

Relatórios Técnicos do Departamento de Informática Aplicada da UNIRIO n°0014/2009

Desenvolvendo web services no BEA Workshop for WebLogic Platform

Leonardo Guerreiro Azevedo Henrique Prado Sousa Fernanda Baião Flávia Santoro

Departamento de Informática Aplicada

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO Av. Pasteur, 458, Urca - CEP 22290-240 RIO DE JANEIRO – BRASIL

Projeto de Pesquisa

Grupo de Pesquisa Participante



Patrocínio



Relatórios Técnicos do DIA/UNIRIO, No. 0014/2009 Editor: Prof. Sean W. M. Siqueira

Agosto, 2009

Desenvolvendo web services no BEA Workshop for WebLogic Platform*

Leonardo Guerreiro Azevedo, Henrique Prado Sousa, Fernanda Baião, Flavia Santoro

Núcleo de Pesquisa e Prática em Tecnologia (NP2Tec) Departamento de Informática Aplicada (DIA) – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

azevedo@uniriotec.br, henrique.souza@uniriotec.br, fernanda.baiao@uniriotec.br, flavia.santoro@uniriotec.br

Abstract. Web services is the main technologic for service implementation in a SOA (Service Oriented Architecture) approach. Therefore, it is very important to know how to develop web services in a well known platform. The main goal of this work is to present a methodology for service development, as well as details about how to implement web services and execute query on databases. The BEA Workshop for Web-Logic Platform and PostgreSQL were chosen as IDE for service development and database to execute the queries, respectively.

Keywords: SOA, web services, modeling, BEA Workshop for WebLogic Platform, PostgreSQL.

Resumo. Web services é a principal tecnologia para implementação de serviços em uma arquitetura orientada a serviços (SOA – Service Orientede Architecture). Logo, conhecer bem a implementação de web services em uma plataforma amplamente utilizada é intrínseco para a implantação de SOA. Este trabalho tem o objetivo de apresentar uma metodologia para desenvolvimento de serviços, bem como detalhes para a codificação de serviços e conexão com banco de dados. Neste trabalho, a ferramenta escolhida para codificação de serviços foi a BEA Workshop for WebLogic Platform e o banco de dados PostgreSQL.

Palavras-chave: SOA, web services, modelagem, BEA Workshop for WebLogic Platform, PostgreSQL.

^{*} Trabalho patrocinado pela Petrobras.

Sumário

1	In	trodução	5	
	1.1	Motivação	5	
	1.2	Objetivo	5	
	1.3	Metodologia de desenvolvimento	5	
	1.4	Estrutura do Relatório	6	
2	Et	apas do desenvolvimento de serviços	6	
	2.1	Softwares utilizados	6	
	2.2 Passo a passo para implementação de um serviço de consulta a dados de			
		Unidade Operativa	6	
	2.3	Criação de DataSource para Postgresql	22	
3	Τe	estes do WebService implementado	24	
	3.1	Realização de testes utilizando o testClient do Workshop	24	
	3.2	Realização de testes utilizando o SOAPUI	29	
	3.3	Realização de testes implementando um cliente Java para o serviço	32	
4	Сс	onclusão	43	
5	Re	eferências	43	

1 Introdução

1.1 Motivação

A implantação de SOA em uma organização apresenta uma série de desafios, incluindo questões relacionadas a recursos de hardware e software (por exemplo, aplicações de software, dispositivos de hardware, servidores etc), infra-estrutura de ESB (*Enterprise Service Bus*) (tais como, segurança, integração de aplicações com processos de negócio etc), bem como questões relacionadas à modelagem, projeto, implementação e monitoramento e gestão de serviços [Papazoglou *et al.*, 2007].

Web services é a principal tecnologia para desenvolvimento de serviços [Erl, 2005]. Conhecer bem uma tecnologia para desenvolvimento de web services deve fazer parte de qualquer estratégia para implantação de uma arquitetura orientada a serviços (SOA).

1.2 Objetivo

O objetivo deste trabalho é apresentar detalhes da implementação de serviços na ferramenta BEA Workshop for WebLogic Platform¹, bem como detalhes do uso do componente componente Database Control² para acesso a banco de dados. Neste trabalho, o banco de dados PostgreSQL³ foi utilizado.

O componente Database Control permite o acesso simples a bancos de dados relacionais. O Database Control automaticamente traduz consultas enviados ao banco de dados em objetos Java, de modo que os resultados das consultas podem ser facilmente acessados a partir destes objetos.

1.3 Metodologia de desenvolvimento

A metodologia de desenvolvimento aqui apresentada é bottom-up, ou seja, o serviço é gerado a partir da necessidade de acesso ao banco de dados. Em outras palavras, a partir de uma consulta realizada na base de dados, a qual foi solicitada por uma demanda de acesso a dados.

A partir da consulta, é criado um arquivo XSD para representar a estrutura dos elementos a serem retornados, os quais são:

- Tipo complexo representando a estrutura de retorno da consulta;
- Tipo complexo para representar uma lista de registros;
- Variável para o tipo complexo que representa a lista de objetos.

Além disso, é criada uma classe POJO (*Plain Old Java Object*) para representar os objetos retornados pela consulta.

O componente Database Control é utilizado para consultar a base de dados. Este componente retorna um conjunto de objetos Java, os quais devem ser transformados

¹ http://edocs.bea.com/workshop/docs92/platform.html

² http://edocs.bea.com/workshop/docs81/doc/en/core/index.html

³ http://www.postgresql.org/

para um arquivo XML de acordo com a estrutura do arquivo XSD⁴. Este arquivo é retornado ao cliente que invocou o serviço.

O aplicativo SOAPUI⁵ é utilizado para testar o WebService. Testes também podem ser realizados utilizando o Test Client, disponível no Workshop for WebLogic Platform, ou mesmo através da implementação de um cliente específico para acessar o WebService implementado. Estas formas de realizar testes também são apresentadas neste trabalho.

1.4 Estrutura do Relatório

O relatório está estruturado em elementos pré-textuais, 5 (cinco) capítulos, e elementos pós-textuais. Na primeira parte há itens como sumário e índices. Em seguida, estão os capítulos que estão brevemente descritos a seguir, e, por fim, os elementos pós-textuais.

O capítulo 1 explica a motivação e o objetivo para este trabalho, bem como apresenta uma descrição em alto nível da metodologia utilizada atualmente para desenvolvimento de serviços.

O capítulo 2 apresenta o passo-a-passo para desenvolvimento de serviços na GDIEP.

O capítulo 3 é dedicado a apresentar possibilidades para testes de serviços.

O capítulo 4 apresenta as conclusões do presente trabalho, listando algumas possibilidades de melhoria na metodologia atual de desenvolvimento.

O capítulo 5 apresenta as referências bibliográficas.

2 Etapas do desenvolvimento de serviços

Este capítulo apresenta o passo-a-passo para desenvolvimento de serviços.

2.1 Softwares utilizados

Os seguintes softwares foram utilizados para a implementação dos serviços.

- 1. BEA Workshop for WebLogic Platform 9.02
- 2. JDK 1.5
- 3. PostgreSQL
- 4. SOAPUI

2.2 Passo a passo para implementação de um serviço de consulta a dados de Unidade Operativa

A seguir são apresentados os detalhes para desenvolvimento de web services.

1. Criar projeto "Web Service Project" em File → New → Other... → Web Service → Web Service Project. Ao avançar, na tela "Web service Project", o target run-

⁴ http://www.w3.org/XML/Schema

⁵ http://www.soapui.org/

time "BEA Weblogic v9.2" deve ser escolhida. Avançando novamente, em "Project Facets", a opção "Annotated Web Service Facets" deve estar marcada e a opção "XML Beans → XMLBeans Builder" deve ser marcada.

- a. Criar os seguintes pacotes no pacote "src":
 - i. Services: classes webservices, por exemplo, br.uniriotec.services
 - ii. Controls: classes controls, por exemplo, br.uniriotec.controls
 - iii. Beans: classes POJO (Plain-Old Java Object) correspondente aos objetos lidos do banco, por exemplo, br.uniriotec.beans
- 2. Elaborar consulta para a tabela (ou conjunto de tabelas) que se deseja consultar, em ferramenta cliente do banco de dados.
 - a. Por exemplo:

select nome, cpf, id from cliente;

b. A partir desta consulta, definir os atributos que serão utilizados pelo POJO. Por exemplo:

BigDecimal id; String cpf; String nome;

- 3. Um arquivo XSD deve ser criado definindo a estrutura dos objetos a serem transferidos na mensagem de resposta do WebService. O componente utilizado para conexão com o banco de dados é o Database Control disponibilizado pelo Workshop. Este componente retorna um array de objetos POJO. Entretanto, no tráfego de mensagens entre WebServices são enviadas mensagens XML. Logo, os objetos POJO devem ser transformados em uma estrutura XML, de acordo com o XSD definido. São especificados três elementos complexos no XSD:
 - a. Definição da estrutura de cada registro da tabela *cliente*

c. Definição de variável para a lista de elementos

<rs:element name="ClienteList" type="lstCliente" />

d. O arquivo XSD produzido correspondente à estrutura do XML que re-

presenta a classe *Cliente*, e deve ser criado no pacote "Schemas", seguindo-se o menu File \rightarrow New \rightarrow Other \rightarrow XML \rightarrow XML Schema. Preencher o nome do arquivo com "Cliente.xsd". Após criar o arquivo, o targetNamespace e o nameSpace devem ser ajustados (através da edição do xsd) para o pacote do elemento "http://controls.beans/cliente". Além disso, deve ser criado o atributo xmlns:wld="http://www.bea.com/2002/10/weblogicdata".

```
<?xml version="1.0"?>
<xs:schema targetNamespace="http://controls.beans/cliente"</pre>
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://controls.beans/cliente" elementFormDefault="qualified"
xmlns:wld="http://www.bea.com/2002/10/weblogicdata"
attributeFormDefault="unqualified">
<xs:complexType name="cliente">
      <xs:sequence>
            <xs:element name = "id" type="xs:decimal" nillable="true"</pre>
      minOccurs="0" maxOccurs="1" />
            <xs:element name = "nome" type="xs:string"</pre>
nillable="true" minOccurs="0" maxOccurs="1" />
            <xs:element name = "cpf" type="xs:string" nillable="true"</pre>
minOccurs="0" maxOccurs="1" />
      </xs:sequence>
</xs:complexType>
<xs:complexType name="lstCliente">
      <xs:sequence>
            <xs:element name="Cliente" type="cliente" minOccurs="0"</pre>
maxOccurs="unbounded" />
     </xs:sequence>
</xs:complexType>
<xs:element name="ClienteList" type="lstCliente" />
</xs:schema>
```

 No pacote Bean ("br.uniriotec.beans") criar classe POJO para representar o elemento <Unidade Operativa> - Todos os atributos criados são privados com métodos get e set (Figura 1).

```
package br.uniriotec.beans;
 import java.math.BigDecimal;

public class Cliente {

     BigDecimal id;
     String cpf;
     String nome;
Θ
     public String getCpf() {
         return cpf;
     }
Θ
     public void setCpf(String cpf) {
         this.cpf = cpf;
     3
\Theta
     public BigDecimal getId() {
         return id;
     3
\Theta
     public void setId(BigDecimal id) {
         this.id = id;
     }
     public String getNome() {
         return nome;
     }
\Theta
     public void setNome(String nome) {
         this.nome = nome;
     3
 }
```

Figura 1 – Código da classe POJO Cliente

- 5. A partir do XSD criar arquivo JAR com os tipos de dados.
 - a. No pacote "br.uniriotec.controls", clicar com o botão direito no arquivo Cliente.xsd → Web Services → "Generates Types JAR File..."
 - b. Escolher o tipo "Apache XML Beans".
 - c. O JAR criado fica na biblioteca do projeto, em "...\WebContent\WEB-INF\lib"
- 6. O próximo passo é criar um objeto Control para acessar a base de dados.
 - a. Clicar em File \rightarrow New \rightarrow JDBC Control
 - b. Nomear o objeto como: <nome do objeto> + DbControl (Figura 2), por exemplo, ClienteDbControl.

W New Control					
JDBC Control					
Create a new Beehive JDBC Control for simplified database access.					
Enter or select the parent folder:					
UniRioTec/src/br/uniriotec/controls					
⊞ - 1999 UnidadeOperativa					
⊡ ﷺ UniRioTec					
in the state of t					
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □					
beans					
Controis					
Control name: ClienteDbControl					
< <u>Back</u> <u>Next</u> <u>Finish</u> Cancel					

Figura 2 – Janela de criação do controle JBDC

Na criação do objeto Control, será solicitada a associação a um Data-Source. O DataSource deve estar associado a um servidor, que por sua vez deve estar associado a um domínio. Deverão ser criados e associados, em ordem: o domínio, o servidor, o DataSource e por fim, o Control. Caso algum destes componentes não esteja criado, siga os passos 6.1, 6.2 e 6.3, e ao concluir estes passos, retorne para executar a letra o passo a seguir (c). Senão, se o domínio, servidor e DataSource já estiverem configurados, continue no passo seguinte (c).

c. Definir a conexão a ser utilizada, clicando em Browser (Figura 3).

W New Control		8	
JDBC Control			
Create a new Beehive	JDBC Control for simplified database access.		
			^
Connection name for	this Database control		
Data source:		Browse	6
			E
Í	W Browse Server Resources		
	Select JDBC Resource		-
	ClienteDS		
			_
			-
		<u>F</u> inish	Cancel

Figura 3 – Janela para definição do data source

d. Substituir o código gerado pela SQL elaborada anteriormente.

IMPORTANTE: O nome dos campos de retorno da consulta (projeção da consulta) devem ser iguais aos nomes dos atributos da classe POJO criada (Figura 1). O Database Control utiliza estes nomes para fazer introspecção e transformar os registros lidos do banco de dados em objetos da classe Java. Logo, por exemplo, se na tabela "Cliente" o nome da coluna id fosse "cliente_id" então na SQL deveria ser utilizado um alias para que fosse retornado id, por exemplo, "cliente_id as id".

e. Na consulta podem ser utilizados os parâmetros: {sql: } para código SQL e {var} para variável, quando for o caso.

100	1100 101	o cabo.	
ii ii	$\{sql:$	p_where} "	+
"	sql:	p_groupby}	" +
"	sql:	p_having}	" +
"	$\{sql:$	p_orderby}	")

f. Criar método que retorna um array de elementos. Cliente[] getCliente(String p_where,

> String p_groupby, String p_having, String p_orderby) throws

SQLException;

```
g. É necessário importar a classe Cliente e SQLException.
import java.sql.SQLException;
import br.uniriotec.beans.Cliente;
```

h. A classe resultante é apresentada na Figura 4.

```
package br.uniriotec.controls;
import org.apache.beehive.controls.system.jdbc.JdbcControl;
 import org.apache.beehive.controls.api.bean.ControlExtension;
 import java.sql.SQLException;
 import br.uniriotec.beans.Cliente;
⊖@ControlExtension
@JdbcControl.ConnectionDataSource(jndiName = "ClienteDS")
public interface ClienteDbControl extends JdbcControl {
\Theta
     @JdbcControl.SQL(statement="select nome, " +
                           cpf," +
                     .....
                     .
                          id" +
                     " FROM cliente" +
                     " Where 1=1 " +
                     " {sql: p where} " +
                     " {sql: p_groupby} " +
                     " {sql: p_having} " +
                     " {sql: p_orderby} ")
     Cliente[] getCliente(String p where,
                 String p_groupby,
                 String p having,
                 String p_orderby) throws SQLException;
     static final long serialVersionUID = 1L;
```

Figura 4 – Classe ClienteDbControl

- 6.1 Um domínio pode ser criado utilizando o "BEA Web Logic Cofiguration Wizard", que pode ser acessado através do menu padrão "BEA Products/Tools/Configuration Wizard" ou no wizard disponibilizado ao iniciar um servidor no Workshop.
 Para criar o domínio através do "BEA WebLogic Cofiguration Wizard" faca:
 - a. Avançar na primeira tela (Figura 5).

BEA WebLogic Configuration Wizard	- 0 💌
Welcome Choose between creating and extending a domain. Based on your selection, the Configuration Wizard guides you through the steps to generate a new or extend an existing domain.	Kbea 🔨
Create a WebLogic domain in your projects directory.	
O Extend an existing WebLogic domain	
Extend an existing WebLogic domain.	
Use this option to add applications and services, or to override existing database access (JDBC) an messaging (JMS) settings. You can also incorporate additional functionality in your domain, for exa including AquaLogic Service Bus.	d mple, by
	Previous Next

Figura 5 – Janela para criação de um novo domínio

b. Selecionar o template padrão e avançar (Figura 6).

elect Domain Source		
lect the source from which the domain will be cre the required components or by selecting from a	ated. You can create the domain by clicking list of existing domain templates.	őbea [.]
○ Generate a doma	in configured automatically to support the follo	wing BEA products:
WebLogic Serve	r (Required)	
U Workshop for V	/ebLogic Platform	
🔲 WebLogic Integ	ration	
🗌 WebLogic Porta		
WebLogic Porta	I GroupSpace Framework	
🗌 WebLogic Porta	l GroupSpace Application	
• Base this domain	on an existing template	
Template location:	C:\bea\weblogic92\common\templates\domains\wls.jar	Browse

Figura 6 – Janela para seleção do template padrão

c. Inserir senha padrão (weblogic) e clicar em "Next" (Figura 7).

BEA WebLogic Configuration Wizard						
Configure Administrato Create a user to be assigned to t this user is the default administra	ponfigure Administrator Username and Password eate a user to be assigned to the Administrator role. s user is the default administrator used to start development mode servers.					
💍 Discard Changes						
*User name:	weblogic					
*User password:	*****					
*Confirm user password:	жжжжжж					
Description:	This user is the default administrator.					
1						
Exit Help		Previous Next				

Figura 7 – Janela para configuração de nome de usuário e senha

d. Inserir um nome para o domínio e clicar em "Create" (Figura 8).

BEA WebLogic Configuration Wizard			
Create WebLogic Domain When you click Create the domain will be generated	f in the location specified.		a bea
Enter the na	ne and location for the domain:		
Domain name:	Dominio_Servicos		
Domain location:	C:\bea\user_projects\domains	Browse	
Exit Help			Previous Create

Figura 8 – Janela de configuração do domínio

e. Os domínios são criados por padrão no diretório "C:\bea\user_projects\domains", ou seja, neste diretório será criado o domínio Dominio_Servicos. Vá até este diretório e certifique-se que ele foi criado.

Para saber mais sobre domínios e criação de domínios, acessar http://edocs.bea.com/platform/docs81/confgwiz/intro.html#1052972.

- 6.2 Para criar o servidor faça:
 - a. No BEA Workshop for WebLogic Platform, abrir a janela de criação do servidor em File → New → Server, escolher o servidor BEA WebLogic v9.2 Server e clicar em "Next" (Figura 9).

W New Server	X
Define a New Server	
Choose the type of server to create	
Server's <u>h</u> ost name: localhost	•
Select the <u>s</u> erver type:	
🕀 🦢 Apache	
EA Systems, Inc.	
IGes BEA WebLogic v9.2 Server	
1	View But Vender
Description REA Webl exists Q 2 Server	view by: vendor
Description: BEA WebLogic V9.2 Server	
Server runtime: BEA WebLogic v9.2	→ Installed Runtimes
< <u>B</u> ack <u>N</u> ext >	Einish Cancel

Figura 9 – Janela para criação de novo servidor

b. Escolher o domínio para instalar o servidor. Por padrão, os domínios estão instalados em C:\bea\user_projects\domains. Ex: O domínio criado na sessão anterior está em C:\bea\user_projects\domains\Dominio_Servicos. Após inserir o domínio, clicar em "Next" (Figura 10).

W New Server	×
BEA WebLogic v9.2 Server	-
Define a WebLogic Server 🖇 🖇	ea
Domain home: C:\bea\user_projects\domains\Dominio_Servicos	se
Click here to launch Configuration Wizard to create a new dor	<u>main</u>
Name	
Server name: BEA WebLogic v9.2 Server [Dominio Servicos]	_
< <u>Back</u> <u>N</u> ext > <u>Finish</u> Cancel	

Figura 10 – Janela para configuração do domínio

c. Adicionar o projeto que está sendo desenvolvido para configuração do mesmo no servidor e clique no botão *Finish* (Figura 11).

🔞 New Server			—
Add and Remove Projects Modify the projects that are con	s figured on the server		
Move projects to the right to con	figure them on the serve	r	
<u>Available projects:</u>		Configured projects:	
		🐻 UniRioTec	
	A <u>d</u> d >		
	< <u>R</u> emove		
	Add A <u>l</u> I >>		
	<< Re <u>m</u> ove All		
 < <u>B</u> ack	Next >	Einish	Cancel

Figura 11 – Janela para adição de projetos para configuração no servidor

- 6.3 A configuração do DataSource pode depender do banco de dados a ser usado. Abaixo segue os passos de uma configuração genérica que pode não funcionar com todos os bancos. Para maiores informações, verificar a documentação específica da versão que pretende usar. Na sessão 6.3 encontra-se o passo a passo para a criação do DataSource para o Postgres.
 - a. Obter o driver JDBC do banco que será utilizado.
 - b. Copiar o driver JDBC para C:\bea\weblogic92\server\lib.
 - c. Abrir para edição o arquivo "startWebLogic.cmd" (com o notepad, por exemplo) que se encontra na pasta Bin, dentro da pasta do domínio que está sendo utilizado. Ex: C:\bea\user_projects\domains\Dominio_Servicos\bin\startWebL ogic.cmd
 - d. Incluir o caminho para o diretório onde o driver JDBC do banco foi salvo na linha onde é referenciada a variável CLASSPATH, por exemplo:

```
set
SAVE_CLASSPATH=%CLASSPATH%;C:\bea\weblogic92\ser
ver\lib\postgresql-8.3-604.jdbc3.jar
```

e. Salvar o arquivo e executá-lo.

Obs: Cada domínio criado possui um "startWebLogic.cmd " distinto. Para acessar o console do servidor, é necessário que este arquivo seja executado. Através do menu de instalação padrão, é possível encontrar o atalho para o "WebLogic Server 9.2", que ao ser executado, abrirá uma janela de console semelhante . Entretanto, esse atalho NÃO aponta para o arquivo do domínio que foi criado pelo usuário. Portanto sempre execute o "startWebLogic.cmd" de dentro da pasta do domínio desejado.

- f. Quando o servidor permanecer em "RUNNING MODE", ou seja, quando for exibida a mensagem "<Server started in RUNNIN mode>", executar o browser e abrir o console na URL http://localhost:7001/console, com o nome de usuário padrão weblogic e senha weblogic.
- g. Na janela principal do console, clicar em "Lock and Edit" (Figura 12).

Chea WEBLOGIC SERVI								
Change Center	Welcome, weblogic	(Connected to: Dominio_Servicos	🟠 Home	Log Out	Preferences	Help	AskBEA
View changes and restarts	Home							
No pending changes exist. Click the Release Configuration button to allow others to edit the domain.	Domain Information and Resources Helpful Tools	General Information						
Lock & Edit > Configure applications Release Configuration > Recent Task Status		Common Administration Task Descriptions Set your console preferences Seed the documentation						
Domain Structure Dominio_Servicos								
B-Environment Deployments Services Security Realms B-Interoperability B-Diagnostics	Domain Configurations Domain	Services Messaging > JMS Servers	Interoperability WTC Servers Jolt Connection Pools					
How do L Use the Change Center View pending changes Release the configuration lock Change Console preferences Monitor servers	Environment © Servers © Clusters © Virtual Hosts © Migratable Targets © Machines	 > Store-and-Forward Agents > JMS Modules > Bridges > DDBC > Data Sources > Multi Data Sources > Data Source Factories 	Diagnostics © Log Files © Diagnostic Modules © Diagnostic Images © Archives					
System Status	Work Managers Startup And Shutdown Classes	Persistent Stores Path Services XML Registries XML Entity Caches	Context SNMP Agent Proxies Monitors					

Figura 12 – Tela de configuração do domínio criado Dominio_Servicos

- h. Abrir a configuração do DataSource dentro do frame Domain Configurations, em Services/JDBC/Data Sources.
- i. Na janela que abrir, clicar em New.
- j. Inserir um nome para o DataSource em "Name". Copiar o mesmo nome e inserir em "JNDI Name". Selecionar o tipo do banco de dados em "DataBase Type" e selecionar o driver em "DataBase Driver" (Figura 13). Por fim, clicar em "Next".

Change Center	Welcome, weblogic	Connected to: Dominio_Servicos
View changes and restarts	Home > Summary of JDBC Data Sources	
No pending changes exist. Click the Release Configuration button to allow others to edit the domain.	Create a New JDBC Data Source	cel
Lock & Edit Release Configuration	JDBC Data Source Properties The following properties will be u	used to identify your new JDBC data source.
Domain Structure	What would you like to name	your new JDBC data source?
Dominio_Servicos	What JNDI name would you like	ClienteDS te to assign to your new JDBC Data Source? ClienteDS
How do L	JNDI Name:	
Create JDBC data sources Create LLR-enabled JDBC data sources	What database type would you	u like to select?
System Status	Database Type:	PostgreSQL
Health of Running Servers	What database driver would y	ou like to use to create database connections?
Failed (0) Critical (0)	Database Driver:	PostgreSQL's Driver (Type 4) Versions:Any
Overloaded (0)	Back Next Finish Can	cel

- k. Na janela "Transaction Options" clicar em "Next".
- Preencher as informações de conexão com o banco de dados (Figura 14) e clicar em "Next".

view changes and restarts		
No pending changes exist. Click the Release Configuration button to allow others to edit the	Create a New JDBC Data Source	
domain.	Back Next Finish C	Cancel
Lock & Edit	Connection Properties Define Connection Properties.	
Domain Structure	What is the name of databa	se you would like to connect to?
Dominio_Servicos	Database Name:	UniRioTec
Environment Deployments Services	What is the name or IP add	ress of the database server?
	Host Name:	localhost
How do L	What is the port on the data	base server used to connect to the database?
Create JDBC data sources	Port:	5432
Create LLR-enabled JDBC data sources	What database account use	r name do you want to use to create database connections?
System Status	Database User Name:	postgres
Health of Running Servers	What is the database accou	nt password to use to create database connections?
Failed (0)	Password	
Critical (0)		
Overloaded (0)	Confirm Password:	
Warn (0)		
OK (1)	Back Next Finish C	Cancel

Figura 14 – Configuração da conexão com o banco de dados

m. Inserir os parâmetros de configuração do driver de conexão com o

banco. Testar a configuração clicando em "Test Configuration" (Figura 15). Caso ocorra algum erro, rever as configurações. Clicar em Next e depois em Finish.

No pending changes exist. Click the Release Configuration button	Create a New JDBC Data Source
to allow others to edit the domain.	Test Configuration Back Next Finish Cancel
Lock & Edit	Test Database Connection
Release Configuration	Test the database availability and the connection properties you provided.
Domain Structure	What is the full package name of JDBC driver class used to create database connections in the connection pool? (Note that this driver class must be in the classpath of any server to which it is deployed.)
Dominio_Servicos ⊕…Environment	Driver Class Name: org.postgresgl.Driver
Deployments Services	
Security Realms	What is the URL of the database to connect to? The format of the URL varies by JDBC driver.
■ Diagnostics	URL: jdbc:postgresql://localhost
How do I	What database account user name do you want to use to create database connections?
Create JDBC data sources	Database User Name: postgres
sources	What is the database account accound to use to create database connections?
System Status	(Note: for secure password management, enter the password in the Password field instead of the Properties field below)
Health of Running Servers	Password:
Failed (0)	Confirm Password:
Critical (0)	
Overloaded (0)	What are the properties to pass to the JUBC driver when creating database connections?
Warn (0)	user=postgres
OK (1)	Properties:

Figura 15 – Teste da configuração do Datasource

n. Para finalizar, clicar em "Activate changes" para aplicar a nova configuração (Figura 16).

WEBLOGIC SERVI Administration consc	ER									
Change Center	Welcome, weblogic	ome, weblogic Connected to: Dominio_Servicos 🗳 Home Log Out Preferences Help AskBEA								
View changes and restarts	Home > Summary of JDBC Data Sources									
Pending changes exist. They must be activated to take effect.	Summary of JDBC Data Sources	immary of JDBC Data Sources								
Activate Changes Undo All Changes Domain Structure	A JDBC data source is an object bound to the JNDI tree th source on the JNDI tree and then borrow a database con This page summarizes the JDBC data source objects that	A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool of JDBC connections. Applications can look up a data source on the JNDI tree and then borrow a database connection from a data source. This page summarizes the JDBC data source objects that have been created in this domain.								
Dominio_Servicos	Customize this table									
Deployments	Data Sources	Data Sources								
Security Realms	New Delete	New Delete Showing 1 - 1 of 1 Previous Next								
Diagnostics	🔲 Name 🗞 🛛 J	INDI Name	Targets							
How do I	ClienteDS C	ClienteDS								
 Create JDBC data sources Delete JDBC data sources 	New Delete		Showin	g 1 - 1 of 1 Previous	Next					

Figura 16 – Ativação das mudanças realizadas no servidor

- 7. Criar serviço com o padrão de nomenclatura <nome do objeto>+ Service, por exemplo, ClienteService.
 - a. No pacote "br.uniriotec.services" clicar em "New → WebLogic Web Service" (Figura 17).

W New Web Service	x
Web Service	
Create a new web service.	
Enter or select the parent folder:	
UniRioTec/src/br/uniriotec/services	
$rac{1}{2}$	_
🖃 🕵 UniRioTec	
È	
E (→ br	
beans	
controls	
services	
File name: ClienteSenvice	_
riena <u>m</u> e: Johenteservice	
	_
<u>F</u> inish Cancel	

Figura 17 – Criação do web service

b. Incluir referência para o objeto Control criado anteriormente.

```
@Control
ClienteDbControl clienteDbControl;
c. É necessário importar ClienteDbControl e beehi-
ve.controls.api.bean.Control.
import org.apache.beehive.controls.api.bean.Control;
```

```
import org.apache.beehive.controls.api.bean.Control
import br.uniriotec.controls.ClienteDbControl;
```

d. Criar WebMethod para retornar o elemento que representa o Array de objetos (Figura 18).

Na implementação deste método, a rotina verificarStringNull é utilizada para verificar se a string recebida como parâmetro é nula (Figura 19). O método parseCliente2XML é responsável por transformar cliente PO-JO em cliente XML (Figura 20). O XML dos clientes é o que será retor-

nado para o consumidor do serviço. O método parseCliente2XML invoca o método parsePOJO2XML, o qual é um método genérico capaz de transformar uma classe POJO em elementos de um XML, seguindo o padrão de nomes especificado (Figura 21). Observe que, neste método, existe um teste se ele contém a palavra "uniriotec". Isto é usado para garantir que o método só trate classes definidas no namespace "uniriotec".

```
return xmlCliente;
```

3

Figura 18 – Método getCliente

```
/** Verifica se uma string é nula e levanta uma exceção caso seja.
 * 
  * Uma das verificações que todo serviço deve fazer é se os parâmetros de
  * entrada são válidos. Este método verifica se a string passada como
  * parâmetro é nula. Caso seja nula, ele levanta uma exceção.
  * 
   * @param str string que se deseja verificar.
  * @param parameterName o nome do parâmetro do serviço, para a composição da
          mensagem de erro.
  *
  * @throws IllegalArgumentException caso a string seja nula.
  */
 public static void verificarStringNull(String str, String parameterName)
   throws IllegalArgumentException {
   if (str == null || str.trim().length() == 0)
     throw new IllegalArgumentException ("Parameter " + parameterName
       + " is required.");
 }
```

Figura 19 – Método verificarStringNull

```
private ClienteListDocument parseCliente2XML( Cliente[] cliente ) throws Exception {
    ClienteListDocument pDoc = ClienteListDocument.Factory.newInstance();
    pDoc.addNewClienteList();
    for (int i=0; i < cliente.length; i++) {
        parsePOJ02XML(cliente[i], pDoc.getClienteList().addNewCliente());
    }
    return pDoc;
}</pre>
```

Figura 20 – Método parseClente2XML

```
public static void parsePOJO2XML(Object pojo, Object xml) throws Exception {
  if (pojo == null) {
   throw new IllegalArgumentException("pojo == null");
  if (xml == null) {
    throw new IllegalArgumentException("xml == null");
  // Referências feitas fora do loop para otimizar chamadas.
 Class pojoClass = pojo.getClass();
 Class xmlClass = xml.getClass();
 Method[] methods = pojoClass.getMethods();
  for (Method m : methods) {
   String methodName = m.getName();
    // Trata apenas métodos get que pertencem as classes definidas no
    // namespace da uniriotec.
   if (methodName.startsWith("get")
      && m.getDeclaringClass().getName().contains("uniriotec")) {
      // Obtém o método get do atributo da classe "pojo".
     Method mget = pojoClass.getMethod(methodName, new Class[] {});
      // Obtém o método set do atributo respectivo na classe xml.
     methodName = "s" + methodName.substring(1);
     Method mset = xmlClass.getMethod(methodName, mget.getReturnType());
      // Obtém o atributo efetivamente retornado.
     Object getted = mget.invoke(pojo, new Object[] {});
      // Se o objeto não for nulo, atualiza o objeto xml.
     if (getted != null) {
       mset.invoke(xml, new Object[] { getted });
     3
   }
 }
3
```

Figura 21 – Método parsePOJO2XML

e. Importar Cliente e ClienteListDocument. import br.uniriotec.beans.Cliente; import beans.controls.cliente.ClienteListDocument;

i. O objeto ClienteListDocument encontra-se no jar criado a partir do XSD, por exemplo, ClienteTypes_xmlbeans_apache.jar.

2.3 Criação de DataSource para Postgresql

Caso deseje-se utilizar o SGBD PostgreSQL como banco de dados para os testes, os seguintes passos devem ser realizados.

1. Iniciar o servidor no domínio criado, por exemplo, GDIEP_domain (Figura 22).

<pre>return siglaUnidadeOper } public void setSiglaUnidade this.siglaUnidadeOperat } </pre>	ativa; Operativa(String siglaUnic iva = siglaUnidadeOperati	dadeOperativa) { va;
Problems 웨 Servers 🛛 Javadoc Tasks		
Server	Status	State
🚱 BEA WebLogic v9.2 Server [GDIEP_domain]	Started	Republish



2. Executar Admin Server Console (Figura 23).



Figura 23 – Menu para execução do Admin Server Console

3. Após logar no servidor, ir para a página de configuração de DataSources (Figura 24).

Change Center	Welcome, weblogic		Connected to: GDIEP_domain	🕼 Home 🛛 L	og Out Preference:	s Help As	skBEA
View changes and restarts	Home > Summary of JDBC Data Sour	rces					
Click the Lock & Edit button to modify, add or delete items in this domain.	Messages	vated. No restarts are necessary.					
Lock & Edit Release Configuration	Summary of JDBC Data Sou	urces					
Domain Structure	A 1DBC data source is an ob:	ject bound to the INDI tree that provide	s database connectivity through a poo	l of JDBC connect	ions. Applications can	look un a data	a
GDIEP_domain	source on the JNDI tree and This page summarizes the J Customize this table Data Sources Click the Lock & Edit butt	then borrow a database connection fro DBC data source objects that have beer	m a data source. created in this domain. the buttons on this page.				
GDIEP_domain B-Environment Deployments Services B-Messaging Data Sources - Data Sources - Multi Data Sources - Partisent Stores - Partisent Stores - Partisenvices -	Source on the JNDI tree and This page summarizes the JN Customize this table Data Sources Click the Lock & Edit butt New Delete	then borrow a database connection fro DBC data source objects that have beer ton in the Change Center to activate all	m a data source. created in this domain. the buttons on this page.		Showing 1 - 6 of 6	Previous Next	
DIEP_domain Deployments Deployments Services Descrices Data Sources Data Sources Data Sources Data Sources Persistent Stores Persites Foreign JNDI Providers Work Contexts XMR. Registries	Source on the JNDI free and This page summarizes the JI Customize this table Data Sources Click the Lock & Edit butt New Delete Rame &	then borrow a database connection fro DBC data source objects that have been ton in the Change Center to activate all INDI Name	m a data source. created in this domain. the buttons on this page.		Showing 1 - 6 of 6	Previous Next Targets	-
CDIEP_domain B-Environment Deployments Services B-Messaging Data Sources - Multi Data Sources - Data Source Factories - Path Services - Parth Services - Foreign JNDI Providers - Work Contexts - XMIL Registries - XMIL Entity Caches - Scom	Source on the JNDI free and This page summarizes the JI Customize this table Data Sources Click the Lock & Edit butt New Delete Hame & bpmArchDataSource	then borrow a database connection fro DBC data source objects that have beer ton in the Change Center to activate all INDI Name bpmArchDataSource	m a data source. created in this domain.		Showing 1 - 6 of 6	Previous Next Targets AdminServer	

Figura 24 – Página de configuração de data sources

- 4. Clicar em "Lock & Edit" e então no botão "New" em "Data Sources"
- 5. Preencher as informações para configuração do Data Source, testar a conexão, escolher o servidor e concluir a configuração.
- 6. Importante:
 - a. No caso do WebLogic Server 9.2, o driver do postgres é o da versão JDBC3 postgresql-8.3-604.jdbc3.jar.
 - b. Este arquivo deve estar configurado no Classpath do servidor.
 - i. Em "C:\bea\user_projects\domains\Domain_Servicos\bin", abrir o arquivo startWebLogic.cmd no notepad.
 - ii. Incluir o caminho para o diretório onde o driver foi salvo na linha onde é referenciada a variável CLASSPATH, por exemplo:

set SAVE_CLASSPATH=%CLASSPATH%;C:\bea\weblogic92\ser ver\lib\postgresql-8.3-604.jdbc3.jar

3 Testes do WebService implementado

Este capítulo descreve as possibilidades de testes de serviços.

3.1 Realização de testes utilizando o testClient do Workshop

1. No próprio Workshop, executar o serviço no servidor (Figura 25)

🖻 🥵 UniRioTec					import org.apaci
Jointol rec Jointol r	New Open Open With Open Type Hierarchy Paste >> Delete Build Path Source Insert Refactor >> Import 2* Export References Declarations	F3 F4 Ctrl+C Ctrl+V Alt+Shift+S Alt+Shift+T	* *	H	Import br.unirid import br.unirid import breans.com ©@WebService public class Cl: @Control ClienteDbCom © @WebMethod public Client @WebMethod public Client @WebMethod Public Client @WebMethod @Web
	🔗 Refresh	F5			
	Run As Debug As Upgrade Source File(s) Team Compare With Replace With Restore from Local Histor	<i>i</i>	 Image: 1 minipage Image: 2 minipage<	Run on Se Java Appl Java Appl SWT Appl Run	et Alt+Shift+X, R et Alt+Shift+X, A ication Alt+Shift+X, J ication Alt+Shift+X, S
br.uniriotec.services.0	Properties	Alt+Enter][

Figura 25 – Execução do serviço no servidor

2. Caso necessário, confirme o servidor que executará o serviço e clique em "Next" (Figura 26).



Figura 26 – Janela para escolha do servidor que executará o serviço

3. Adicione o projeto que será executado (Figura 27) e clique em "Finish".

Move projects to the right to	configure them on the se	rver Configured project	te
	_	UniRioTec	
	A <u>d</u> d >		
	< <u>R</u> emove		
	Add All >>	1	
	<< Ke <u>m</u> ove All		

Figura 27 – Janela para adição do projeto a ser executado

4. Após a publicação do serviço no servidor, a tela do Test Client deverá ser exibida (Figura 28). Caso ocorra algum erro, verifique no console do servidor se a botão "Release Configuration" está habilitado, conforme a Figura 29. Se estiver, clique nele para liberar a configuração e retorne novamente ao primeiro passo.

👿 Workshop - Test Client - BEA Workshop for	r WebLogic Platforn	n							- 6 -
<u>File Edit Source Refactor Navigate Se</u>	e <u>a</u> rch <u>P</u> roject <u>R</u> u	un <u>W</u> indow <u>H</u> elp							r
] 📬 🕶 🛛 🕾 👌 🖉 🕶 🖓 🕶 🛛 🖓 🕶 🖓 🛃	🖶 🞯 🕶] 🥭	🥔] 🕲] 😤] 🍫	⇔ • ⇔ •					1	😭 🕅 Workshop
📲 Package Explorer 🛛 🖳 🗖	Cliente.java	ClienteDbContro	I 🐼 ClienteService.ja	ClienteService.java	I Test Client	1 🕅 🕺 "1	- 0	Properties	»2 "D
(a a 🔳 🔗	http://localhost:7001/	wls_utc/?wsdlUrl=http%3A%2f	%2Flocalhost%3A7001%2FUniRio	ioTec%2FCliente	eService%3 👻 🕨	-		🗄 🏇 🖾 🌣
🗄 🕵 UniRioTec 🔹	1000				control of	-		Property	Value
E- 🕮 src	l bea				WebLo	gic Test Clier	nt 📄	🖃 Info	
E B Cliente inc						Chooco		derived	false
	Show Opera	ations	http://localhost:70	01/UniRioTec/Clientes	Service?	lanother		editable last modif	true £ 29/07/09 21:05
🕀 🖟 ClienteDbControl.java			WSDL			WSDL	10	linked	false
E Br.uniriotec.services	Message Lo	g - refresh						location	C:\bea\user_projects\
ClienteService.java	Cle	ar Log	Operations					name	ClienteService.java
🖶 🛋 JRE System Library [BEA WebLo								path	/UniRioTec/src/br/unir
🗈 🛋 BEA WebLogic v9.2 [BEA Produ-			getCliente					size	4232
Web App Libraries			string						
Webl egic I2EE Library July con			p_cpf:						
WebLogic J2EE Library [webLogic J2EE]			getCl	iente					
+ A beehive-controls.iar - C:\be									
beehive-ejb-control.jar - C:\									
🕀 🛃 beehive-jdbc-control.jar - C									
😨 📠 beehive-jms-control.jar - C:									
🗄 🍒 commons-codec-1.3.jar - C									
🕀 🔬 commons-discovery.jar - C:							+		
Commons-loading.iar - C:\t		~ 1	li internet interne					<u> </u>	
	Problems 4% Se	rvers 🕅 Javadoc Ta	asks Search					Ø (
Design View Palette	Server	Sta	tus	State					
	🕀 🎧 BEA WebL	ogic v9.2 Server [Doi 🌇	Started	Synchronized					
							12		
br.uniriotec.services.ClienteService.	java - UniRioTec/sr	c							

Figura 28 – Tela do Test Client, para o teste do serviço

Change Center	Welcome, weblogic		Connected to: Dor
View changes and restarts	Home		
No pending changes exist. Click the Release Configuration button	Domain		
to allow others to edit the domain.	Information and Resources Helpful Tools	General Information	
Lock & Edit Release Configuration	 Configure applications Recent Task Status 	 Common Administration Task Des Set your console preferences Read the documentation 	scriptions
Domain Structure Dominio_Servicos -Environment			
	Domain Configurations Domain	Services	Interoperability
⊡Interoperability ⊡Diagnostics	Domain	 Messaging > JMS Servers 	WTC Servers Jolt Connection Pools
How do I		> Store-and-Forward Agents	

Figura 29 – Liberação da configuração necessária (Release Configuration)

- 5. Preencher o parâmetro do método, por exemplo, p_cpf, clicar no botão referente ao método e analisar o resultado.
 - a. A Mensagem de requisição de execução do método getCliente gerada pelo Test Client (Service Request) e a mensagem de resposta (Service response) são apresentadas na Figura 30.



6. Caso não seja preenchido o parâmetro do método para realizar a consulta, é retornada uma mensagem de erro (Figura 31 e Figura 32).





<pre>csoapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> csoapenv:Header /> csoapenv:Header /> csoapenv:Fault> csoapenv:Fault+fotopenvicesComponent+fault(ClientEService.java:51) at</pre>	Service Response
<pre>csoapenv:Envelope xmms:soapenv=nttp://schemas.xmsoap.org/soap/envelope/> csoapenv:Header/> csoapenv:Fault> csoapenv:Fault> cfaultstring>Failed to invoke end componentFailed to invoke methodParameter p_cpf is required.</pre> cfaultstring> cdetail>	
<pre>csoapenv:Bodys csoapenv:Bodys csoapenv:Bodys csoapenv:Bodys csoapenv:Bodys csoapenv:Bodys cfaultcode>soapenv:Server cfaultstring>Failed to invoke end componentFailed to invoke methodParameter p_cpf is required.</pre> faultstring>Failed to invoke end componentFailed to invoke methodParameter p_cpf is required. cdetails cbea_fault:stacktrace xmins:bea_fault="http://www.bea.com/servers/wls70/webservice/fault/1.0.0">java.lang.IllegalArgumentException: Parameter p_cpf is required. at br.uniriotec.services.ClienteService.yetCliente(ClienteService.java:21) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:25) at weblogic.wsee.ws.dispatch.server.Component.invoke(JavaClassComponent.java:29) at weblogic.wsee.ws.dispatch.server.Component.Handler.terator.java:127) at weblogic.wsee.ws.dispatch.server.Component.thandler.terator.java:127) at weblogic.wsee.ws.dispatch.server.Component.fandleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.component.fandleRequest(SoapProcessor.java:127) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServletSarutorizedInvoke.run(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServletServletServlet.java:291 at weblogic.servet.internal.StubSecurityHelper/servletServletServletServletServlet.java:233 at weblogic.servet.internal.StubSecurityHelper.invokeServletStubImpl.java:233 at weblogic.servet.internal.ServletStubImpl.excute(ServletStubImpl.java:125) at weblogic.servet.internal.StubSecurityHelper.invokeServletStubImpl.java:233 at weblogic.servet.internal.StubSecurityHelper.invokeServletStubImpl.java:233 at weblogic.servet.internal.StubSecurityHelper.invokeServletStubImpl.java:233 at weblogic.servet.internal.StubSecurityHelper.invokeServletStubImpl.java:233 at weblogic.servet.internal.StubSecurityHelper.invokeServletStubImpl.java:233 at weblogic.servet.internal.Stub	<pre><soapenv:envelope xmins:soapenv="nttp://schemas.xmisoap.org/soap/envelope/"></soapenv:envelope></pre>
<pre>csoapenv:Fault> <soapenv:fault> <faultcode>soapenv:Server</faultcode> <faultstring>Failed to invoke end componentFailed to invoke methodParameter p_cpf is required.</faultstring> <details< td=""><td><sopenv:header></sopenv:header></td></details<></soapenv:fault></pre>	<sopenv:header></sopenv:header>
<pre>csoapenv:FaultS <faultstring>Failed to invoke end componentFailed to invoke methodParameter p_cpf is required.</faultstring> <details cdetails cdetails thea_fault:stacktrace</details </pre> <mins:bea_fault:stacktrace </mins:bea_fault:stacktrace symIns:bea_fault=http://www.bea.com/servers/wis70/webservice/fault/1.0.0">java.lang.IllegalArgumentException: Parameter p_cpf is required. at br.uniriotec.services.ClienteService.getClienteClienteService.java:51) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) 	<soapenv:body></soapenv:body>
<pre>cfaultcode>Solpenv:Server </pre> <pre>cfaultcode>Solpenv:Server</pre> <pre>cfaultcode>Solpenv:Server</pre> <pre>cfaultcode>Solpenv:Server</pre> <pre>cdetail> </pre> <pre>cdetail</pre>	<soapenv:faut></soapenv:faut>
<pre>clautisting>clautisting> clautisting>required.</pre> (faultisting> clautisting>clautisting> clautisting>clautisting> clautisting>clautisting> clautisting>clautisting> clautisting>clautisting> clautisting>clautistick Parameter p_cpf is required. at br.unirotec.services.ClienteService.yetfCliente(ClienteService.java:51) at br.unirotec.services.ClienteService.getCliente(ClienteService.java:21) at sun.reflect.NativeMethodAccessorImpl.invoke0(NativeMethod) at sun.reflect.NativeMethodAccessorImpl.invoke0(NativeMethodAccessorImpl.java:39) at sun.reflect.NativeMethodAccessorImpl.invoke0(NativeMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:585) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:29) at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.java:127) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.java:127) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServletSauthorizedInvoker.un(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServletServletServletServlet.java:92) at java.servlet.http.HttpServlet.service(HttpServletServletServletSctin.run(StubSecurityHelper.java:223) at weblogic.servet.internal.StubSecurityHelperServletServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(Au	<pre><tautcode>soapenv:server</tautcode> fautcode> fautc</pre>
required. <details <</br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></details 	<taultstrings-railed component-railed="" end="" invoke="" is<="" methodparameter="" p_cpi="" td="" to=""></taultstrings-railed>
<pre>cdetail> cdetail> cdetail></pre>	required.
<pre>coea_taut:*tacktrace xmns:bea_taut:*http://www.bea.com/servers/wls70/webservice/fault/1.0.0*>java.lang.IllegalArgumentException: Parameter p_cpf is required. at br.uniriotec.services.ClienteService.yetCliente(ClienteService.java:51) at br.uniriotec.services.ClienteService.getCliente(ClienteService.java:21) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.Method.invoke(Method.java:S85) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServletSauthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.servet.internal.StubSecurityHelper/ServletServletGaseWSServlet.java:223) at weblogic.servlet.internal.StubSecurityHelper/ServletServletStubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblog</pre>	<detail></detail>
<pre>Xmins:bea_Tault="ntp://www.bea.com/servers/wis/0/webservice/tault/1.0.0">java.lang.illegalArgumentException: Parameter p_cpf is required. at br.uniriotec.services.ClienteService.getCliente(ClienteService.java:21) at sun.reflect.NativeMethodAccessorImpl.invoke(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:S85) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.mandler.Handlertterator.handleRequest(Handlertherdor.java:127) at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServletServletSoapProcessor.java:44) at weblogic.wsee.server.servlet.BaseWSServletServletServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServletServletGaseWSServlet.java:92) at weblogic.servet.internal.StubSecurityHelper\$servletServletServletGatom.run(StubSecurityHelper.java:223) at weblogic.servet.internal.StubSecurityHelper\$servletServletStubImpl.java:175) at weblogic.servet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at we</pre>	<pre> company starktrace</pre>
Parameter p_cpr is required. at br.uniriotec.services.ClienteService.yerificarStringNull(ClienteService.java:51) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke0(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:585) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:127) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.ws.WsSkel.invoke(WsSkel.java:80) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServletSAuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.server.servlet.BaseWSServletSAuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.servet.internal.StubSecurityHelper\$ServletServletServiteAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper\$ServletServletStubImpl.java:125) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.acti.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:321) at weblogic.security.service.Securi	xmins:bea_tault="http://www.bea.com/servers/wis/0/webservice/fault/1.0.0">java.lang.lilegalArgumentException
at br.uninotec.services.ClienteService.yetCliente(ClienteService.java:51) at br.uninotec.services.ClienteService.getCliente(ClienteService.java:21) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.Nethod.invoke(Method.java:585) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.services.ClienteService.java:585) at weblogic.wsee.services.orgonent.thandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.service.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ComponentHandler.java:127) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:66) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:44) at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServlet.java:85) at weblogic.servet.internal.StubSecurityHelper/ServletServletCation.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.invokeServletSubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.StubSecurityHelper.servletServletSubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.ja	Parameter p_cpt is required.
at br.uninotec.services.clienteservice.getClienteservice.java:21) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethod) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.NativeMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.sus.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.handler.HandlerIterator.handleRequest(HandlerIterator.java:127) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.ws.WsSkel.invoke(WsSkel.java:80) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServletServletGaseWSServlet.java:173) at weblogic.server.servlet.BaseWSServletServletGaseWSServlet.java:173) at weblogic.servet.internal.StubSecurityHelper;invokeServlet.java:850 at weblogic.servet.internal.StubSecurityHelper;invokeServletStubImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;ServletServletSubImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;ServletSubImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;invokeServletStubImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;invokeServletImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;invokeServletStubImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;invokeServletImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;invokeServletImpl.java:223) at weblogic.servet.internal.ServletStubImpl.execute(ServletStubImpl.java:125) at weblogic.servet.internal.ServletStubImpl.execute(ServletStubImpl.java:125) at weblogic.servet.internal.ServletStubImpl.execute(ServletImpl.java:125) at weblogic.servet.internal.AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:21) at weblogic.security.service.SecurityManager.runAs(Se	at br.uniriotec.services.clienteService.vermcarstringNull(ClienteService.Java:51)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:585) at weblogic.wee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BoseWSServletSAuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.seve.server.servlet.BaseWSServletServletServletJava:92) at javax.servlet.http.HttpServlet.service(HtpServletServletGatton.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelperServletServletStubImpl.java:175) at weblogic.servlet.internal.StubSecurityHelperServletServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.acti.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityMelaper.runAs(SecurityMelaper.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:321) at weblogic.security.serv	at br.uninotec.services.ClienteService.getCliente(ClienteService.java:21)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:585) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.invoke(Method.java:585) at weblogic.wsee.invoke(JavaClassComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.ws.Wskel.invoke(Wsskel.java:80) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:46) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:44) at weblogic.wsee.server.servlet.BaseWSServletAuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.see.server.servlet.BaseWSServlet.service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServletServletSecution.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper/ServletServletStubImpl.java:125) at weblogic.servlet.internal.StubSecurityHelper/ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocation.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.ServletStubImpl.execure(ServletManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.Secur	at sun.reflect.NativeMethodAccessorImpl.invokeU(Native Method)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:585) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.handler.HandlerIterator.handleRequest(ComponentHandler.java:64) at weblogic.wsee.handler.HandlerIterator.handleRequest(HandlerIterator.java:127) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServletSetNet.service(BaseWSServlet.java:173) at weblogic.server.servlet.BaseWSServletServletSetServlet.java:173) at weblogic.servet.internal.StubSecurityHelper;invokeServlet.java:850 at weblogic.servet.internal.StubSecurityHelper;invokeServletStubImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;invokeServletStubImpl.java:125) at weblogic.servet.internal.StubSecurityHelper;invokeServletInvocation.run(StubSecurityHelper.java:223) at weblogic.servet.internal.StubSecurityHelper.invoksServletInvocation.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.invoksServletInvocation.run (WebAppServletContext.java:3245) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at java.lang.reflect.Method.invoke(Method.java:S85) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99) at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.ws.dispatch.server.serverDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:64) at weblogic.wsee.server.servlet.BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.servet.internal.StubSecurityHelper\$ServletServletAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper\$ServletServletStubEmpl.java:175) at weblogic.servlet.internal.ServletStubEmpl.execute(ServletStubEmpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubEmpl.execute(ServletEmvocationAction.run (WebAppServletContext.java:3245) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:221) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321)	at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at weblogic.wsee.ws.dispatch.server.Component.invoke(JavaClassComponent.java:99) at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:127) at weblogic.wsee.ws.WsSkel.invoke(WsSkel.java:80) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:44) at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServlet.java:86) at weblogic.servlet.internal.StubSecurityHelper/ServletServiceXction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper/ServletServletStubImpl.java:125) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocation.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.servlet.ServletSubImpl.execute(ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.servlet.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.securityManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.securityManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.securityManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.securityManager.runAs(SecurityManager	at java.lang.reflect.Method.invoke(Method.java:585)
at weblogic.wsee.handler.Handlertterator.handleRequest(ComponentHandler.java:64) at weblogic.wsee.handler.Handlertterator.handleRequest(Handlertteror.java:127) at weblogic.wsee.handler.Handlertterator.handleRequest(Handlertteror.java:127) at weblogic.wsee.ws.WsSkel.invoke(WsSkel.java:80) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServletSAuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServletServletSauthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.servet.http.HttpServlet.service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServletServletSatton.run(StubSecurityHelper.java:223) at weblogic.servet.internal.StubSecurityHelper\$servletServletStubImpl.java:125) at weblogic.servlet.internal.StubSecurityHelper.invokeServlet[StubSecurityHelper.java:125) at weblogic.servlet.internal.StubSecurityHelper.invokeServletInvocation.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at weblogic.wsee.component.pojo.JavaClassComponent.invoke(JavaClassComponent.java:99)
at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:127) at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.servet.internal.StubSecurityHelper\$ServletServletAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper\$ServletServlet(StubSecurityHelper.java:125) at weblogic.servlet.internal.StubSecurityHelper\$ServletSubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticatedSubject.java:321) at weblogic.servlet.internal.WebAppServletContext serverdEvertexter	at weblogic.wsee.ws.dispatch.server.ComponentHandler.handleRequest(ComponentHandler.java:64)
at weblogic.wsee.ws.WsSkel.invoke(WsSkel.java:80) at weblogic.wsee.ws.WsSkel.invoke(WsSkel.java:80) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:46) at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:44) at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServlet.java:856) at weblogic.servlet.internal.StubSecurityHelper/ServletServletServletGaseWSServlet.java:223) at weblogic.servlet.internal.StubSecurityHelper/invokeServlet(StubSecurityHelper.java:223) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:283) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$servletInvocation.run (WebAppServletContext.java:3245) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.servlet.servletManager.runAs(SecurityManager.java:121) at weblogic.security.servlet.severnex.servletSubImpl.ava:2002) at weblogic.security.servlet.severnex.seve	at weblogic.wsee.handler.HandlerIterator.handleRequest(HandlerIterator.java:127)
at weblogic. wsee. WsSkel.invoke(WsSkel.java:80) at weblogic. wsee. server. servlet. SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic. wsee. server. servlet. BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic. wsee. server. servlet. BaseWSServlet. service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServlet.GaveWSServlet.java:92) at weblogic. servet.internal.StubSecurityHelper\$ServletServletAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:125) at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:125) at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocation.run (WebAppServletContext.java:3245) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.secuty.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at weblogic.wsee.ws.dispatch.server.ServerDispatcher.dispatch(ServerDispatcher.java:85)
at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66) at weblogic.wsee.server.servlet.BoapProcessor.process(SoapProcessor.java:44) at weblogic.wsee.server.servlet.BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServlet\$Servlet\$Servlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServletServiceAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper\$ServletService(StubSecurityHelper.java:125) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:283) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:320) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at weblogic.wsee.ws.WsSkel.invoke(WsSkel.java:80)
at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:44) at weblogic.wsee.server.servlet.BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServlet.java:856) at weblogic.servlet.internal.StubSecurityHelper\$ServletServletServiceAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:125) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticateGSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticateGSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.WebAppServletStubImpServletSubImpl.java:321) at weblogic.servlet.internal.AuthenticateGSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.AuthenticateGSubject.doAs(AuthenticatedSubject.java:321) at weblogic.servlet.internal.WebAppServletStubImpl.execute(MyAhanServletSubImpl.java:3002) Serverstational WebAppServletStubImpl.execute(MyAhanServletSubImpl.java:3002)	at weblogic.wsee.server.servlet.SoapProcessor.handlePost(SoapProcessor.java:66)
at weblogic.wsee.server.servlet.BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173) at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServlet.java:856) at weblogic.servlet.internal.StubSecurityHelper\$ServletServletAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.invoksServlet(StubSecurityHelper.java:125) at weblogic.servlet.internal.StubSecurityHelper.invoksServlet(StubSecurityHelper.java:223) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.servletSubAppServletContext securidWabAppServletContext internal.	at weblogic.wsee.server.servlet.SoapProcessor.process(SoapProcessor.java:44)
at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:92) at javax.servlet.http.HttpServlet.service(HttpServlet.java:856) at weblogic.servlet.internal.StubSecurityHelper\$ServletServiceAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:125) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AservletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AservletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AservletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.AservletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.servlet.internal.WebAppServletContext securedEventedWebAppServletContext izus:3002) Secure 20 Secure secure and	at weblogic.wsee.server.servlet.BaseWSServlet\$AuthorizedInvoke.run(BaseWSServlet.java:173)
at weblogic.servlet.internal.StubSecurityHelpersiava:856) at weblogic.servlet.internal.StubSecurityHelpersiava:223) at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:223) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:283) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:275) at weblogic.servlet.internal.WebAppServletContext\$ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.servlet.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at weblogic.wsee.server.servlet.BaseWSServlet.service(BaseWSServlet.java:92)
at weblogic.servlet.internal.StubSecurityHelper\$ServletServletAction.run(StubSecurityHelper.java:223) at weblogic.servlet.internal.StubSecurityHelper.invoksServlet(StubSecurityHelper.java:125) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.act.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at javax.servlet.http.HttpServlet.service(HttpServlet.java:856)
at weblogic.servlet.internal.StubSecurityHeiper.invokeServlet(StubSecurityHeiper.java:125) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(Securits/WebAppServletContext java:3002)	at weblogic.servlet.internal.StubSecurityHelper\$ServletServiceAction.run(StubSecurityHelper.java:223)
at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:283) at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContextSservletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:125)
at weblogic.servlet.internal.WebAppServletStubImpl.execute(ServletStubImpl.java:175) at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:283)
at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run (WebAppServletContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)	at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:175)
(WebAppServietContext.java:3245) at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic security internal WebAppServiceContext securedEvector(WebAppServiceTontext isus:3002) at weblogic security internal WebAppServiceContext securedEvector(WebAppServiceTontext isus:3002)	at weblogic.servlet.internal.WebAppServletContext\$ServletInvocationAction.run
at weblogic.security.act.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321) at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic security.actionater and the security and the se	(WebAppServletContext.java:3245)
at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121) at weblogic soulds internal WebAnnEorgiatContext securedEverte(WebAnnEorgiatContext iaua:3002)	at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321)
at wonlogic conjust internal WohlansConjust Context conjugative statistics (2002)	at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)
	st walladie caniat intensi walangeaniat antony caninate walta (Walangeaniat attaits)

3.2 Realização de testes utilizando o SOAPUI

Outra forma de realizar o teste do serviço é utilizando a ferramenta SOAPUI⁶. O SOA-PUI é uma ferramenta free amplamente utilizada para testes de web services. A seguir, são descritos os passos necessários para testar o serviço ClienteService usando o SOA-PUI.

- 1. Primeiro é necessário gerar o arquivo WSDL do WebService implementado.
 - a. Clicar com o botão direito no WebService ClienteService → WebServices → Generate WSDL (Figura 33).

⁶ http://www.soapui.org/

<mark>[]</mark> ▼ 🛛 🕘] 🎄 ▼ 🕻) • Q •] 😃 🕸 @ •	3 1			- 🖓 - 🍫 🗘
Package Explorer		Clie	nteDbControl.	🐼 Clien	teService.ja
UniRioTec Src bruniriotec.b Clientejav Clientejav Cliented bruniriotec.c D		F3 F4 Ctrl+C Ctrl+V Alt+Shift+S)	<pre>* iguais, * Fortant mesmo * * @param * @throws */ public sta if (pojc throw } if (xml throw } // Refer Class pc Class xm</pre>	só variand o, é import iome e tipo. pojo pojo r xml XMLBean Exception tic void pa == null) { new Illegal ëncias feit joClass = pm llClass = xm
commons-	Export		2 Source \	fiew S Javadoc	Tasks Search
er Design View Palette 2	Declarations		• WebLogic	v9.2 Server [Doi	Status Debugging
3	Run As Debug As Upgrade Source File(s) Team Compare With Replace With Restore from Local History		> > > >		
	Web Services		Gen	erate WSDL	
brunirioter se	Properties	Alt+Enter			

Figura 33 – Geração do arquivo WSDL do WebSevice implementado

- 2. Executar SOAPUI.
- 3. Criar projeto a partir do WSDL do serviço (Figura 34)

El soaporz			
<u>File</u> <u>T</u> ool:	s <u>D</u> esktop	<u>H</u> elp	
1 1	× 🐚		
III Proje	ects	New WSDL Project New WSDL Project Creates a new WSDL Project in this workspace Project Name UnidadeOperativaTst Initial WSDL (com\petrobras\services\UnidadeOperativaServiceService.wsd) Create Requests I Create sample requests for all operations? Create Project File Creates a file for the project (can always be created later)	al Browse
			OK Cancel

Figura 34 – Criação do projeto a partir do WSDL do serviço

4. Abrir o Request Editor correspondente ao método que se deseja testar (Figura 35).

soapUI 2.0.2				
e <u>T</u> ools <u>D</u> esktop <u>H</u> e	elp			
s 🖹 🐇 😼				
Projects □ ClienteServiceTest □ I ClienteService □ Z cliente □ \$ Reque: □ \$ Reque: □	e ServiceSoapBinding t 1 Show Request Editor	Enter		
	'≡ Ad Shows the Reque ⁸ / ₈ Add to MockService	st Editor for thi	s Request	
	Clone Request	P9		
	Rename Delete	F2 Delete		
	Online Help	F1		
Dequest Pronetier				
Request Properties				
Name F Description Message Size 2	Value Request 1	_		
Encoding	JIF-8	cooplil log	http://og	inte

Figura 35 – Abrindo do request editor

5. Preencher o parâmetro do método, por exemplo, substituir "?" por "0123450000" (Figura 36).



Figura 36 – Ajuste do parâmetro cpf do método do serviço

6. Clicar no botão de execução (Figura 37).

🛓 soapUI 2.0.2	
<u>F</u> ile <u>T</u> ools <u>D</u> esktop <u>H</u> elp	
🖻 🗟 🔓 💥 🌬	
E Projects ⊕ ClienteServiceTeste ⊕ I ClienteServiceSoapBinding ⊕ ♀ getCliente ↓ I Request 1	<pre>Request 1</pre>

Figura 37 – Botão para execução do serviço

7. O XML de resposta deve ser semelhante ao apresentado a seguir (Figura 38).



Figura 38 – XML de resposta

3.3 Realização de testes implementando um cliente Java para o serviço

O serviço também pode ser testado através da implementação de um cliente Java para o mesmo, como descrito no passo-a-passo a seguir.

1. Criar projeto Java (Figura 39).

🚾 New Project
Select a wizard
Create a Java project
Wizards:
E CVS
EJB
in in the second secon
🚊 🗁 Java
Java Project
🗄 🗁 Simple
🗄 🔁 Web
🗄 🗁 WebLogic Integration
🗄 🗁 WebLogic Portal
🗄 🗁 Web Services
Examples

Figura 39 – Criação do projeto Java

- 2. O próximo passo corresponde à geração do Stub para o webservice utilizando uma task do Ant. Para isto é necessário incluir referência no projeto para weblogic.jar, webserviceclient.jar e para axis.jar.
 - a. Para adicionar referência para weblogic.jar, configurar o buildpath, estendendo a variável WEBLOGIC_HOME, de acordo com as figuras a seguir (Figura 40, Figura 41, Figura 42, Figura 43).

type filter text 👻	Java Build Path	
Info BeanInfo Path Builders	Source Brojects Survey Content Sector Sec	nd Export
File Templates Iava Build Path		Add JARs
⊕ Java Code Style		Add External JARs
 Java Compiler Javadoc Location 		Add <u>V</u> ariable
 JSP Design Palette Profile Compliance anc Project References Server Task Tags Validation WebLogic Portal Server XMLBeans 		Add Li <u>b</u> rary
		Add Class <u>F</u> older
		<u>E</u> dit
		<u>R</u> emove
	Default output folder:	Browce
e m b		
		OK Cancel

Figura 40 – Janela Libraries de Java Build Path para extensão da variável WEBLOGIC_HOME

w Variable Classpath Entry	- • •
Select <u>v</u> ariables to add to build path:	
ECLIPSE_HOME - C:\bea\workshop92\eclipse JRE_LIB - C:\bea\jdk150_06\jre\lib\rt.jar JRE_SRC - (empty)	E <u>x</u> tend
JRE_SRCROOT - (empty)	
WEBLOGIC_HOME - C:\bea\weblogic92	
۰	
Configur <u>e</u> Variables	
Variable points to a folder: To choose ahive inside the fold	er, click 'extend'.
ОК	Cancel

Figura 41 – Seleção do WEBLOGIC_HOME para extensão de variáveis

Wariable Extension	
Choose extensions to 'WEBLOGIC_HOME'.	
 javelin javelin platform portal samples server bin consoleann 	
in in interpreter	Ŧ
OK	Cancel

Figura 42 – Janela de escolha de variáveis para extensão WEBLOGIC_HOME (server\lib)

W Variable Extension	- • •
Choose extensions to 'WEBLOGIC_HOME'.	
Image: sql-8.3-604.jdbc3.jar Image: sql-8.3-604.jdbc4.jar Image: sql-8.3-604.jdbc4.jar <td< td=""><td>-</td></td<>	-
ОК	Cancel

Figura 43 – Seleção da variável de extensão weblogic.jar

- b. Realizar as mesmas etapas para o arquivo webserviceclient.jar.
- c. Para adicionar referência para axis.jar, estender a variável ECLIP-SE_HOME (Figura 44, Figura 45, Figura 46).

	Java bullu Palli	(P * G)	
Info BeanInfo Path Builders	@ Source @ Projects ▲ Libraries	Ì	
- File Templates	WEBLOGIC_HOME/server/lib/weblogic.jar - C:\bea\v	Add JARs	
Java Code Style		Add E <u>x</u> ternal JARs Add <u>V</u> ariable Add Li <u>b</u> rary	
Java Compiler Javadoc Location			
JSP Design Palette			
– Project References – Server – Task Tags – Validation – WebLogic Portal Server – XMLBeans		Add Class <u>F</u> older	
		<u>E</u> dit,	
		<u>R</u> emove	
	< ₩ →		
	Defaul <u>t</u> output folder:		
	WSCliente1	Bro <u>w</u> se	

Figura 44 – Janela de extensão de arquivos para a variável ECLIPSE_HOME

www.Variable Classpath Entry	- • •
Select <u>v</u> ariables to add to build path:	
 ECLIPSE_HOME - C:\bea\workshop92\eclipse JRE_LIB - C:\bea\jdk150_06\jre\lib\rt.jar JRE_SRC - (empty) JRE_SRCROOT - (empty) JUNIT_HOME - C:\bea\workshop92\eclipse\plugins\org. WEBLOGIC_HOME - C:\bea\weblogic92 	E <u>x</u> tend
< <u> </u>	
Configur <u>e</u> Variables	
Variable points to a folder: To choose ahive inside the folder	er, click 'extend'.
ОК	Cancel

Figura 45 - Seleção do ECLIPSE_HOME para extensão de variáveis



Figura 46 - Seleção da variável de extensão axis.jar

3. Criar arquivo com o nome "build.xml" (Figura 47).

W New File		X
File		
Create a new file resource.		
Enter or select the parent folder:		
UniRioTecTeste		
🕀 😥 UniRioTec		
UniRioTecTeste		
File na <u>m</u> e: Build.xml		
<u>A</u> dvanced >>		
		1
	<u> </u>	Cancel

Figura 47 – Criação de arquivo para o código da task Ant.

4. Criar task Ant, segundo o código a seguir (Figura 48), colar o código no arquivo "build.xml" criado no passo anterior.



Figura 48 – Ant task para criação do cliente Java

Observe que foi necessário incluir no classpath o caminho para o arquivo weblogic.jar: "classpath="c:/bea/weblogic92/server/lib/weblogic.jar""

5. Executar o arquivo (Run As → Ant Build) (Figura 49).

6. Após a execução, atualize o projeto (pressionar F5).



Figura 49 – Execução do arquivo build.xml

7. O resultado de execução será algo semelhante ao apresentado na Figura 50. A classe Cliente_Stub é criada. Esta classe é responsável por realizar o encapsulamento da chamada do web service. Desta forma, o cliente invoca o método da classe stub, que por sua vez invoca o método do serviço. O serviço retorna a resposta para o stub que repassa para o cliente (Figura 51).



Figura 50 - Resultado da execução do build.xml



8. Implementar classe cliente Java de acordo com o código a seguir (Figura 52):



Figura 52 – Criação de classe Java

Source fol <u>d</u> er: UniRioTecTeste Package: src.br.uniriotec.testecliente Enclosing type:	Browse Browse Browse
Package: src.br.uniriotec.testecliente	Bro <u>w</u> se Bro <u>w</u> se
Enclosing type:	Browse
Na <u>m</u> e: WSClient	
Modifiers:	ted
Superclass: java.lang.Object	Brows <u>e</u>
Interfaces:	<u>A</u> dd
	Remove
Which method stubs would you like to create?	
public static void main(String[] args) Constructors from superclass	
✓ Inherited abstract methods	
Do you want to add comments as configured in the properties of the curr	ent project?

Figura 53 – Especificação da classe Java WSClient



Figura 54 – Código da classe WSClient

7. Executar o cliente Java. (Run As → Java Application) (Figura 55) Caso ocorra algum erro, verifique no console do servidor se a botão "Release Configuration" está habilitado, conforme a Figura 29. Se estiver, clique nele para liberar a configuração e execute novamente o servidor.

Package Explorer ⊗ Hierarchy	Java - WSClient.java - BE Eile Edit Source Refac	A Workshop for WebLogic Pla tor <u>N</u> avigate Se <u>a</u> rch <u>P</u> roj) + Q, + 년 冊 영 •	itform ect <u>R</u> un <u>W</u> ind · 🎒 🄗 🌙	ow <u>H</u> elj) 월 순] + 등] +	*500+0-		
BDPTServices BUEPServices e p-utils-1.0 UnidadeOperativa Image: ClientEspace Image: ClientEspace <th>Package Explorer 🙁</th> <th>Hierarchy 🛛 🗘 🧔</th> <th></th> <th></th> <th>lienteService.java</th> <th>Test Client</th> <th>4</th>	Package Explorer 🙁	Hierarchy 🛛 🗘 🧔			lienteService.java	Test Client	4	
Refresh for (int i = 0; Cliente_cli Run As I Run on Server Alt-Shift+X, R Debug As Upgrade Source File(s) Team Compare With Replace With Javadoc Declaration E Console S	BDPTServices BIEPServices ep-utils-1.0 Integrado UniRioTec UniRioTecTeste Deans.controls.c Dietentejava Diete	liente New Open With Open Type Hierarchy Copy Paste Zolete Build Path Source Refactor Refactor Secont References Declarations	F3 F4 Ctrl+C Ctrl+V Alt+Shift+S Alt+Shift+T	> - > - > - > -	package src.b import java.u import beans. import ber.uni import br.uni import org.ag public class public st /** * @param */ public st try[j j j j j j j j j j j j j	controls.cli controls.cli controls.cli criotec.teste riotec.teste ache.axis.cli WSClient { ache axis.cli wSClient { atic Clientes atic Clientes atic Clientes atic void ma: g wsdlurl = costor = new //locator.set system.out.pr stCliente 1st	serv in (S clii ient Serv in (S clii Main loca intl	
Run As I Run on Server Alt+Shift+X, R Debug As I 2 Java Application Alt+Shift+X, J Upgrade Source File(s) II 3 SWT Application Alt+Shift+X, S Team IIII 3 SWT Application Alt+Shift+X, S Compare With IIII Automation Alt+Shift+X, S Replace With IIIII Javadoc Declaration	4	🧬 Refresh			f	for (int i =)); i lien	
Debug As Upgrade Source File(s) Team Compare With Replace With Interface With		Run As		1 R	un on Server	Alt+Shift+X, R	.pr	
Upgrade Source File(s) Team Compare With Replace With IJavadoc Declaration E Console S See		Debug As		۲ ت 2 J	ava Application	Alt+Shift+X, J	.pr	
Team Compare With Replace With	Upgrade Source File(s)			DWT 3 S	3 SWT Application Alt+Shift+X, S			
		Team Compare With Replace With		D Run				
Restore from Local History BEA WebLogic v9.2 Server [Dominio Servicos]	< [Restore from Local Histo	ry	BEA	WebLogic v9.2 Serve	er IDominio Servico	s1 (BE	

Figura 55 – Execução do cliente java como aplicação java

4 Conclusão

O presente relatório teve como objetivo ressaltar os principais aspectos do desenvolvimento de serviços, em uma metodologia BOTTOM-UP. Dessa forma, o serviço é gerado a partir de uma consulta na base de dados, a qual foi solicitada por uma demanda de acesso a dados.

Foram apresentados detalhes da implementação de serviços utilizando as ferramentas da BEA para acessar um banco de dados armazenado no PostgreSQL. Um classe POJO foi definida para armazenar o resultado da consulta e rotinas de transformação são responsáveis por transformar objetos POJO em estrutura XML de acordo com arquivo XSD definido qual é retornada para o consumidor do serviço.

5 Referências

ERL, T., 2005, Service-Oriented Architecture: concepts, technology, and Design, Prentice Hall.

PAPAZOGLOU, MIKE P.; HEUVEL, WILLEM-JAN, 2007, Service oriented architectures: approaches, technologies and research issues, VLDB Journal, Springer-Verlag.