, pois a proposta consiste em analisar um BPEL e armazená-lo, independente do seu contexto (a questão de datamarts com diferentes focos não é abordada neste trabalho). Com isso, tem-se a informação de quais são os Web services quem atuam em um BPEL específico e quais deles eventualmente atuam em mais de um BPEL. Aqueles serviços que são utilizados em mais de um BPEL são bons candidatos a recomendação, partindo da hipótese de que os bons serviços (com melhor reputação) são mais utilizados do que serviços ruins (pois estes últimos seriam normalmente substituídos por outros melhores no caso de apresentarem problemas ou falhas).

Deste modo, o módulo de recomendação foi implementado de maneira a indicar um serviço mais adequado (mais utilizado, na verdade) para uma determinada composição. A recomendação é feita a partir de um serviço escolhido pelo usuário. Todos os BPEL dessa mesma categoria do serviço passado são buscados e são listados os serviços encontrados neles. Os serviços são organizados por ordem de ocorrência nos BPEL.

Um ponto central na recomendação é a análise de similaridade entre os diferentes serviços utilizados nos BPEL. Este problema está sendo analisado por outros pesquisadores do projeto Dodona, e não será foco deste trabalho estudar os diferentes mecanismos de análise de similaridade de Web services. No entanto, algum mecanismo de similaridade deve ser utilizado. Para o trabalho atual, considerar-se-á que os Web services utilizados nos diferentes BPEL foram previamente analisados e classificados em categorias semânticas. Logo, o componente de análise e recomendação tem a sua disposição um catálogo de serviços Web e suas respectivas categorias. Considerar-se-á, portanto, um Web service similar a outro se ele fizer parte da mesma categoria semântica.

A idéia central do projeto consiste, portanto, em analisar arquivos BPEL com diversos

serviços, identificar suas categorias semânticas, e verificar quais desses serviços são os mais utilizados para assim poder recomendá-los.

Os processos de análise de arquivos BPEL e recomendação de Web services relacionados são descritos nas seções seguintes.

4.1 Implementação da Solução Proposta

A solução proposta realizou-se através de um programa Java versão 6, com o uso do Firebird, um banco de dados relacional que oferece recursos multiplataforma. Para criação dos relatórios foi usada a ferramenta Ireport versão 3.0.0. Os diagramas de sequência e de classe foram gerados com a ferramenta JUDE.

4.1.1 Banco de Dados

Foi utilizada uma ferramenta de código aberto para criar e administrar banco de dados usando o servidor Firebird, o FlameRobin.

O banco de dados possui seis tabelas e o relacionamento, como podemos observar na figura 4.1 é n para n entre todas as tabelas:

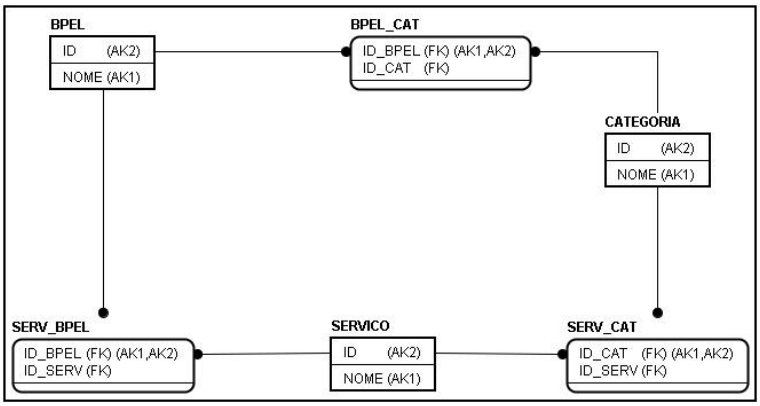


Figura 4.1: Modelagem do Banco de Dados

Como se pode ver na figura 4.2, foram criados três Generators e três Triggers. Eles existem porque é a forma com a qual se consegue auto-incrementar valores no Firebird, já que até a versão 2.0 não existem campos com auto-incremento.

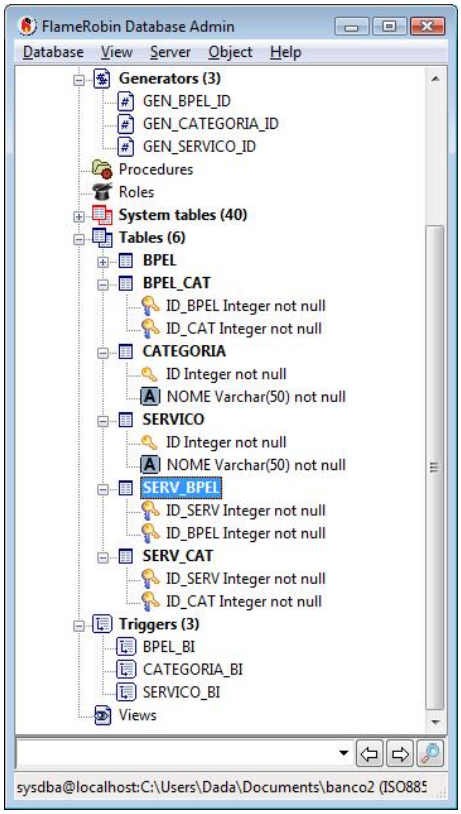


Figura 4.2: Estrutura do Banco de Dados

Tabela Serviço:

Field	Type	NULL	Default	Description
ID	Integer	**not null**		Identificador do serviço. [edit]
NOME	Varchar(50)	**not null**		Nome do serviço. [edit]

Figura 4.3: Dicionário de Dados Tabela Serviço

Tabela Categoria:

Field	Type	NULL	Default	Description
ID	Integer	**not null**		Identificador da categoria. [edit]
NOME	Varchar(50)	**not null**		Nome da categoria. [edit]

Figura 4.4: Dicionário de Dados Tabela Categoria

Tabela Bpel:

Field	Type	NULL	Default	Description
ID	Integer	**not null**		Identificador do bpel. [edit]
NOME	Varchar(50)	**not null**		Nome do bpel. [edit]

Figura 4.5: Dicionário de Dados Tabela Bpel

Tabela ServBpel:

Field	Type	NULL	Default	Description
ID_SERV	Integer	**not null**		Identificador do serviço. [edit]
ID_BPEL	Integer	**not null**		Identificador do bpel. [edit]

Figura 4.6: Dicionário de Dados Tabela ServBpel

Tabela ServCat:

Field	Type	NULL	Default	Description
ID_SERV	Integer	**not null**		Identificador do serviço. [edit]
ID_CAT	Integer	**not null**		Identificador da categoria. [edit]

Figura 4.7: Dicionário de Dados Tabela ServCat

Tabela BpelCat:

Field	Type	NULL	Default	Description
ID_BPEL	Integer	**not null**		Identificador do bpel. [edit]
ID_CAT	Integer	**not null**		Identificador da categoria. [edit]

Figura 4.8: Dicionário de Dados Tabela BpelCat

4.1.2 A Ferramenta

O diagrama da figura 4.9 mostra as ações que podem ser feitas pelo usuário (que atualmente é um cliente e no futuro será um Web Service) na ferramenta Recomendação de Serviços, que serão detalhadas a seguir:

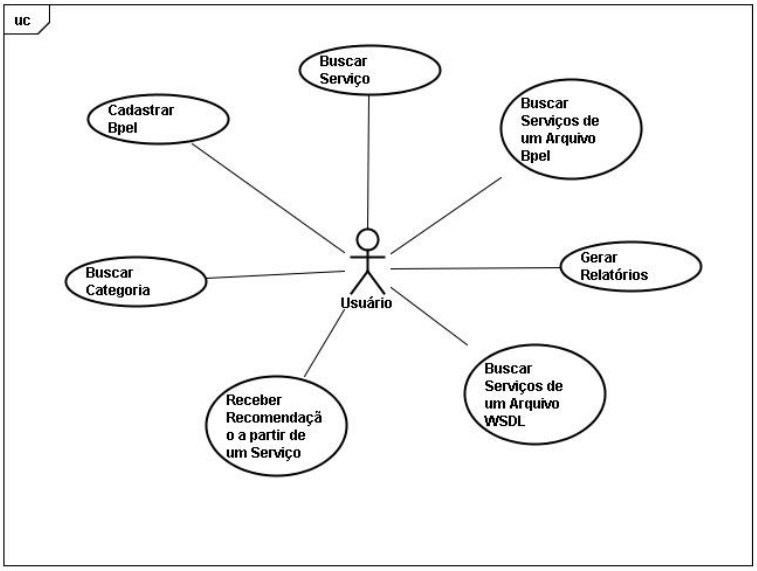


Figura 4.9: Caso de Uso

4.1.2.1 Cadastro de BPEL

Aqui é feito o cadastro de BPEL associado a uma determinada categoria. Todo BPEL, ao ser cadastrado pelo usuário, deverá conter uma categoria. Não é possível o cadastro de BPEL sem associá-lo a uma determinada categoria.

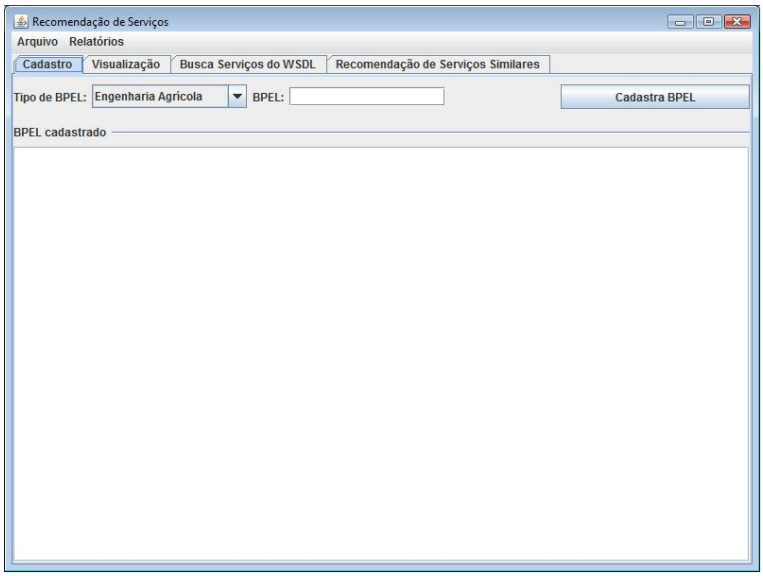


Figura 4.10: Cadastro de Bpel

- Combo **Tipo de BPEL**: onde são mostradas todas as categorias ao usuário. É feito um SELECT no banco de todas as categorias cadastradas na tabela CATEGORIA.
- Campo **BPEL**: campo para o usuário escrever o nome do serviço a ser cadastrado.
- Botão Cadastra BPEL: O BPEL é salvo na tabela BPEL, onde o id é gerado automaticamente e o nome é aquele que foi escolhido pelo usuário no campo BPEL. Depois de salvo na tabela BPEL, as informações são salvas na tabela BPELCAT, id do BPEL e o id da categoria que foi escolhida no combo.
- Área **BPEL cadastrado**: mostra ao usuário o serviço que foi cadastrado.

No diagrama de seqüência da figura 4.11 são mostradas as ações do usuário para o cadastro do BPEL. A categoria deve ser selecionada na tela e é chamado o método da classe DBUtils para a busca do id no banco (que será utilizado para o cadastro na tabela BPELCAT). Em seguida o usuário escolhe um nome para o BPEL e o INSERT é feito no banco na tabela BPEL. Após o BPEL ser cadastrado é feito o INSERT na tabela BPELCAT, onde são inseridos o id da categoria que foi buscada anteriormente e o id do BPEL que é buscado logo após a inserção no banco.

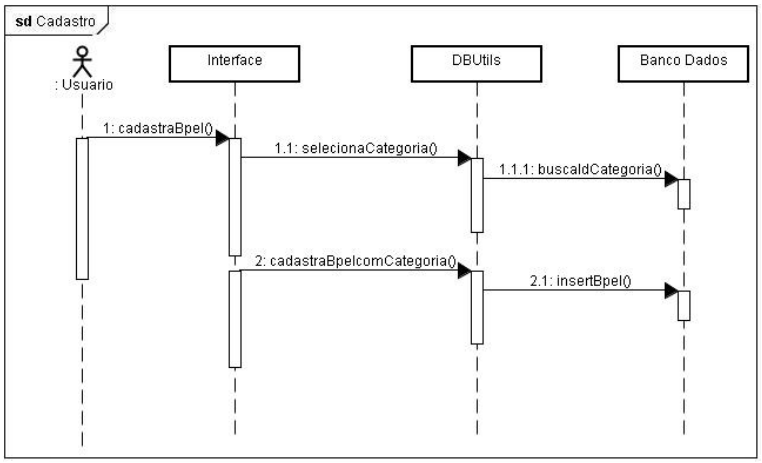


Figura 4.11: Diagrama de Sequência de Cadastro de Bpel

4.1.2.2 Visualização de Serviços e Categorias

Onde são feitas as buscas e visualizações de serviços e categorias. O usuário faz a busca de serviços ou categorias para serem visualizados na tela.

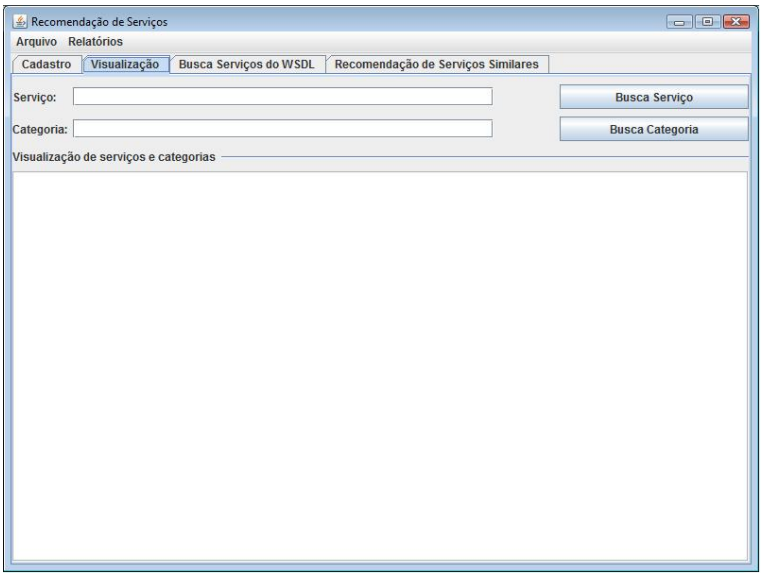


Figura 4.12: Visualização de Serviços e Categorias

- Campo **Serviço**: usuário digita o nome para busca.
- Campo **Categoria**: usuário digita o nome para busca.
- Botão **Busca Serviço**: busca serviço pelo nome digitado no campo Serviço. É feito um SELECT na tabela SERVICO com a palavra digitada pelo usuário e são retornados todos os serviços que possuem aquela String como parte do nome.
- Botão **Busca Categoria**: busca categoria pelo nome digitado no campo Categoria. É feito um SELECT na tabela CATEGORIA com a palavra digitada pelo usuário e são retornadas todas as categorias que possuem aquela String como parte do nome.

O diagrama da figura 4.13 mostra como são feitas as buscas de categoria e serviços. Ambos possuem um método na classe DBUtils que acessa diretamente o banco para serem feitos os SELECT.

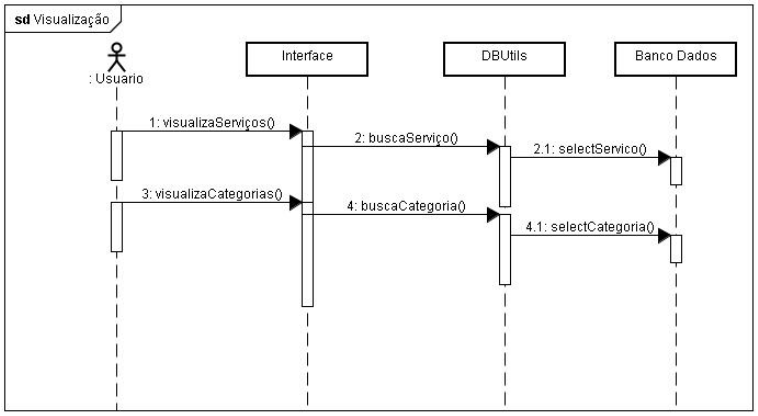


Figura 4.13: Diagrama de Sequência da Visualização de Serviços e Categorias

4.1.2.3 Busca Serviços do WSDL

Busca funções/métodos encontrados num arquivo wsdl. O usuário informa o arquivo (.wsdl) e logo após são mostrados todos os métodos/funções encontrados no arquivo.

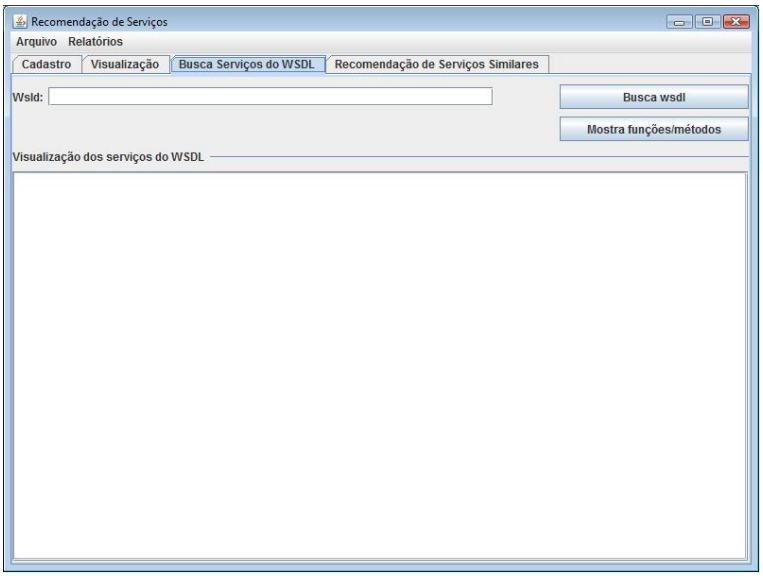


Figura 4.14: Busca Serviços do WSDL

- Campo **WSDL**: caminho do arquivo wsdl que foi selecionado no botão Busca wsdl.
- Botão **Busca wsdl**: seleciona arquivo wsdl em algum diretório do computador.
- Botão **Mostra funções/métodos**: lista todos os métodos/funções ou funcionalidades encontrados no arquivo selecionado. Realiza-se uma busca por um arquivo wsdl, que depois é lido e são extraídas todas as operações contidas nesse arquivo. São utilizados métodos do pacote javax.wsdl do Java, que possibilita descobrir operações de um wsdl, dentre outras coisas.

O diagrama da figura 4.15 mostra a busca dos serviços do arquivo wsdl escolhido pelo usuário. É feita a busca pelo arquivo, logo em seguida é chamado um método dentro da classe PercorreWsdl onde é feito a análise do arquivo (como foi explicado acima) mostrando todos os serviços daquele wsdl.

O diagrama da figura 4.16 mostra a classe que faz a análise no arquivo wsdl escolhido pelo usuário. O método recebe o caminho do arquivo no computador como uma String e retorna uma String, que é a lista dos serviços do arquivo wsdl.

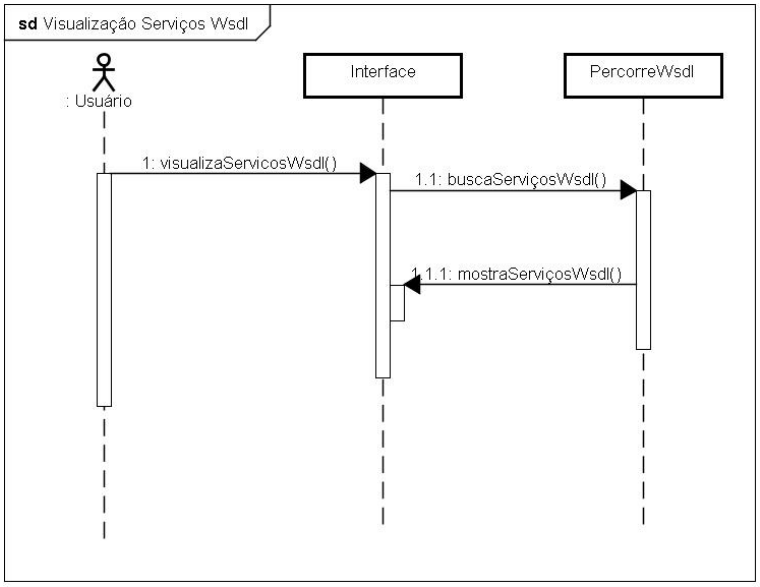


Figura 4.15: Diagrama de Sequência Visualização de Serviços do WSDL

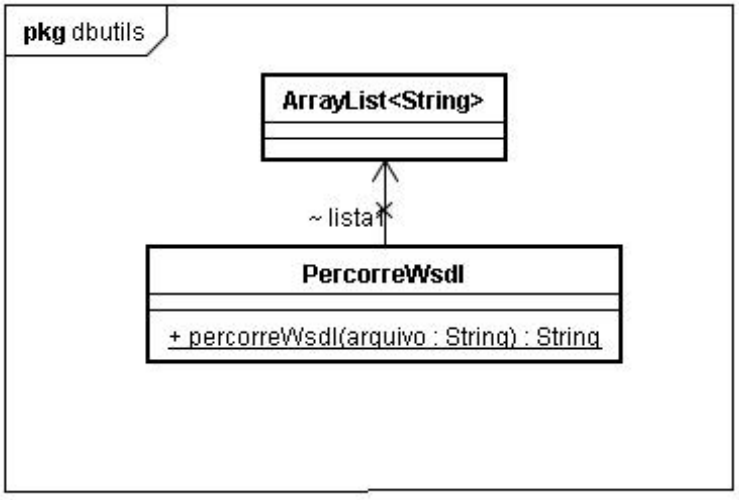


Figura 4.16: Classe PercorreWsdl

4.1.2.4 Recomendação de Serviços Similares para fins de Substituição

Uma vez que diferentes BPEL tenham sido cadastrados, é possível realizar a recomendação dos Web Services mais adequados para fins de substituição. Para tanto, dado um BPEL, os seus Web Services são identificados, assim como a categoria individual de cada um deles. Com base nisso, obtém-se a lista de serviços alternativos para cada categoria e seleciona-se, para cada serviço, aquele que possui maior freqüência de uso. A lista dos serviços que poderiam substituir os serviços atuais do BPEL é retornada.

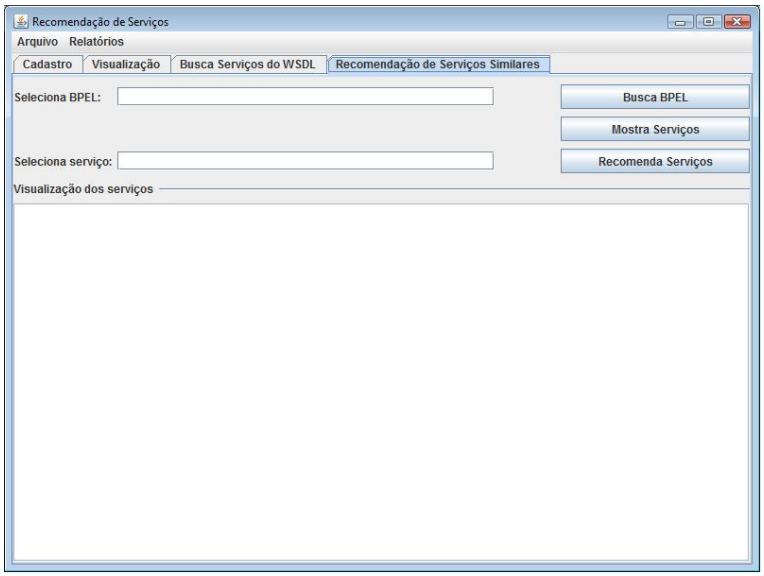


Figura 4.17: Recomendação de Serviços

- Campo **Selecionar BPEL**: nome do arquivo que o usuário busca.
- Botão **Buscar BPEL**: usuário indica arquivo ou nome do BPEL que indica quais são os serviços que fazem parte da composição.
- Botão **Mostra Serviços**: lista todos os serviços encontrados no arquivo selecionado. Aqui utiliza-se um método que percorre o arquivo BPEL e guarda o nome dos processos envolvidos e todos os serviços desse processo. Esse programa faz um parser no arquivo BPEL, que é em XML, devolvendo o nome do processo (nome usado para definir o BPEL) e também os serviços Web que fazem parte deste BPEL. Depois de ter a lista dos serviços disponíveis em um BPEL, eles podem ser cadastrados e ser separados em categorias, que é feito via JUnit. A idéia é identificar os Web Services utilizados em cada BPEL, e cada Web Service terá sua categoria. Sabendo a categoria, podemos identificar e listar esses serviços, que podem servir como alternativas, caso algum serviço falhe.
- Campo **Selecionar serviço**: com base no BPEL selecionado, usuário escolhe o serviço exemplo para o qual gostaria de receber uma recomendação.
- Botão **Recomenda serviços**: faz a recomendação a partir do serviço escolhido pelo usuário.

A recomendação funciona da seguinte forma:

- Usuário ou Cliente indica um serviço. Ele faz isso porque gostaria de receber recomendações de serviços alternativos.
- A ferramenta identifica a categoria do serviço (C).
- A ferramenta identifica todos os serviços que fazem parte da mesma categoria (WS(C))
- Para cada serviço, conta-se quantos BPEL ele participa, chamado de Frequencia do Serviço na Categoria (Fs).

  Fs é calculada pela seguinte fórmula:

  Fs = NB(s, C)

  onde NB(s) é a quantidade de BPEL em que o serviço s aparece na categoria C.
- Cria-se um ranking dos serviços, com base em Fs (ou seja, serviços ordenados pela frequencia mais alta).

A figura 4.18 mostra a sequência dos eventos para a recomendação.

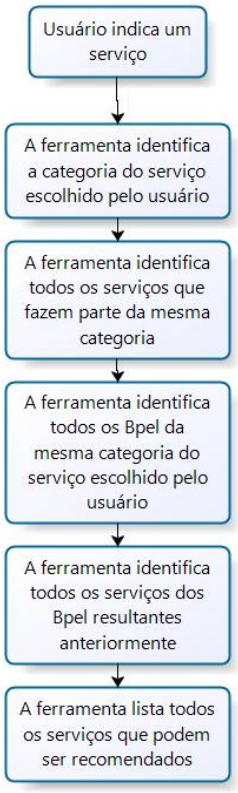


Figura 4.18: Fluxograma da Recomendação de Serviços

O diagrama da figura 4.19 mostra as operações realizadas quando o usuário busca os serviços para recomendação. Primeiramente o usuário deseja visualizar os serviços do BPEL. Então é feita a busca pelo arquivo, logo em seguida é chamado um método dentro da classe AnaliseXml onde é feito a análise do arquivo (como foi explicado acima) mostrando todos os serviços daquele BPEL. Depois da visualização dos serviços, o usuário

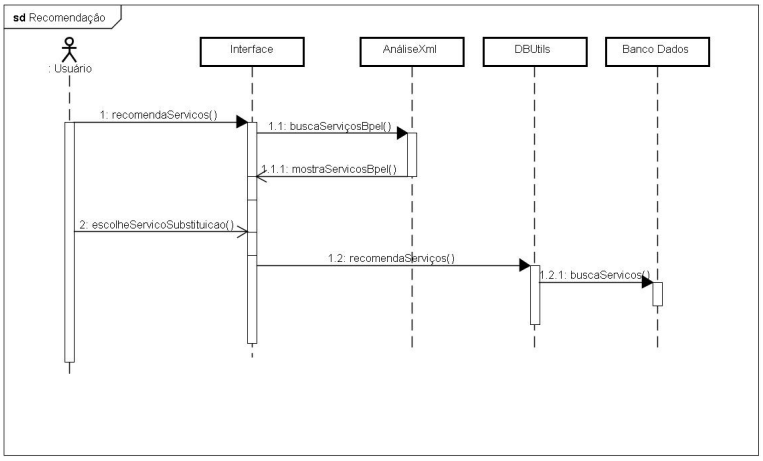


Figura 4.19: Diagrama de Sequência Recomendação de Serviços

pode escolher o serviço desejado e visualizar o relatório com os detalhes desse serviço (explicado na seção Relatórios).

O diagrama da figura 4.20 mostra a classe que faz o parser no arquivo BPEL escolhido pelo usuário. O método recebe o caminho do arquivo no computador como uma String e retorna uma String, que são os serviços mais o nome do processo BPEL.

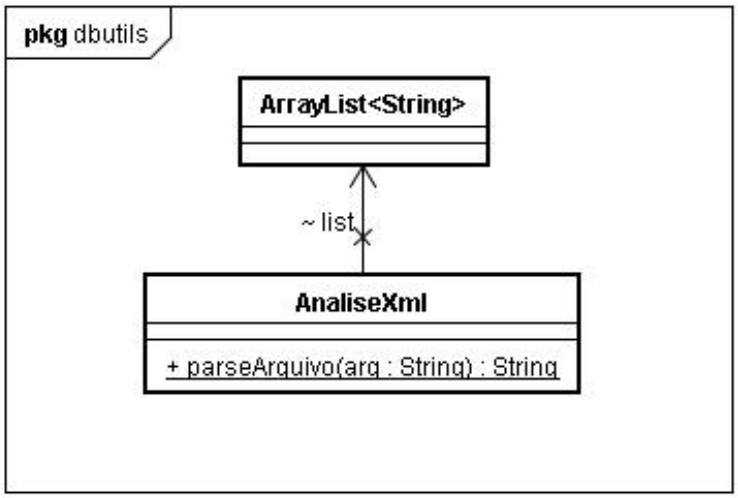


Figura 4.20: Classe AnaliseXml

4.1.3 Relatórios

Os relatórios foram elaborados com a ferramenta iRport 3.0.0. Como mostra a figura 4.21 existem seis tipos de relatórios diferentes. Tambem existe o relatório da recomendação de serviços que é gerado a partir da tela Recomendação de Serviços Similares, que será mostrado na figura 4.42.

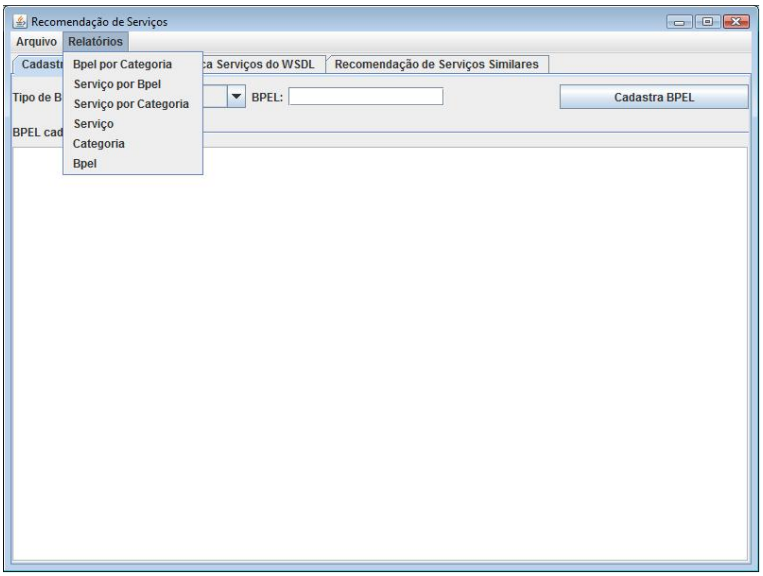


Figura 4.21: Relatórios

Para cada relatório foi criada uma classe que implementa a interface JRDataSource, que é a representação abstrata de uma fonte de dados JasperReports. Todos os tipos de fonte de dados devem implementar essa interface. Os métodos mostrados nos diagramas de classes (getFieldValue(JRField field) e next()) são da interface JRDataSource. Os construtores das classes de cada relatório recebem como parâmetro uma lista, que são os resultados das buscas feitas no banco de dados. É a partir dessa lista que são montados os relatórios.

O relatório BPEL por Categoria, figura 4.24, mostra todos os BPEL e suas respectivas categorias. É feito um SELECT no banco nas tabelas CATEGORIA e BPEL. A consulta retorna uma lista do tipo BpelCategoria, como mostra o diagrama de classes na figura 4.22.

O diagrama da figura 4.23 mostra detalhes da classe BpelCategoria. A classe possui dois atributos que são os nomes do BPEL e da categoria.

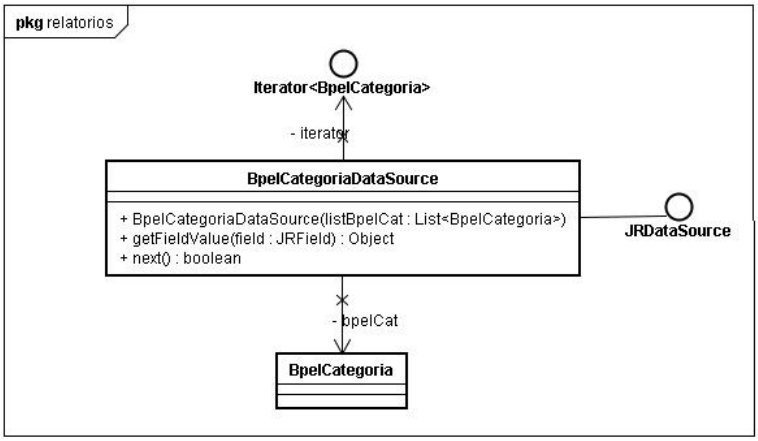


Figura 4.22: Diagrama BpelCategoriaDataSource

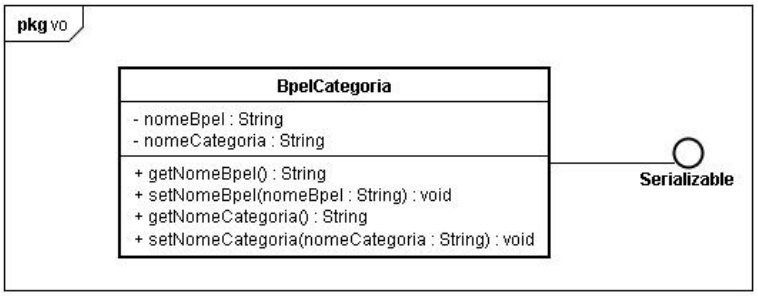
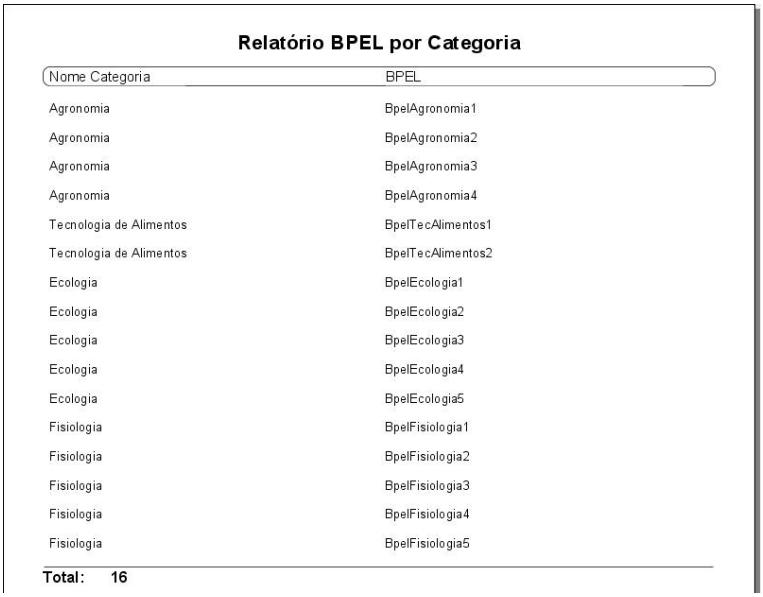


Figura 4.23: Diagrama BpelCategoriaVO



Relatório BPEL por Categoria

Nome Categoria	BPEL
Agronomia	BpelAgronomia1
Agronomia	BpelAgronomia2
Agronomia	BpelAgronomia3
Agronomia	BpelAgronomia4
Tecnologia de Alimentos	BpelTecAlimentos1
Tecnologia de Alimentos	BpelTecAlimentos2
Ecologia	BpelEcologia1
Ecologia	BpelEcologia2
Ecologia	BpelEcologia3
Ecologia	BpelEcologia4
Ecologia	BpelEcologia5
Fisiologia	BpelFisiologia1
Fisiologia	BpelFisiologia2
Fisiologia	BpelFisiologia3
Fisiologia	BpelFisiologia4
Fisiologia	BpelFisiologia5

Total: 16

Figura 4.24: Relatório BPEL por Categoria

O relatório Serviço por BPEL, figura 4.27, mostra todos os serviços e seus respectivos BPEL. É feito um SELECT no banco nas tabelas SERVIÇO e BPEL. A consulta retorna uma lista do tipo BpelServico, como mostra o diagrama de classes da figura O relatório Serviço por BPEL mostra todos os serviços e seus respectivos BPEL. É feito um SELECT no banco nas tabelas SERVIÇO e BPEL. A consulta retorna uma lista do tipo BpelServico, como mostra o diagrama de classes da figura 4.25.

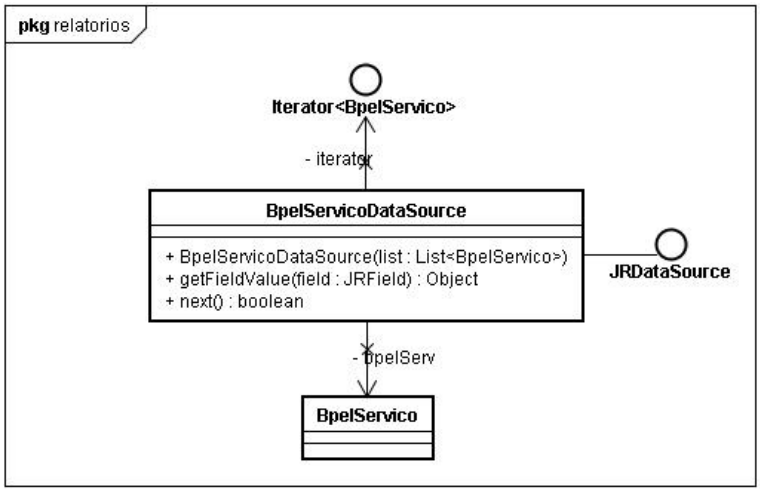


Figura 4.25: BpelServicoDataSource

O diagrama da figura 4.26 mostra detalhes da classe BpelServico. A classe possui dois atributos que são os nomes do BPEL e do serviço.

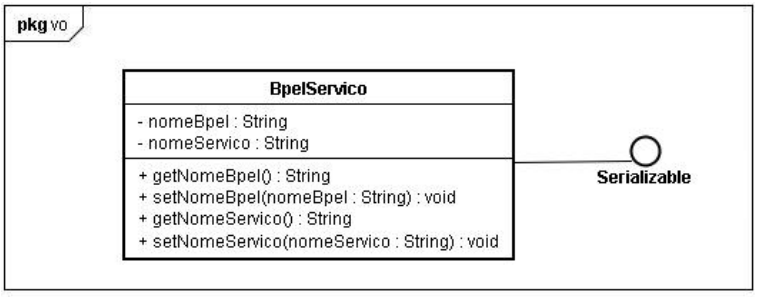


Figura 4.26: Diagrama BpelServVO

Relatório Serviços por BPEL

BPEL	Nome Serviço
BpelAgronomia1	servicoAgronomia1
BpelAgronomia2	servicoAgronomia1
BpelAgronomia3	servicoAgronomia1
BpelEcologia1	servicoEcologia1
BpelEcologia2	servicoEcologia2
BpelFisiologia1	servicoFisiologia1
BpelFisiologia2	servicoFisiologia2
BpelFisiologia3	servicoFisiologia3

Total: 8

Figura 4.27: Relatório Serviço por BPEL

O relatório Serviço por Categoria, figura 4.30, mostra todos os serviços e suas respectivas categorias. É feito um SELECT no banco nas tabelas SERVIÇO e CATEGORIA. A consulta retorna uma lista do tipo ServicoCategoria, como mostra o diagrama de classes da figura 4.28.

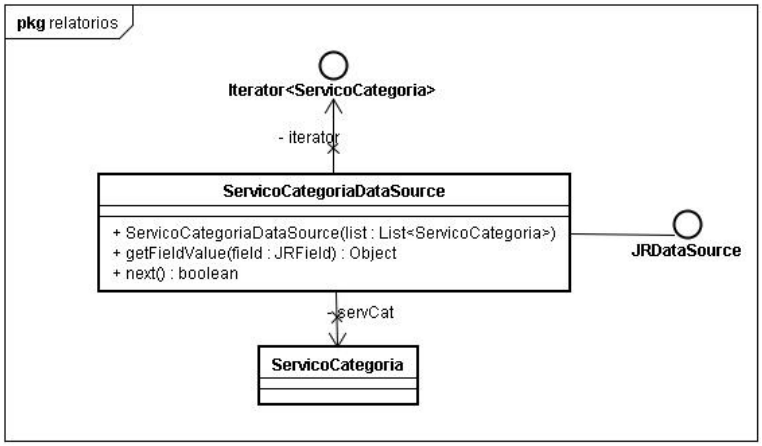


Figura 4.28: Diagrama ServicoCategoriaDataSource

O diagrama da figura 4.29 mostra detalhes da classe ServicoCategoria. A classe possui dois atributos que são os nomes da categoria e do serviço.

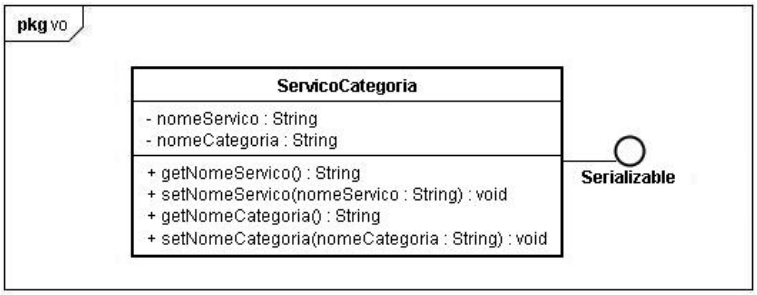
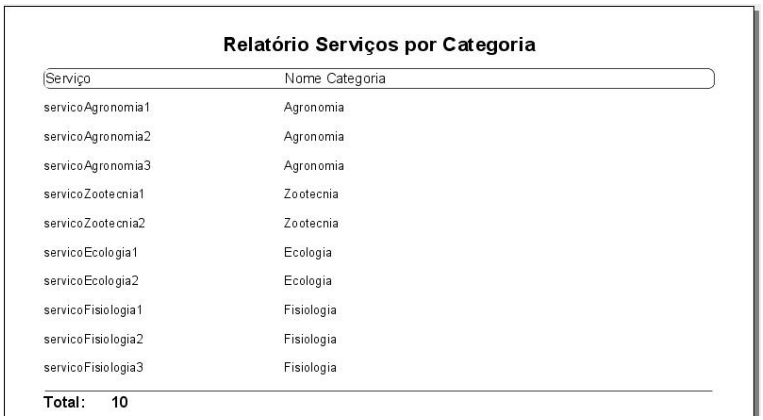


Figura 4.29: Diagrama servCatVO



Relatório Serviços por Categoria

Serviço	Nome Categoria
servicoAgronomia1	Agronomia
servicoAgronomia2	Agronomia
servicoAgronomia3	Agronomia
servicoZootecnia1	Zootecnia
servicoZootecnia2	Zootecnia
servicoEcologia1	Ecologia
servicoEcologia2	Ecologia
servicoFisiologia1	Fisiologia
servicoFisiologia2	Fisiologia
servicoFisiologia3	Fisiologia

Total: 10

Figura 4.30: Relatório Serviços por Categoria

O relatório Serviço, figura 4.33, mostra todos os serviços cadastrados. É feito um SELECT no banco na tabela SERVICO. A consulta retorna uma lista do tipo Servico, como mostra o diagrama de classes da figura 4.31.

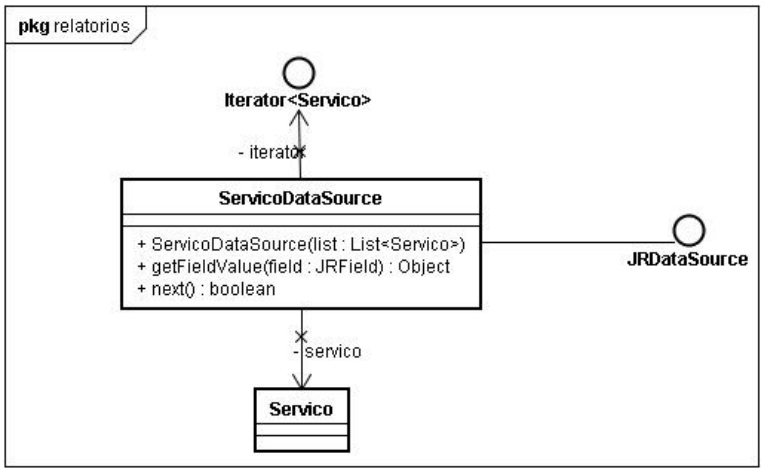


Figura 4.31: Diagrama ServicoDataSource

O diagrama da figura 4.32 mostra detalhes da classe Servico. A classe possui dois atributos, nome do serviço e o id (identificador único).

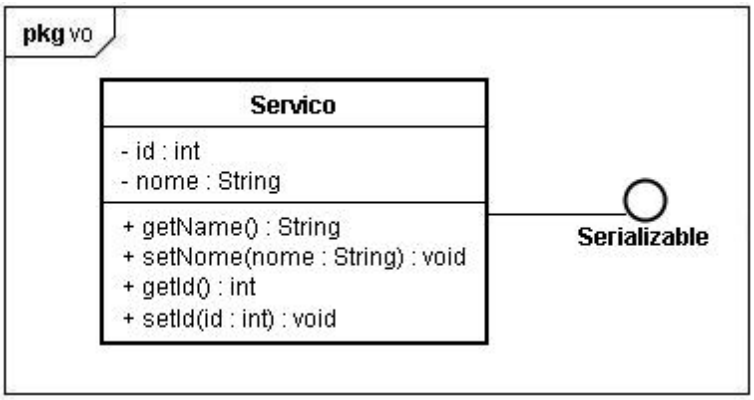


Figura 4.32: Diagrama ServicoVO

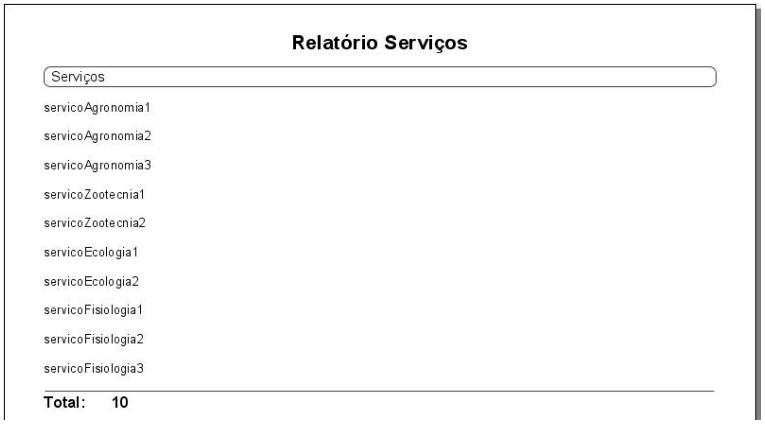


Figura 4.33: Relatório Serviços

O relatório Categoria, figura 4.36, mostra todas as categorias cadastradas. É feito um SELECT no banco na tabela CATEGORIA. A consulta retorna uma lista do tipo Categoria, como mostra o diagrama de classes da figura 4.34.

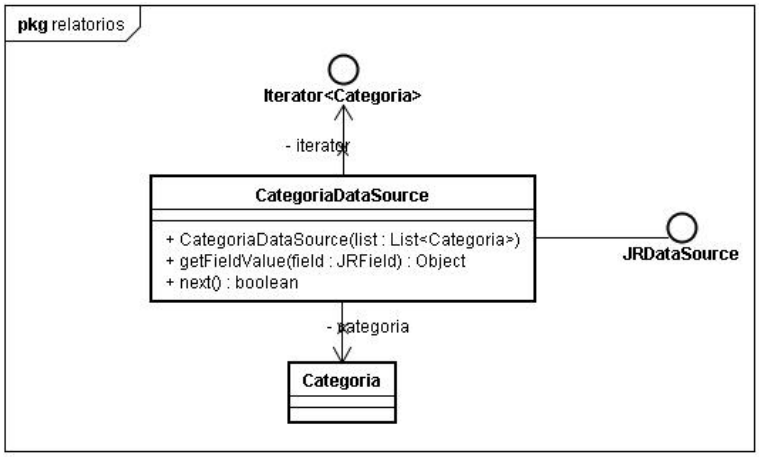


Figura 4.34: Diagrama CategoriaDataSource

O diagrama da figura 4.35 mostra detalhes da classe Categoria. A classe possui dois atributos, nome da categoria e o id (identificador único).

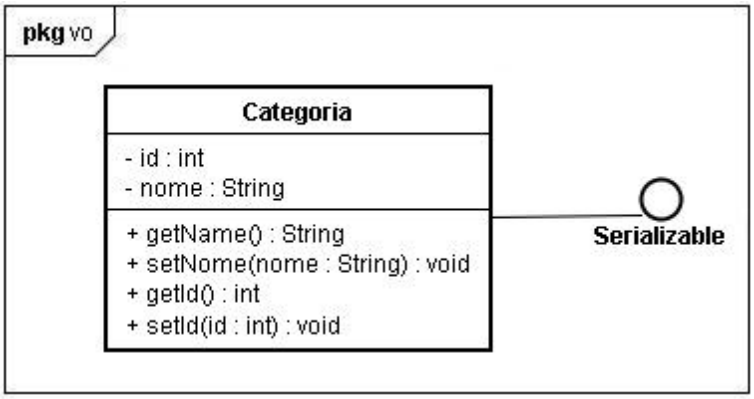


Figura 4.35: Diagrama CategoriaVO

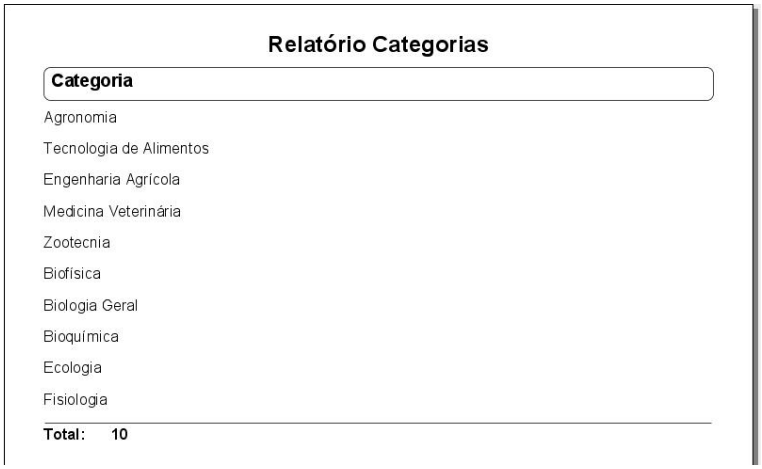


Figura 4.36: Relatório Categoria

O relatório BPEL, figura 4.39, mostra todos os BPEL cadastrados. É feito um SELECT no banco na tabela BPEL. A consulta retorna uma lista do tipo BPEL, como mostra o diagrama de classes da figura 4.37.

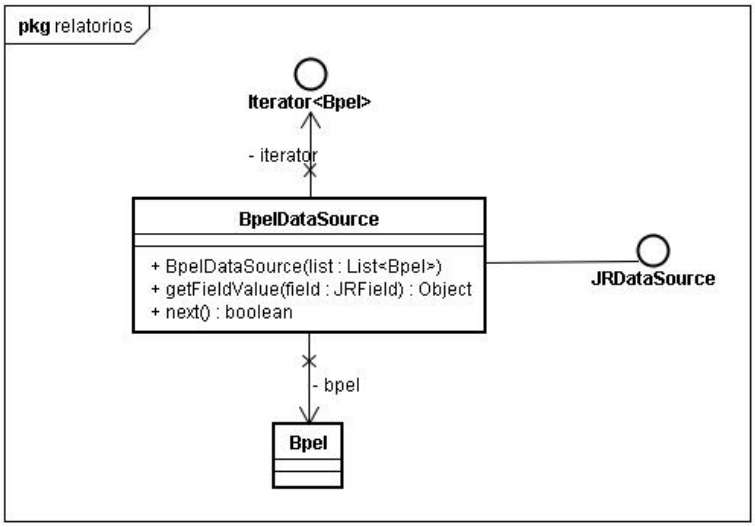


Figura 4.37: Diagrama BpelDataSource

O diagrama da figura 4.38 mostra detalhes da classe BPEL. A classe possui dois atributos, nome do BPEL e o id (identificador único).

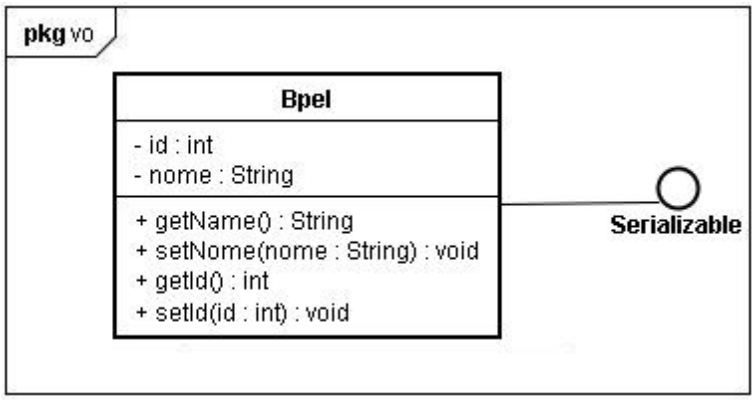


Figura 4.38: Diagrama BpelVO

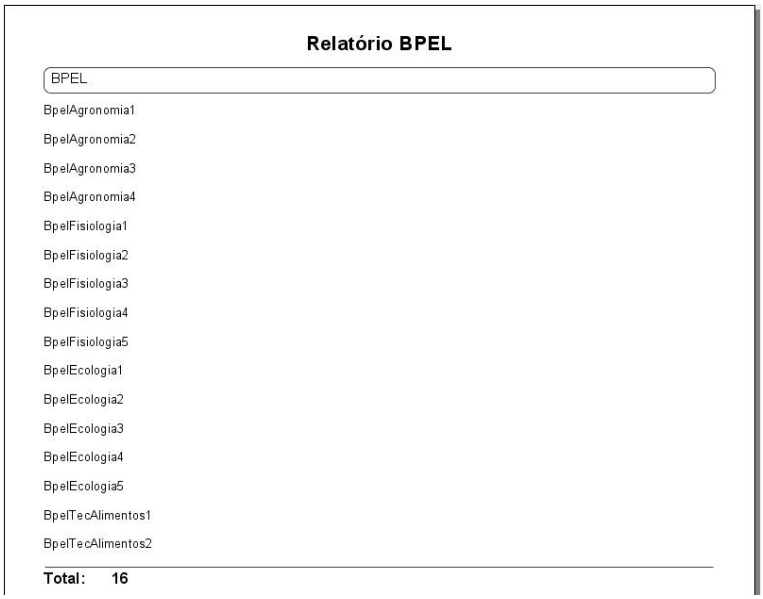


Figura 4.39: Relatório BPEL

O relatório de Recomendação de Serviços, figura 4.42, mostra a partir de um serviço escolhido pelo usuário, a categoria e a quantidade que o serviço aparece em cada BPEL. No relatório da figura 4.42 o usuário digitou o nome do serviço servicoFisiologia7 e foram buscadas todas as informações citadas acima no banco de dados. A consulta retorna uma lista do tipo Recomendacao, como mostra o diagrama de classes da figura 4.40.

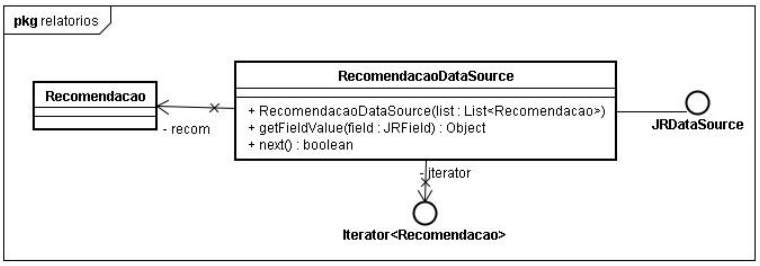


Figura 4.40: Diagrama RecomendacaoDataSource

O diagrama da figura 4.41 mostra detalhes da classe Recomendacao. A classe possui três atributos, nome do serviço, da categoria e qtde, quantidade de serviços por BPEL da mesma categoria.

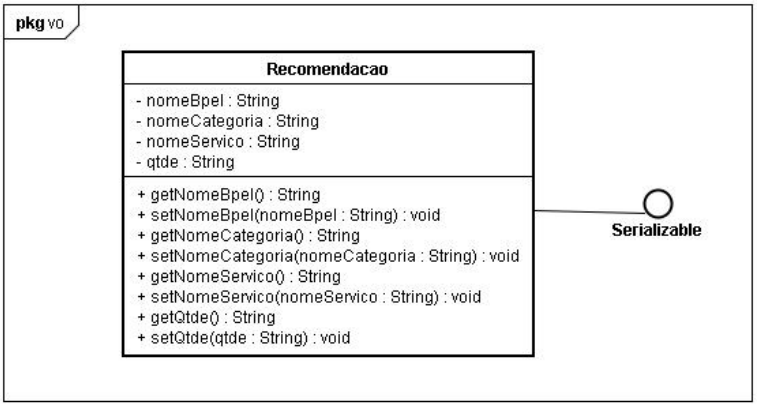
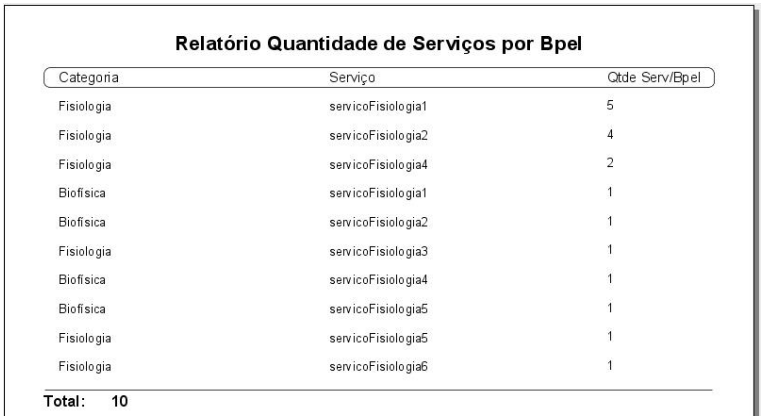


Figura 4.41: Diagrama RecomendacaoVO



Relatório Quantidade de Serviços por Bpel

Categoria	Serviço	Qtde Serv/Bpel
Fisiologia	servicoFisiologia1	5
Fisiologia	servicoFisiologia2	4
Fisiologia	servicoFisiologia4	2
Biofísica	servicoFisiologia1	1
Biofísica	servicoFisiologia2	1
Fisiologia	servicoFisiologia3	1
Biofísica	servicoFisiologia4	1
Biofísica	servicoFisiologia5	1
Fisiologia	servicoFisiologia5	1
Fisiologia	servicoFisiologia6	1

Total: 10

Figura 4.42: Relatório Recomendação de Serviços Similares

4.1.4 JUnit

Foi utilizado JUnit, framawork em Java para criação de testes unitários. Para cada método que acessa o banco na classe DBUtils, foi feito um método de teste. O diagrama da figura 4.43 mostra os métodos utilizados para os testes. Cada método possui acesso ao banco, onde as inserções e consultas podem ser realizadas através dessa classe, DBUtilsTeste.

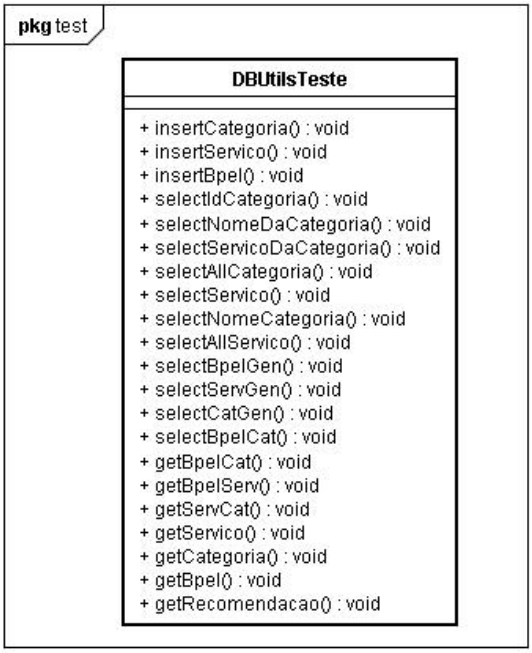


Figura 4.43: Diagrama DBUtilsTest

O diagrama de classes da figura 4.44 mostra a classe e todos seus métodos de acesso ao banco de dados, DBUtils. Na parte superior estão declaradas as constantes utilizadas nos métodos na parte de baixo do diagrama.

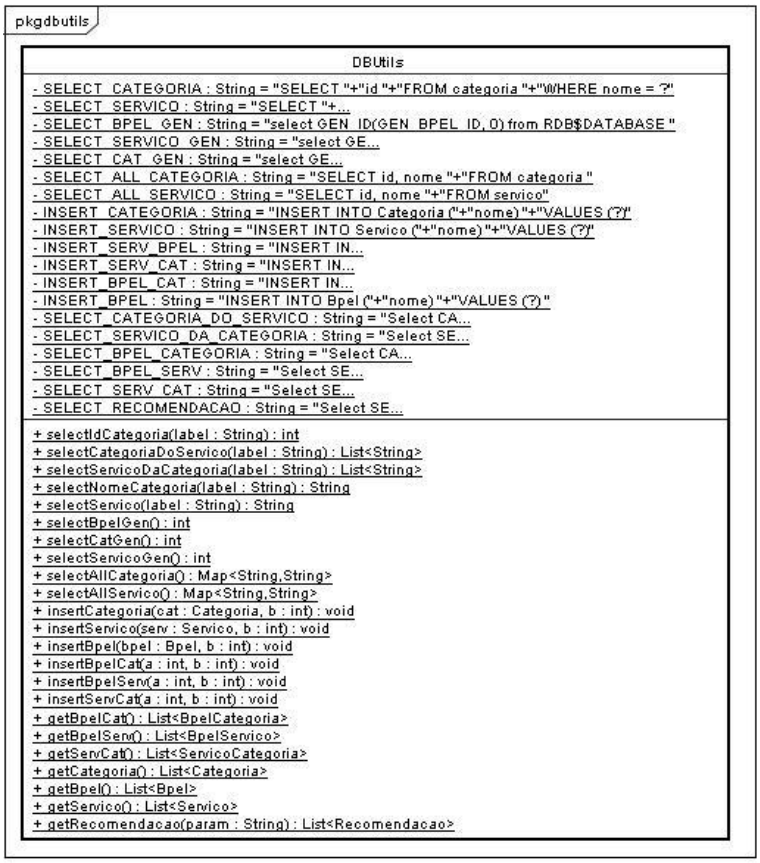


Figura 4.44: Diagrama DBUtils

5 CONCLUSÕES E TRABALHOS FUTUROS

A recomendação de serviços, como visto nos capítulos anteriores, é uma maneira de ajudar usuários/clientes a encontrar um serviço adequado para sua aplicação. Quando um cliente deseja agrupar serviços e formar uma composição ou quando algum serviço de uma composição falha há diversas dificuldades para a escolha do serviço ideal e com a recomendação isso pode ser simplificado.

A ferramenta Recomendação de Serviços possui quatro módulos, onde o de Recomendação de Serviços Similares faz todo o processo de recomendação de serviços para auxiliar a composição. O usuário informa um serviço que pode ter falhado ou não ter atendido às expectativas desejadas e a ferramenta retorna os serviços que podem substituí-lo. A recomendação é feita sugerindo que os serviços mais utilizados em Bpel com a mesma categoria sejam ideais para serem recomendados.

O módulo de Recomendação de Serviços Similares será futuramente introduzido ao CWSMart, mostrado na figura 3.2, que funcionará como o módulo Recommendation.

Seria interessante adicionar à ferramenta algumas técnicas para automatizar o cadastro ao banco de dados, já que a inclusão de serviços e categorias são feitas pelos testes unitários. A busca dos serviços de arquivos bpel e wsdl são mostradas ao usuário, mas o cadastro ainda não é automatizado. Também adicionar análise do Bpel de acordo com entradas e saídas dos métodos contidos nele, para assim dizer quão semelhantes são os serviços e então recomendá-los. A fórmula de geração ranking poderá levar em conta as também categorias para ordenar os serviços, onde aqueles que aparecem em muitas categorias seriam penalizados. O ideal seria ter um serviço e não uma aplicação desktop, que retorne o resultado da recomendação em XML.

APÊNDICE A ANÁLISE DE SIMILARIDADE ENTRE WSDL

A idéia inicial para análise de similaridade entre WSDL consistiu em fazer uma comparação entre arquivos WSDL. Para isso, foi necessário no mínimo dois desses arquivos e também que os dois tivessem o mesmo propósito (os dois arquivos necessariamente seriam previsão do tempo, por exemplo). Essas comparações são feitas por comparação de Strings, sendo mostrados resultados não muito eficientes. Strings com o mesmo propósito, muitas vezes não possuem similaridade, pois as pessoas pensam diferente na hora de nomear operações. Isso contribuiu para a busca de outra forma e procurar outra maneira de recomendar serviços. A análise de similaridade está sendo realizada por alunos do professor Antônio Rodrigo de Vit, trabalho complementar ao de recomendação de serviços.

A comparação foi feita utilizando uma biblioteca do Java, wsdl4j-1.4 (TURRELL, 2008) e o Simmetrics (CHAPMAN, 2006) (biblioteca de funções que visa identificar similaridades entre strings).

O algoritmo utilizado para a análise de similaridade funciona da seguinte maneira: primeiramente é necessário informar os endereços dos arquivos WSDL, depois é feita uma busca por operation (onde são definidas as operações no WSDL) e, logo em seguida, essas operações são listadas. Essas listas de operações são percorridas e é feita a comparação de todas as operações com todas, utilizando a métrica Levenshtein (que é dada pelo número mínimo de operações necessárias para transformar uma string em outra).

A tabela A.1 apresenta os resultados encontrados nos experimentos realizados. As colunas MétodoS1 e MétodoS2 representam os nomes das operações (operation) do WSDL. A coluna Similaridade representa a semelhança dos nomes das operações em S1 e S2, utilizando a métrica citada acima (o resultado igual a um significa que os Strings são iguais). As outras métricas disponíveis na biblioteca Simmetrics mostraram resultados ainda mais longe do esperado.

A figura A.1 mostra em detalhes a implementação das classes através do diagrama de classes. A ClasseTeste possui os métodos percorreWsdl1, percorreWsdl2 e outputResult. Nos dois primeiros é feita a comparação (utilizando a métrica que foi citada anteriormente) de cada elemento da primeira lista, que contém o nome das operações do primeiro WSDL, com todos os elementos da segunda lista, que guarda o nome das operações do segundo WSDL fornecido. O terceiro método mostra para o usuário os resultados encontrados no percorreWsdl1 e percorreWsdl2.

Tabela A.1: Resultado das Comparações utilizando *Levenshtein*

Método S1	**Método S2**	**Similaridade**
GetWeather	GetWeatherByIP	0,7143
GetWeather	GetWeatherByZip	0,6667
GetWeather	GetWeatherByWMOID	0,5882
GetWeather	GetWeatherByCityState	0,4762
GetWeather	GetWeatherByStationID	0,4762
GetCitiesByCountry	GetWMOIDByCity	0,4444
GetCitiesByCountry	GetWeatherByCityState	0,4286
GetWeather	GetWeatherHistoricalByZip	0,4000
GetCitiesByCountry	GetFiveDayForecastByZip	0,3478
GetCitiesByCountry	GetWeatherByZip	0,3333
GetCitiesByCountry	GetWMOIDByCity	0,3333
GetCitiesByCountry	GetWeatherHistoricalByZip	0,2800
GetWeather	GetWarningsByState	0,2778
GetWeather	GetFiveDayForecastByZip	0,2609
GetWeather	GetWarningsByZip	0,2500

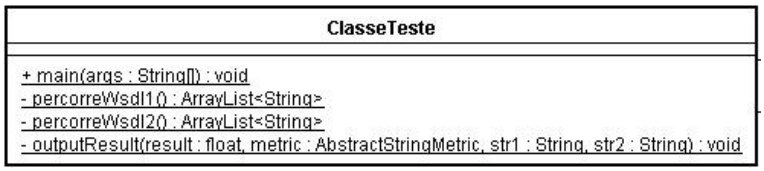


Figura A.1: Diagrama ClasseTeste

REFERÊNCIAS

BARROS, A.; DUMAS, M.; OAKS, P. **A Critical Overview of the Web Services Choreography Description Language (WS-CDL)**. Disponível em: $< http : //www.bptrends.com/publicationfiles/03-05WPWS-CDLBarrosetal.pdf >$. Acesso em: novembro 2009.

BOOTH, D. **Web Services Architecture**. Disponível em: $< http : //www.w3.org/TR/ws-arch/ >$. Acesso em: outubro 2009.

CALLOWAY, A. **Service-oriented Architecture (SOA)**. Disponível em: $< http : //www.accenture.com/NR/rdonlyres/3A5ED706-5ED6-4732-913F-E6B68FDD635C/0/AccentureCollaborateswithPepconSOAGovernancetoDriveHighPerformance.pdf >$ Acesso em: novembro 2009.

CHAPMAN, S. **Simmetrics**. Disponível em: $< http : //www.dcs.shef.ac.uk/sam/simmetrics.html >$. Acesso em: novembro 2009.

CHENG, S.; CHANG, C. K.; ZHANG, L.-J.; KIM, T.-H. Towards Competitive Web Service Market. In: INTERNATIONAL WORKSHOP ON FUTURE TRENDS OF DISTRIBUTED COMPUTING SYSTEMS, 11., 2007. **Proceedings...** [S.l.: s.n.], 2007.

CHINNICI, R.; GUDGIN, M.; MOREAU, J.-J.; WEERAWARANA, S. **Web Services Description Language (WSDL)**. Disponível em: $< http : //www.w3.org/TR/2002/WD-wsdl12-20020709 >$. Acesso em: novembro 2009.

CRUZ, S. M. S. da; CAMPOS, L. M.; CAMPOS, M. L. M.; PIRES, P. F. A Data Mart Approach for Monitoring Web Services Usage and Evaluating Quality of Service. In: XVIII BRAZILIAN SYMPOSIUM ON DATABASES, 2003. **Proceedings...** [S.l.: s.n.], 2003.

CUNHA, D. **Web Services, SOAP e Aplicações Web**. Disponível em: $< http : //devedge-temp.mozilla.org/viewsource/2002/soap-overview/index/_pt/_br.html >$. Acesso em: novembro 2009.

JURETA, I.; FAULKNER, S.; ACHBANY, Y.; SAERENS, M. Dynamic Web Service Composition within a Service-Oriented Architecture. In: IEEE INTERNATIONAL CONFERENCE ON WEB SERVICES, 2007. **Proceedings...** [S.l.: s.n.], 2007.

JURIC, M. B. **A Hands-on Introduction to BPEL**. Disponível em: $< http : //www.oracle.com/technology/pub/articles/matjaz/_bpel1.html >$. Acesso em: novembro 2009.

LEYMANN, F.; ROLLER, D.; THATTE, S. **Goals of the BPEL4WS Specification**. Disponível em: $< http : //xml.coverpages.org/BPEL4WS - DesignGoals.pdf >$. Acesso em: novembro 2009.

MAAMAR, Z.; WIVES, L. K.; ; AL-KHATIB, G. **From Communities of Web Services to Marts of Composite Web Services**. Draft - submitted to AINA 2010.

MAAMAR, Z.; WIVES, L. K.; TATA, S.; SELLAMI, M. **Towards a Marriage between Web Services and Recommender Systems**. Não publicado.

MANIKRAO, U. S.; PRABHAKAR., T. V. Dynamic Selection of Web Services with Recommendation System. In: INTERNATIONAL CONFERENCE ON NEXT GENERATION WEB SERVICES PRACTICE, 2005. **Proceedings...** [S.l.: s.n.], 2005.

MATTEW, B. **Business Process Execution Language for Web Services** : bpel and bpel4ws. [S.l.]: Packt Publishing, 2004.

MAY, B.; MARKOVSKY, D. **Understanding the Travel Reservation Service**. Disponível em: $< http : //netbeans.org/kb/55/understand_trs.html >$. Acesso em: novembro 2009, Sun NetBeans 5.5 Knowledge Base.

NOGUEIRA, J. **Coordenação e Composição de Web Services**. Disponível em: $< http : //www.gsd.inesc - id.pt/ ler/docencia/tm0607/slides/Coordenacao - Composicao - JoaoNogueira.pdf >$. Acesso em: novembro 2009.

RECKZIEGEL, M. **Entendendo os WebServices**. Disponível em: $< http : //imasters.uol.com.br/artigo/4245/webservices/entendendo_os_webservices/imprimir/ >$. Acesso em: novembro 2009.

SONDA, G. C.; MONTEZ, C. **Uma Proposta de Implementação de Diferenciação de Serviços na Arquitetura de Web Services**. Disponível em: $< http : //inf.unisul.br/ ines/workcomp/cd/pdfs/2508.pdf >$. Acesso em: novembro 2009.

STAL, M. **Web services**: beyond component-based computing. [S.l.]: ACM, 2002.

TURRELL, G. C. **Web services Description Languagem for Java (wsdl4j-14)**. Disponível em: $< http : //sourceforge.net/projects/wsdl4j >$. Acesso em: novembro 2009.

WIVES, L. K. **Dodona**: recomendação de serviços web. Descrição de Projeto Universal CNPQ.