

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
ESCOLA DE INFORMÁTICA APLICADA  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Ferramenta para Verificação de Conformidades utilizando Ontologias e Regras  
Semânticas em Governança SOA

Nome do autor:

Daniel Karam Venceslau

Nomes dos Orientadores:

Leonardo Guerreiro Azevedo

Haroldo Maria Teixeira Filho

Dezembro/2013

Ferramenta para Verificação de Conformidades utilizando Ontologias e Regras  
Semânticas em Governança SOA

Projeto de Graduação apresentado à  
Escola de Informática Aplicada da  
Universidade Federal do Estado do Rio  
de Janeiro (UNIRIO) para obtenção do  
título de Bacharel em Sistemas de  
Informação.

Nome do Autor:

Daniel Karam Venceslau

Nomes dos Orientadores:

Leonardo Guerreiro Azevedo

Haroldo Maria Teixeira Filho

Ferramenta para Verificação de Conformidades utilizando Ontologias e Regras  
Semânticas em Governança SOA

Aprovado em \_\_\_\_/\_\_\_\_\_/\_\_\_\_

BANCA EXAMINADORA

---

Prof. Leonardo Guerreiro Azevedo, D.Sc. (IBM Research – Brasil; PPGI-UNIRIO)

---

Haroldo Maria Teixeira Filho (PPGI-UNIRIO)

---

Prof. Sean Wolfgang Matsui Siqueira, D.Sc. (UNIRIO)

---

Prof<sup>a</sup>. Kate Cerqueira Revoredo, D.Sc. (UNIRIO)

O autor deste Projeto autoriza a ESCOLA DE INFORMÁTICA APLICADA da UNIRIO a divulgá-lo, no todo ou em parte, resguardando os direitos autorais conforme legislação vigente.

Rio de Janeiro, \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

---

Daniel Karam Venceslau

## **Agradecimentos**

Aos meus pais, que com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Ao professor e orientador Leonardo Azevedo e Haroldo Teixeira Filho pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia.

A todo o corpo docente da UNIRIO que foram tão importantes na minha vida acadêmica e no desenvolvimento desta monografia.

Aos amigos e colegas, pelo incentivo e pelo apoio constantes que foram essenciais nesses anos de graduação, auxiliando e apoiando.

E, finalmente agradeço a todos aqueles que de alguma forma estiveram próximos, me incentivando sempre a conquistar os meus sonhos.

# Sumário

Capítulo 1: Introdução.....	10
1.1 Motivação .....	10
1.2 Objetivo .....	12
1.3 Contribuições do trabalho .....	12
1.4 Estrutura do texto .....	12
Capítulo 2: Principais Conceitos.....	13
2.1 SOA .....	13
2.1.1 Definição .....	13
2.1.2 Serviços .....	14
2.1.3 Governança SOA.....	15
2.2 Ontologia .....	15
2.3 URI .....	16
2.4 Namespace.....	17
2.5 RDF .....	18
2.6 OWL.....	20
2.7 SWRL.....	22
Capítulo 3: Proposta para modelagem da ferramenta.....	23
3.1 Método centrado em ontologias para governança de Arquitetura orientada a Serviços .....	23
3.2 Ontologia utilizada .....	24
3.3 Casos de uso .....	26
Capítulo 4: Ferramenta para Governança SOA .....	27
4.1 Planejamento e desenvolvimento da ferramenta .....	27
4.2 Página inicial ( <i>Home</i> ).....	28
4.3 Manter políticas .....	29
4.4 Obter lista de não conformidades .....	35
4.5 Manter justificativas .....	37
4.6 Manter ações.....	38
Capítulo 5: Avaliação da Ferramenta .....	42
5.1 Avaliação utilizada .....	42

5.2 Análise dos resultados .....	43
Capítulo 6: Conclusão .....	44
Referências .....	46
Apêndice I – Casos de uso .....	48
Apêndice II – Unificando URIs .....	54
Apêndice III – Preparando o ambiente .....	55
Apêndice IV – Diagrama de estados de uma ação.....	65
Apêndice V – Roteiro de testes para avaliação.....	66
Apêndice VI – Questionários de avaliação.....	68
Apêndice VII – Resposta dos participantes .....	70
Anexo I – Documento de visão.....	73

## LISTA DE FIGURAS

Figura 1 – Exemplo de <i>namespace</i> .....	17
Figura 2 – Exemplo de arquivo RDF .....	19
Figura 3 - Exemplo de arquivo OWL .....	21
Figura 4 - Arquitetura proposta por Teixeira Filho [2013a, 2013b] .....	24
Figura 5 - Ontologia para SOA (adaptada de (The Open Group, 2010)).....	25
Figura 6 – Diagrama de casos de uso.....	26
Figura 7 - Página inicial da aplicação .....	29
Figura 8 - Ontologia disponível na página "Manter políticas".....	30
Figura 9 - Exemplo de inclusão de uma nova política (parte 1) .....	31
Figura 10 - Exemplo de inclusão de uma nova política (parte 2) .....	32
Figura 11 - Preenchendo caixa de seleção "Valores da propriedade selecionada" (parte 1)..	33
Figura 12 - Preenchendo caixa de seleção "Valores da propriedade selecionada" (parte 2)..	34
Figura 13 - Lista de não conformidades (parte 1).....	35
Figura 14 - Lista de não conformidades (parte 2).....	36
Figura 15 - Exemplo para incluir uma justificativa .....	37
Figura 16 - Tabela de justificativas.....	38
Figura 17 - Exemplo para incluir uma ação.....	39
Figura 18 - Tabelas de ações.....	40
Figura 19 - Unificar diversas URI.....	54
Figura 20 - Exemplo de caminho onde JDK será instalado .....	56
Figura 21 - Especificação do caminho de instalação da JRE do JDK.....	57
Figura 22 - Configuração da JRE no eclipse.....	58
Figura 23 - Marcar check-box da nova JRE configurada .....	59
Figura 24 - Exemplo de caminho do projeto "Gestor_Ontologia" .....	60
Figura 25 - Caminho default da JRE selecionada .....	61
Figura 26 - Adicionando servidor web (parte 1).....	62
Figura 27 - Adicionando servidor web (parte 2).....	63
Figura 28 - Adicionando servidor web (parte 3).....	64

## LISTA DE ABREVIATURAS

FTP - *File Transfer Protocol*

HTTP - *Hypertext Transfer Protocol*

IA - Inteligência artificial

OWL - *Ontology Web Language*

RDF - *Resource Description Framework*

SOA – *Service Oriented Architecture*

SWRL - *Semantic Web Rule Language*

TI – Tecnologia da Informação

URI - *Uniform resource identifier*

URN - *Uniform Resource Name*

URL - *Uniform Resource Locator*

XML – *Extensible Markup Language*

## RESUMO

Devido à constante necessidade que as corporações possuem para o aumento da eficiência de seus serviços e ao mesmo tempo redução de custos, surge a necessidade de serem realizadas novas estratégias, e a área de TI (Tecnologia da Informação) exerce papel fundamental em uma empresa para ajudá-la a atingir esses objetivos. Uma estratégia que vem sendo adotada pela TI para ajudar uma organização a atingir os seus objetivos arquiteturais é o uso de uma Arquitetura Orientada a Serviços (SOA - Service-Oriented Architecture). Entretanto, a maior dificuldade de se utilizar esse tipo de arquitetura, está na utilização de uma governança SOA que seja alinhada com a arquitetura para dar suporte aos negócios e, ao mesmo tempo, agilize as atividades e processos das corporações. Esse trabalho apresenta uma ferramenta para governança SOA para verificação de conformidades de regras de governança utilizando ontologias e regras semânticas, tornando a atividade de identificação de não conformidades muito mais ágil.

**Palavras-chave:** SOA, Governança SOA, Ontologias, SWRL, OWL.

# Capítulo 1: Introdução

O objetivo deste capítulo é contextualizar o assunto tratado neste trabalho, assim como apresentar a sua motivação, o seu objetivo e a estrutura de capítulos.

## 1.1 Motivação

Com o atual mercado, constantemente mais exigente com custos e cronogramas, as empresas precisam adotar novos métodos para atendê-lo de forma rápida e eficaz. Sendo assim, a necessidade por softwares que ajudem na tomada de decisões e agilizem o tráfego de informações aumenta de forma significativa, demandando uma fabricação de software ágil e eficaz [Josuttis, 2007]. SOA (Arquitetura orientada a Serviço) é uma estratégia bastante visada atualmente pelas corporações para alcançar esse objetivo.

SOA possui diversas definições. Josuttis (2007) define-a como sendo uma estratégia para a realização e a manutenção dos processos corporativos que se encontram em grandes sistemas distribuídos. Papazoglou (2007) define SOA como sendo um paradigma computacional que flexibiliza o desenvolvimento e reorganiza um conjunto de aplicações independentes em diversos serviços interligados que podem interagir para executar determinadas tarefas.

SOA visa agilizar a construção de novas aplicações, trazendo benefícios como: a redução do tempo total do projeto e prazos de entregas, menores riscos no decorrer do desenvolvimento e redução de custos [Erl, 2005; Papazoglou, 2007]. Entretanto, existem dificuldades para se beneficiar desses ganhos quando SOA é implementado em grandes corporações [The Open Group, 2009].

Alguns desafios citados por Niemann *et al.* (2010) e Schepers *et al.* (2008) incluem: garantir gerenciamento de responsabilidades e papéis, flexibilizar gerenciamento de ambientes com diversos stakeholders, viabilizar a criação de uma política de compartilhamento e reuso, e garantir conformidade com políticas internas.

Estes desafios podem ser tratados através de uma governança SOA, que define o estabelecimento de estruturas, políticas, processos de tomada de decisão, papéis e responsabilidades que garantem a implementação de SOA de forma que esteja de acordo com objetivos e estratégias corporativas [Janiesch *et al*, 2009]. Porém, existem diversos modelos para governança SOA que divergem em diferentes aspectos e formas.

Janiesch *et al.* (2009) e Niemann *et al.* (2010) apresentam propostas dos elementos necessários para o estabelecimento da governança SOA. Porém, estes tratam apenas da estrutura de elementos, sem detalhar como instanciá-los em organizações. Enquanto Hojaji e Shirazi (2010) e The Open Group (2009) apresentam propostas de métodos para implantação. Contudo, estas propostas divergem quanto aos processos necessários para governança, principalmente, quanto à necessidade de administrar aplicações consumidoras de serviços e quanto aos itens a serem monitorados em um ambiente SOA, como evidenciado por Teixeira Filho e Azevedo (2012).

Sendo assim, Teixeira Filho e Azevedo (2012) realizaram uma análise destas propostas para governança SOA e elaboraram uma abordagem consolidada chamada CommonGOV, a qual inclui 51 processos para a implementação de uma governança SOA. A proposta considera tanto o contexto técnico, como acordos de níveis de serviço e mecanismos de autenticação [The open group, 2009], quanto questões relativas ao negócio, como a responsabilidade e limite de competência de cada nível da organização [Janiesch *et al*, 2009].

Entretanto, algumas atividades continuam sendo grandes desafios de uma organização, mesmo utilizando uma governança SOA. Um exemplo é a verificação de conformidades, no qual é necessário alocar diversas equipes para realizar essa atividade, exigindo grande tempo e esforço para sua realização. Segundo Niemann *et al.* (2010), a verificação de conformidades é um dos fatores mais críticos para se obter os benefícios de SOA.

Para viabilizar a atividade de verificação de conformidades, são necessários métodos e ferramentas que simplifiquem esta validação. A grande vantagem do uso de ontologias é a possibilidade de interpretação por agentes computacionais, tornando viável o desenvolvimento de ferramentas que agilizem a verificação de conformidades.

## **1.2 Objetivo**

Este trabalho tem como objetivo desenvolver uma ferramenta que utilize ontologias e regras semânticas para a verificação de conformidade para governança de uma arquitetura orientada a serviços em uma organização. Esta ferramenta está baseada na proposta de Teixeira Filho *et al.* (2013a, 2013b) para governança SOA.

## **1.3 Contribuições do trabalho**

Este trabalho tem as seguintes contribuições:

- Apresentação de uma abordagem visando empregar ontologias para desenvolver aplicações capazes de realizar inferências para avaliação de regras semânticas.
- Criação de um ambiente que permita a evolução da aplicação para incluir novos casos de uso na ferramenta para verificação de conformidade em trabalhos futuros.
- Implementação de 4 dos 13 casos de usos identificados para gestão de conformidade no ambiente SOA (Anexo I – Documento de visão).

## **1.4 Estrutura do texto**

Este trabalho está dividido da seguinte forma. O Capítulo 1 corresponde a presente introdução. O Capítulo 2 apresenta definições e conceitos necessários para o entendimento sobre SOA, serviços, governança SOA, ontologias, URI, *namespace*, RDF, OWL e SWRL. O Capítulo 3 apresenta a proposta e a modelagem para o desenvolvimento da ferramenta. O Capítulo 4 apresenta a ferramenta, explicando suas funcionalidades. O Capítulo 5 apresenta a avaliação da ferramenta e, finalmente, o capítulo 6 apresenta a conclusão do trabalho.

## Capítulo 2: Principais Conceitos

Este capítulo tem como objetivo apresentar os principais conceitos e definições da arquitetura orientada a serviços (SOA) e as ferramentas que são utilizadas para apoiar a implementação da arquitetura SOA.

### 2.1 SOA

Esta seção visa explicar conceitos de SOA e destacar seus principais benefícios que são necessários para contextualizar a necessidade do uso da ferramenta desenvolvida neste trabalho.

#### 2.1.1 Definição

SOA é uma arquitetura que visa realizar a integração de diversas aplicações, independentemente das tecnologias utilizadas (plataforma ou linguagem de programação) [Josuttis, 2007]. Tais sistemas devem estar alinhados com a estratégia da tecnologia da informação e com os objetivos de negócio de uma organização [Erl, 2007].

A arquitetura SOA tem como proposta: a obtenção de aplicações integradas por meio de serviço; a descentralização da computação isenta de protocolos; a definição de padrões para o desenvolvimento; e o mapeamento dos sistemas de TI de uma empresa em conjunto com os processos de negócios [Papazoglou *et al.*, 2007].

Um grande enfoque da arquitetura SOA é descentralizar uma grande aplicação, substituindo-a por um conjunto de aplicações mais simples, com módulos especializados e descentralizados. Essas aplicações podem ser acessadas remotamente e, caso necessário, serem reutilizadas em razão da padronização adotada [Erl, 2007].

O resultado, após a implementação, é um grupo de serviços interligados e o alinhamento da tecnologia com o negócio, o que traz como benefício a possibilidade de reutilização de códigos. Conseqüentemente, há um aumento da produtividade e uma redução de custos no desenvolvimento e manutenção dos sistemas envolvidos. [Marks e Bell, 2006].

SOA tem sido apontada como uma grande estratégia para otimização da produtividade, trazendo assim, uma grande vantagem competitiva empresarial. Porém a implementação dessa estratégia não é simples, sendo necessário um estudo cuidadoso dos processos de negócios da organização para definir uma governança SOA que esteja alinhada com as necessidades da empresa [Josuttis, 2007]. Caso não seja implementado corretamente, além de não atingir os benefícios que SOA tem a oferecer, irá se tornar prejudicial a organização.

### **2.1.2 Serviços**

Serviços são módulos de negócios que podem ser considerados como funcionalidades de negócios independentes, que possuem interfaces expostas podendo ser invocadas via mensagens [Josuttis, 2007]. São auto-contidos e são recursos de software bem definidos através de uma linguagem padrão, provendo funcionalidades ao negócio independente do estado de outros serviços [Erl, 2005].

Provavelmente o elemento mais importante da orientação a serviços seja a forma como os serviços são definidos e utilizados por meio de contratos de serviços [Erl, 2007]. Nele são definidos os propósitos, funcionalidades, responsabilidades, governabilidade e entre outros.

Na utilização dos serviços, temos o relacionamento entre três entidades (ou *stakeholders*), segundo Gu e Lago (2007): provedor de serviço, *broker* de serviço e consumidor de serviço. O provedor tem o compromisso de realizar a tarefa baseado nos resultados pré-estabelecidos. O *broker* atua como papel intermediário entre o provedor e o consumidor, informando a localização de um determinado serviço através de um consulta em um registro de serviços. O consumidor utiliza o serviço da forma que foi previamente acordada no contrato de serviço. A utilização do serviço considerando estes três papéis funciona da seguinte forma: o consumidor faz uma pesquisa pelo serviço no *broker*; o *broker* informa a localização deste serviço; o consumidor invoca o serviço enviando a mensagem de requisição para o provedor; o provedor executa o serviço retornando a resposta para o consumidor.

### **2.1.3 Governança SOA**

O The Open Group (2009) aponta que muitas empresas não conseguem usufruir dos benefícios da implementação da arquitetura orientada a serviços para toda organização. Como exemplo, um portfólio de serviços de uma organização pode ter diversas equipes desenvolvendo e utilizando estes serviços sem uma governança estabelecida, e por essa razão, as definições de responsabilidades e a qual processo de negócio os serviços pertencem não ficam bem alinhados.

Para resolver esse problema, Erl (2007) afirma que seja necessário uma estrutura de governança para definir as responsabilidades e processos. Uma governança SOA eficiente requer regras para o uso apropriado dos padrões, tornando claro as expectativas dos interessados e fornecendo garantia na execução de serviços [Kontogiannis, Lewis, Smith, 2008].

## **2.2 Ontologia**

Esta seção contextualiza os conceitos de ontologia que são extremamente importantes para tornar este trabalho e a sua implementação viável.

Na perspectiva computacional, ontologia é um conteúdo teórico sobre diversos objetos, sobre as propriedades destes objetos e os possíveis relacionamentos entre esses objetos em um determinado domínio [Chandrasekaran, Johnson, Benjamins, 1999]. Gruber (1993) define a ontologia como sendo uma especificação explícita e formal de uma contextualização.

Em uma forma simplificada, para a elaboração de uma ontologia, são definidas categorias para os objetos que pertencem a um mesmo domínio. A ontologia funciona como um “catálogo”, categorizando os tipos de objetos que se supõe existir em um domínio determinado [Sowa, 2000].

A terminologia de um determinado domínio da ontologia são os conceitos essenciais deste domínio, sua taxonomia, suas relações com as hierarquias e restrições e os axiomas do domínio. O começo do uso de ontologias na área da computação teve início na década de 90, com forte destaque nas áreas de inteligência artificial [Guarino, 1995].

Na informática, a ontologia tem o papel de servir como um modelo de organização universal de entidades por representação simbólica, ou seja, uma representação formal, possibilitando o raciocínio dedutivo através de regras e inferências. Existe um método conhecido como lógica formal, que possibilita raciocinar sobre as ideias e conceitos, servindo como base para o desenvolvimento de aplicações, principalmente para a área de IA [Sowa, 1987].

## 2.3 URI

Esta seção explica o que são URIs. Seu conceito é importante para explicar a necessidade de prefixos (capítulo 2.4) em arquivos RDF (capítulo 2.5) e arquivos OWL (capítulo 2.6). Para utilizar a ferramenta proposta neste trabalho, é necessário ter conhecimentos de manipulação de URIs e arquivos OWL.

Identificador Uniforme de Recursos (URI)<sup>1</sup>, é uma cadeia de caracteres compacta utilizada para identificar ou denominar um determinado recurso (tais como, imagens, documentos, serviços e outros).

URL (Uniform Resource Locator)<sup>2</sup> é um tipo de URI em que os recursos estão disponíveis na web. Eles tornam esses recursos disponíveis a partir de uma variedade de nomenclatura de esquemas e métodos de acesso (HTTP, FTP e outros).

A sintaxe URI é um nome de conjunto URI como "HTTP", "FTP", "mailto", "URN", "tel", "rtsp", "file" etc, seguido de um caractere dois pontos e, por fim, a parte específica do conjunto. Alguns exemplos de URI são:

- <http://example.org/absolute/URI/with/absolute/path/to/resource.txt>
- <ftp://example.org/resource.txt>
- <urn:issn:1535-3613>

---

<sup>1</sup> <http://www.w3.org/Addressing/>

<sup>2</sup> <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>

## 2.4 Namespace

Um espaço de nomes (*namespace*)<sup>3</sup> é um delimitador abstrato que fornece um contexto para os itens que ele armazena, o que permite uma desambiguação para itens que possuem o mesmo nome, porém residem em espaços de nomes diferentes. Como um contexto distinto é fornecido para cada container, o significado de um nome pode variar de acordo com o espaço de nomes o qual ele pertence.

Para exemplificar, o caminho de um diretório é um *namespace*, pois podemos ter dois arquivos com o mesmo nome, porém com os seus caminhos de diretórios distintos. É o mesmo conceito para a existência de duas pessoas com um mesmo código identificador, por exemplo “123”, porém uma trabalha em uma empresa X e a outra em uma empresa Y. Nesse caso, cada empresa representa um *namespace* distinto para o código de identificação de seus funcionários.

Um outro exemplo para contextualizar melhor a ideia de *namespace* com este trabalho.

```
1<root>
2   <h:table xmlns:h="http://www.w3.org/TR/html4/" >
3     <h:tr>
4       <h:td>Apples</h:td>
5       <h:td>Bananas</h:td>
6     </h:tr>
7   </h:table>
8
9   <f:table xmlns:f="http://www.w3schools.com/furniture">
10    <f:name>African Coffee Table</f:name>
11    <f:width>80</f:width>
12    <f:length>120</f:length>
13  </f:table>
14
15</root>
```

**Figura 1** – Exemplo de *namespace*<sup>4</sup>

<sup>3</sup> <http://searchsoa.techtarget.com/definition/namespace>

<sup>4</sup> [http://www.w3schools.com/xml/xml\\_namespaces.asp](http://www.w3schools.com/xml/xml_namespaces.asp)

Na Figura 1, temos 2 *tables* que se comportam de formas diferentes, sendo assim, é necessário uma *namespace* diferente para cada uma delas. Na linha 2, podemos ver o atributo “xmlns:h=” com o *namespace* “http://www.w3.org/TR/html4/”, indicando que é uma *table* com comportamento padrão de um documento html4. Já a segunda *table* tem o propósito de guardar informações sobre uma mobília, como está indica na sua 9ª linha com um *namespace* “http://www.w3schools.com/furniture”.

## 2.5 RDF

O “Resource Description Framework” (RDF)<sup>5</sup> é o modelo padrão para troca de dados na Web. O RDF possui características que facilitam a junção dos dados mesmo que estejam em esquemas diferentes e tem suporte para a evolução constante dos esquemas sem precisar efetuar mudanças nos consumidores de dados.

RDF estende a estrutura de links da WEB para utilizar URIs para nomear a relação entre “coisas”. Utilizando esse simples modelo, ele permite a junção, exposição e compartilhamento de dados estruturados ou semi-estruturados entre aplicações.

Para exemplificar, o exemplo apresentado na Figura 2 demonstra algumas linhas de um arquivo RDF.

---

<sup>5</sup> <http://www.w3.org/RDF/>

```

1  <?xml version="1.0"?>
2
3  <rdf:RDF
4  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5  xmlns:cd="http://www.recshop.fake/cd#">
6
7  <rdf:Description
8  rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
9    <cd:artist>Bob Dylan</cd:artist>
10   <cd:country>USA</cd:country>
11   <cd:company>Columbia</cd:company>
12   <cd:price>10.90</cd:price>
13   <cd:year>1985</cd:year>
14 </rdf:Description>
15
16 <rdf:Description
17 rdf:about="http://www.recshop.fake/cd/Hide your heart">
18   <cd:artist>Bonnie Tyler</cd:artist>
19   <cd:country>UK</cd:country>
20   <cd:company>CBS Records</cd:company>
21   <cd:price>9.90</cd:price>
22   <cd:year>1988</cd:year>
23 </rdf:Description>
24
25
26
27 </rdf:RDF>

```

**Figura 2** – Exemplo de arquivo RDF<sup>6</sup>

---

<sup>6</sup> [http://www.w3schools.com/rdf/rdf\\_example.asp](http://www.w3schools.com/rdf/rdf_example.asp)

A primeira linha do arquivo representa a declaração de que este se trata de um documento XML e, logo após (terceira linha), é seguido pelo elemento raiz de um documento RDF: `<rdf:RDF>`.

A quarta linha que começa com “`xmlns:rdf`”, especifica o prefixo “`rdf`” “`http://www.w3.org/1999/02/22-rdf-syntax-ns#`”. O uso de prefixo provê um mecanismo conveniente de organizar referências para conjuntos de termos. Esta declaração especifica que todas as *tags* com este conteúdo prefixado com `rdf` são parte do namespace identificado como <http://www.w3.org/1999/02/22-rdf-syntax-ns#>”.

A quinta linha que começa com “`xmlns:cd`”, especifica que os elementos com prefixo “`cd`” irão possuir namespace “`http://www.recshop.fake/cd#`”.

Os elementos `<rdf:Description>`, que estão nas linhas 7 e 16, contêm as propriedades do conceito identificado pelo atributo `rdf:about`. Neste exemplo, as descrições são sobre diferentes “`cd`”. Elementos filhos são utilizados para representar propriedades deste `<rdf:Description>`. Neste caso, os elementos: `<cd:artist>`, `<cd:country>`, `<cd:company>` e outros, são propriedades de “`http://www.recshop.fake/cd`”.

## 2.6 OWL

A OWL<sup>7</sup> (Ontology Web Language) é uma linguagem de marcação semântica para a publicação e compartilhamento de ontologias na Web. A OWL foi desenvolvida para ser uma extensão do vocabulário do RDF. Basicamente, ela é uma linguagem utilizada para definir e instanciar ontologias na Web, incluindo descrições de classes e suas respectivas propriedades e relacionamentos.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!DOCTYPE rdf:RDF [
4 <!ENTITY xsd http://www.w3.org/2001/XMLSchema#>
5 <!ENTITY rdf http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 <!ENTITY rdfs http://www.w3.org/2000/01/rdf-schema#>
7 <!ENTITY owl http://www.w3.org/2002/07/owl#>
8 <!ENTITY ns_transport file://www.ibm.com/WSRR/Transport#>
9 ]>
```

---

<sup>7</sup> <http://www.w3.org/TR/owl-ref/>

```

10
11 <rdf:RDF
12   xmlns:xsd="&xsd;"
13   xmlns:rdf="&rdf;"
14   xmlns:rdfs="&rdfs;"
15   xmlns:owl="&owl;"
16   xmlns:ns_transport="&ns_transport;"
17 >
18
19 <owl:Ontology rdf:about="&ns_transport;TransportOntology">
20   <rdfs:label>Transport</rdfs:label>
21   <rdfs:comment>Classe raiz de nível superior para
transporte.</rdfs:comment>
22 <owl:Class>
23
24 <owl:Class rdf:about="&ns_transport;LandTransport">
25   <rdfs:subClassOf rdf:resource="&ns_transport;Transport"/>
26   <rdfs:label>Transporte Terrestre.</rdfs:label>
27   <rdfs:comment>Classe de transport terrestre de nível
médio.</rdfs:comment>
28   </rdfs:comment>
29 </owl:Class>
30
31 <owl:Class rdf:about="&ns_transport;AirTransport">
32   <rdfs:subClassOf rdf:resource="&ns_transport;Transport"/>
33   <rdfs:label>Transporte Aéreo.</rdfs:label>
34   <rdfs:comment>Classe de transporte aéreo de nível
médio.</rdfs:comment>
35 </owl:Class>
36
37 <owl:Class rdf:about="&ns_transport;Bus">
38   <rdfs:subClassOf rdf:resource="&ns_transport;LandTransport"/>
39   <rdfs:label>Ônibus.</rdfs:label>
40   <rdfs:comment>Classe de ônibus e nível
inferior.</rdfs:comment>
41 </owl:Class>
42
43 <owl:Class rdf:about="&ns_transport;Car">
44   <rdfs:subClassOf rdf:resource="&ns_transport;Landtransport"/>
45   <rdfs:label>Carro.</rdfs:label>
46   <rdfs:comment>Classe de carro de nível
inferior.</rdfs:comment>
47 <owl:Class>
48
49 </rdf:RDF>

```

**Figura 3 - Exemplo de arquivo OWL**

Pode-se observar, na Figura 3, a declaração da ontologia na linha 19, pois é iniciada com a tag `<owl:Ontology>` e o nome está no atributo “`rdf:about=`” seguido com uma label na tag `<rdfs:label>` e um comentário sobre ela na tag `<rdfs:comment>`.

Após a declaração da ontologia, pode-se identificar a declaração de 5 classes, declaradas com a tag `<owl:class>` (linhas 24, 31, 37, 43). O nome dado a cada classe também está no atributo “`rdf:about=`” e sua label e comentário estão, respectivamente, na tag “`<rdfs:label>`” e “`<rdfs:comment>`”. Caso uma classe seja sub-classe de outra, esta classe declara no atributo “`rdf:resource=`” da tag `<rdfs:subClassOf>` o nome da classe pai (linhas 25, 32, 38, 44).

## 2.7 SWRL

Esta seção explica regras SWRL, que serão utilizadas para a criação de políticas através da ferramenta desenvolvida deste trabalho.

A SWRL<sup>8</sup> (Semantic Web Rule Language) é uma linguagem proposta para a Web Semântica, sendo usada para expressar lógica em forma de regras. As regras são expressas na forma de implicação entre um antecedente (*body*) e um conseqüente (*head*). A regra pode ser lida como: quando as condições do antecedente forem atendidas, as condições do conseqüente serão atendidas ou válidas.

“`temPai(?x1,?x2) ∧ temIrmão(?x2,?x3) ⇒ temTio(?x1,?x3)`” é um exemplo de regra escrita em SWRL

No exemplo acima, temos a declaração de uma regra SWRL. Sabendo que *x1*, *x2* e *x3* são instâncias de pessoas, podemos interpretar esta regra da seguinte forma: Se *x2* é pai de *x1* e *x2* tem um irmão *x3*, então *x3* é tio de *x1*. Nesta regra podemos definir o antecedente (*body*) como sendo o trecho “`temPai(?x1, ?x2) ∧ temIrmão(?x2, ?x3)`” e o conseqüente (*head*) como sendo o trecho “`temTio(?x1, ?x3)`”.

---

<sup>8</sup> <http://www.w3.org/Submission/SWRL/>

## **Capítulo 3: Proposta para modelagem da ferramenta**

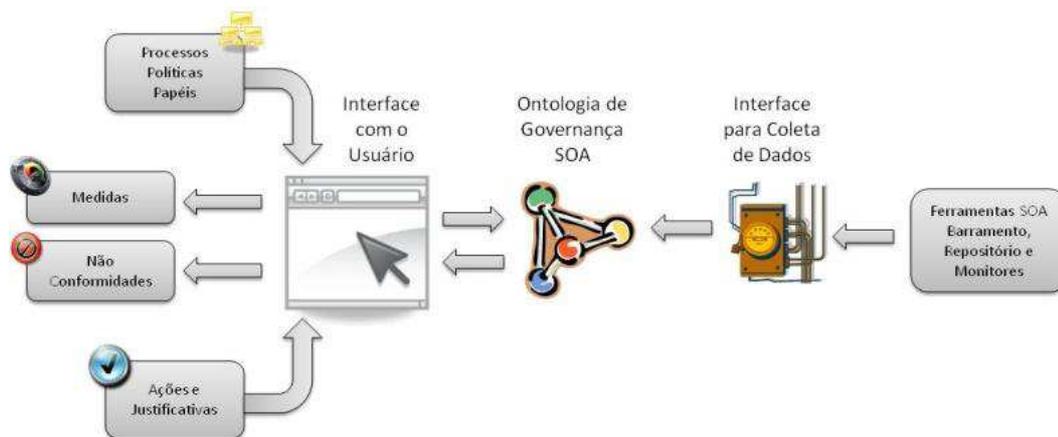
Nesta seção, iremos abordar as análises que foram realizadas por Teixeira Filho *et al.* (2013a, 2013b) para direcionar o início do desenvolvimento da ferramenta, tais como: levantamento dos casos de uso, especificação da ontologia e a escolha da arquitetura de implementação.

### **3.1 Método centrado em ontologias para governança de Arquitetura orientada a Serviços**

Teixeira Filho *et al.* (2013a, 2013b) propõem implementar a governança SOA com o uso de uma ontologia como base de conhecimento, possibilitando, o uso de mecanismos de inferência que viabilizam a monitoração e diagnósticos mais precisos do ambiente governado. Eles defendem que sua utilização irá possibilitar identificar ganhos de qualidade e reduções de esforços ao longo do processo de implantação e gestão.

A solução proposta por Teixeira Filho *et al.* [2013a, 2013b], utilizando uma ontologia como base central de conhecimento, serve como uma ferramenta, possibilitando documentar o cenário, agregar dados referentes ao ambiente SOA e inferir evoluções e conformidades com o modelo. As instâncias desta ontologia correspondem a informações sobre os elementos que compõe o modelo de governança, tais como: arquitetura; operação de serviços; padronização de dados; gerenciamento de ativos e projetos; tecnologia; segurança; finanças e regras; e, estruturas organizacionais. A criação das instâncias ocorre através de uma interface com o usuário e através de integrações com as ferramentas que suportam o ambiente SOA.

Teixeira Filho *et al.* (2013a, 2013b) propõem a construção de ferramentas com interface gráfica para o usuário e ferramentas para coleta de dados nas plataformas que compõe o ambiente SOA (tais como barramentos e repositórios) para inserção das instâncias referentes aos elementos de governança e para consolidação e apresentação dos resultados.



**Figura 4** - Arquitetura proposta por Teixeira Filho [2013a, 2013b]

A Figura 4 apresenta uma visão da arquitetura proposta. Para exemplificar, um usuário que queira adicionar uma nova política e verificar as não conformidades desta política, deverá adicioná-la através da interface com o usuário. Nessa mesma interface, ele irá gerar o relatório de não conformidades, possibilitando cadastrar justificativas e ações para estas não conformidades. Para possibilitar a geração desse relatório, serão necessários os dados que estão cadastrados na ontologia, no qual, a interface de coleta de dados irá coletar (através de barramentos, repositórios e monitores) e instancia na ontologia.

A partir da proposta feita por Teixeira Filho *et al.* (2013a, 2013b), foi definido que o escopo deste trabalho será desenvolver a interface com o usuário para viabilizar o cadastro de políticas, cadastro de ações e justificativas e emitir relatório de não conformidades.

### 3.2 Ontologia utilizada

A ontologia utilizada neste para representar os conceitos envolvidos segue a proposta do The Open Group (2010).



### 3.3 Casos de uso

Utilizando o documento de visão (Anexo I – Documento de visão) e entendendo os papéis definidos na organização, solução proposta e os requisitos de negócio, Teixeira Filho *et al.* (2013a, 2013b) elaboraram o diagrama de casos de uso apresentado na Figura 6.

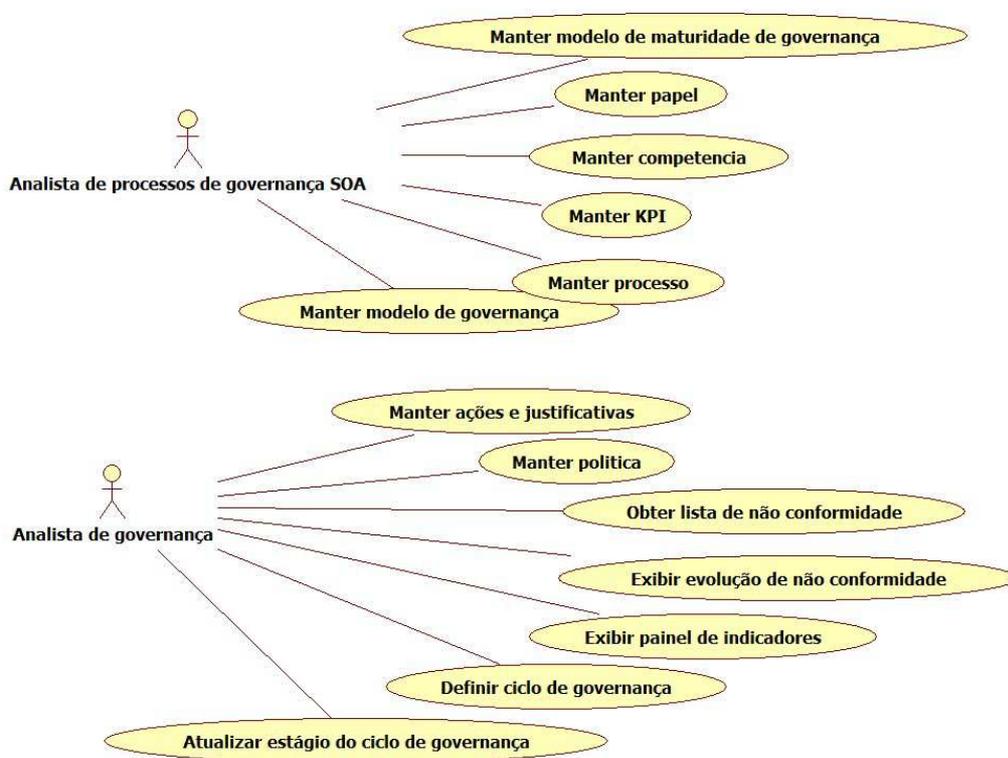


Figura 6 – Diagrama de casos de uso

## **Capítulo 4: Ferramenta para Governança SOA**

Neste trabalho, foi desenvolvido uma ferramenta que utiliza ontologias e regras semânticas (SWRL) para a verificação de conformidade de serviços desenvolvidos por uma organização.

### **4.1 Planejamento e desenvolvimento da ferramenta**

Neste trabalho, foi desenvolvido uma ferramenta que utiliza ontologias e regras semânticas (SWRL) para a verificação de conformidade de serviços desenvolvidos por uma organização.

Com os casos de uso apresentados na seção 3.3, foi determinado que os mais relevantes e essenciais para tratar as não conformidades na primeira versão da ferramenta seriam: manter políticas, obter lista de não conformidades, manter ações e justificativas. Após definir este escopo, cada caso de uso foi detalhado. As descrições dos mesmos são apresentadas no Apêndice I – Casos de uso.

A seguir são apresentados descrições resumidas destes casos de uso:

- Manter políticas: Caso de uso utilizado para a inclusão de políticas SWRL através de uma interface que ajuda o usuário a instanciá-las na ontologia.
- Obter lista de não conformidades: Caso de uso utilizado para emitir o relatório das instâncias que não estão conformes com as políticas que foram definidas.
- Manter justificativas: Caso de uso utilizado para inserir, visualizar, editar e remover justificativas para não conformidades existentes.
- Manter ações: Caso de uso utilizado para inserir, visualizar, editar e remover ações para resolver não conformidades.

Sabendo-se que já existem APIs que realizam a manipulações de ontologias em arquivos OWL (Jena<sup>9</sup>, Protégé-OWL API<sup>10</sup>) e inferências (Pellet<sup>11</sup>), foi decidido realizar a implementação da ferramenta em JAVA.

O atual protótipo desenvolvido não suporta a utilização de diversos arquivos OWL ao mesmo tempo, nem importações e nem diversos URIs distintos. Caso uma ontologia esteja dividida em diversos arquivos OWL, será necessário utilizar o Protégé para unificar os arquivos e seus URIs. Uma explicação de como realizar esse procedimento é apresentada no Apêndice II – Unificando URIs.

Para utilizar a ferramenta implementada, é necessário que o arquivo OWL contenha alguns requerimentos mínimos, como a ontologia proposta pelo The Open Group (2010) e algumas adaptações que foram feitas para o correto funcionamento das funcionalidades “Manter políticas”, “Manter justificativas” e “Manter ações”. Sendo assim, é recomendado que o usuário utilize o arquivo OWL<sup>12</sup>, disponibilizado junto com o código, e realize as alterações a partir dele.

As interfaces implementadas são detalhadas a seguir.

## 4.2 Página inicial (*Home*)

A página “Home” tem a função de referenciar qual arquivo OWL será utilizado e a URI que é usada neste arquivo, além de ser a página inicial da aplicação. Esta página possui 2 campos de entrada, que são “Arquivo” e “URI”, e um botão para adicionar o arquivo, apresentado na Figura 7.

---

<sup>9</sup> <http://jena.apache.org/documentation/ontology/>

<sup>10</sup> <http://protege.stanford.edu/plugins/owl/api/>

<sup>11</sup> <http://clarkparsia.com/pellet/>

<sup>12</sup> <https://code.google.com/p/verificador-conformidade-ontologia/>

Home [Manter políticas](#) [Obter lista de não conformidades](#) [Manter justificativas](#) [Manter ações](#)

Arquivo:  Ex: *C:/Users/Daniel/Desktop/SimpleGov/SimpleGov.owl*

URI:  Ex: *http://www.semanticweb.org/ontologies/2012/11/SimpleGov.owl#*

Arquivo:  
Prefixo do OWL (URI):

*Desenvolvido por Daniel Karam.*

**Figura 7** - Página inicial da aplicação

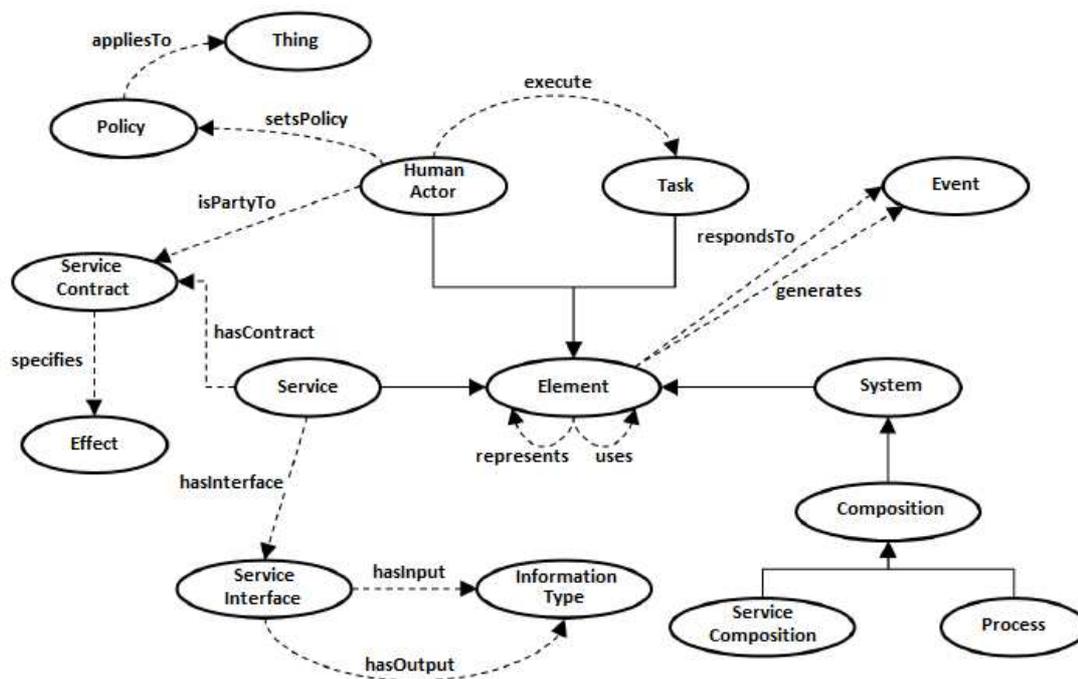
A URI é importante para a ferramenta, pois as APIs que estão sendo utilizadas na implementação devem saber a URI que está sendo utilizada no arquivo OWL para realizar a manipulação dos objetos da ontologia, tais como: criação/remoção de classes, propriedades e outras funções.

Após carregar o arquivo OWL e indicar a URI as opções “Manter política”, “Obter lista de não conformidades”, “Manter justificativas”, “Manter ações”. Todas essas opções foram especificadas nos casos de uso do capítulo anterior e estão detalhadas no Apêndice I – Casos de uso.

### 4.3 Manter políticas

A página “Manter políticas” tem a função de facilitar a adição de regras SWRL para o usuário. A página tem campos ao lado esquerdo e uma imagem da ontologia para ajudar o usuário a preencher e inserir regras SWRL válidas.

A Figura 8 apresenta o grafo da ontologia, em inglês, proposta pelo Open Group (2010) que representa os conceitos envolvidos em uma arquitetura orientada a serviços. Este grafo foi implementado para ser interativo, ou seja, ao clicar em um elemento, a descrição referente ao elemento é incluída no campo texto para descrição da regra.



**Figura 8** - Ontologia disponível na página "Manter políticas"

Para cadastrar uma política, é necessário preencher, em ordem, o nome da nova política (por exemplo, “Politica\_1”) e depois a classe que a política irá validar (por exemplo, “ServiceContract”). Após preencher esses campos, um trecho da regra SWRL será gerado na caixa de texto “Política SWRL”, apresentado na Figura 9.

Após realizar os primeiros passos essenciais, qualquer trecho da política SWRL poderá ser editada (para validação de políticas, é aconselhável somente mudar o trecho antes do operador “->”). A edição pode ser realizada tanto manualmente, quanto com as caixas de seleção “propriedades de dados”, “valores da propriedade selecionada”, “operadores” e utilizando a imagem interativa da ontologia apresentada na Figura 8.

Quando se clica em algum elemento da imagem da ontologia, automaticamente é preenchido o nome do elemento na caixa de texto “Política SWRL”, de acordo como está cadastrado na ontologia. Caso clique em uma classe, deverá ser nomeado (manualmente) uma variável. Se clicar em uma propriedade de objeto, deverá nomear (manualmente) duas variáveis.

Para exemplificar, se o usuário clicar na classe “Effect”, será preenchido “Effect(?)” na caixa de texto “Política SWRL”, sendo necessário nomear (manualmente) a variável. Por exemplo uma variável com nome “e” dentro dos parênteses, resultando em “Effect(?e)”.

---

[Home](#)   [Manter políticas](#)   [Obt](#)

---

### Adicionar política

Política

Classe que será validada

 | ⚡  
 | ⚡  
 | ⚡  
 | ⚡

Política SWRL

**Figura 9** - Exemplo de inclusão de uma nova política (parte 1)

Se o usuário clicar em uma propriedade de objeto, por exemplo “specifies”, será preenchido o nome da propriedade na caixa de texto “Política SWRL”, por exemplo, “specifies(?,?)”. É então necessário nomear (manualmente) duas variáveis que já foram declaradas anteriormente. Supondo que já exista uma variável “s” da classe “ServiceContract” e outra “e” da classe “Effect”, e queira validar se “s” especifica “e”, deve ser adicionado dentro dos parênteses “s” e “e”, resultando em “specifies(?s, ?e)” como apresentado na Figura 10.

## Adicionar política

Política

Classe que será validada

Política SWRL

```
ServiceContract(?s) ^ Effect(?e) ^ specifies(?s,?e) ->  
Politica_1(?s)
```

Incluir política

**Figura 10** - Exemplo de inclusão de uma nova política (parte 2)

A caixa de seleção “Propriedades de dados” somente é preenchida quando o usuário clicar em uma classe na imagem que tenha pelo menos uma propriedade de dado cadastrada na ontologia. Após este clique, esta caixa de seleção é preenchida, sendo permitido que o usuário selecione uma propriedade de dado. Quando se seleciona uma das propriedades, o nome é preenchido automaticamente na caixa de texto “Política SWRL”.

A caixa de seleção “Valores da propriedade selecionada” somente é preenchida quando for selecionado uma propriedade de dado na caixa de seleção “Propriedades de dados”, no qual exista alguma instância na ontologia que possua algum valor preenchido com esta propriedade de dado. Após esta caixa de seleção ter uma propriedade de dado escolhida, é possível selecionar algum valor. Quando se seleciona um dos valores, o valor é preenchido na caixa de texto “Política SWRL” automaticamente. Um exemplo é apresentado na Figura 11 e Figura 12. Neste caso, o usuário selecionou a propriedade de dados “deadline” na caixa de seleção “Propriedades de dados” (Figura 11). Em seguida, ele selecionou os valores que são apresentados para a propriedade de dados “deadline” (Figura 12).

---

## Adicionar política

### Política

### Classe que será validada

Propriedades de dados

- deadline
- executionDate
- actionDescription
- cancelDate

### Política SWRL

**Figura 11** - Preenchendo caixa de seleção "Valores da propriedade selecionada" (parte 1)

---

## Adicionar política

Política

Classe que será validada

Política SWRL

**Figura 12** - Preenchendo caixa de seleção "Valores da propriedade selecionada" (parte 2)

A caixa de seleção “Operadores” já vem preenchida com alguns valores fixos. Quando se seleciona um dos operadores, o operador é preenchido automaticamente na caixa de texto “Política SWRL”.

Caso a política que for inserir não seja válida, o sistema irá retornar o erro e a razão do mesmo. Se a política estiver correta, o sistema irá retornar com a mensagem “Política incluída com sucesso!”.

## 4.4 Obter lista de não conformidades

A página “Obter lista de não conformidades” tem a função de informar quais instâncias não estão conformes com as políticas cadastradas na ontologia. Na página temos o botão “Executar reasoner”, dois filtros (instâncias resolvidas e instâncias justificadas) e a tabela que apresenta as instâncias não conformes encontradas após a execução do reasoner.

Para realizar a verificação de instâncias não conformes, será necessário executar o “reasoner”. A sua primeira execução irá popular a tabela (que está localizada na mesma página) com todas as instâncias não conformes atualmente no sistema. A Figura 13, demonstra a tabela preenchida após a primeira execução.

### Emitir relatório de não conformidades

Executar reasoner

### Reasoner executado com sucesso!

Resolvida Sem filtro | ⌵ Justificada Sem filtro | ⌵ Filtrar

INSTANCIA	POLITICA	DATA	RESOLVIDA	JUSTIFICATIVA	DESCRIÇÃO
COMPANY_2.0	Service_ManagedBy_HumanActor	11/05/2013	Não		
COMPANY_1.0	Service_ManagedBy_HumanActor	11/05/2013	Não		
COMPANY_2.1	Service_ManagedBy_HumanActor	11/05/2013	Não		

Figura 13 - Lista de não conformidades (parte 1)

Na primeira coluna da tabela (INSTÂNCIA), temos o nome das instâncias não conformes encontradas na ontologia. A segunda coluna (POLÍTICA) informa a qual política a instância não está conforme. A terceira coluna (DATA) representa a data que a instância foi verificada não estar conforme. A quarta coluna (RESOLVIDA) informa se foram tomadas as devidas ações para tornar a instância conforme. Após a realização das ações, é necessário executar novamente o reasoner para verificar se a instância foi resolvida. A funcionalidade Manter ações é explicada na seção 4.5. A quinta coluna (JUSTIFICATIVA) informa, caso tenha sido adicionado, o nome da justificativa dada a instância não conforme. A sexta coluna (DESCRIÇÃO) informa a descrição da justificativa, caso tenha sido adicionado. A Figura 14 exemplifica uma segunda execução do reasoner com algumas informações preenchidas.

#### Emitir relatório de não conformidades

Executar reasoner

Resolvida Sem filtro |⇅ Justificada Sem filtro |⇅ Filtrar

INSTANCIA	POLITICA	DATA	RESOLVIDA	JUSTIFICATIVA	DESCRIÇÃO
COMPANY_1.0	Service_ManagedBy_HumanActor	11/05/2013	Sim		
COMPANY_2.0	Service_ManagedBy_HumanActor	11/05/2013	Não	COMPANY_2.0_Excessao	Este serviço ainda não tem um responsável.
COMPANY_2.1	Service_ManagedBy_HumanActor	11/05/2013	Não		

**Figura 14 -** Lista de não conformidades (parte 2)

Como exemplo, a partir da Figura 14, pode-se observar que a instância “COMPANY\_2.0” não está resolvida, porém foi adicionado uma justificativa. Isto significa que apesar de não conforme com a política, esta não conformidade representa uma exceção que foi justificada. Já a instância não conforme “COMPANY\_1.0” foi resolvida e agora está conforme com a política “Service\_ManagedBy\_HumanActor”, porém ela irá continuar na tabela com a finalidade de armazenar histórico.

## 4.5 Manter justificativas

A página “Manter justificativas” tem o propósito de adicionar, editar e remover as justificativas das instâncias não conformes que foram encontradas. A página possui 4 campos, que devem ser preenchidos para adicionar uma justificativa, e uma tabela com as justificativas que foram criadas, permitindo editá-las ou removê-las da tabela. A interface é apresentada na Figura 15.

---

[Home](#)   [Manter políticas](#)   (

---

### Adicionar justificativa

Política  
CompliantThingWithPolicy2 | ⌵

Instancia não conforme  
7G6R | ⌵

Justificativa  
Justificativa\_7G6R

Descrição  
Descrição da justificativa 7G6R

[Incluir justificativa](#)

**Figura 15** - Exemplo para incluir uma justificativa

Para adicionar uma justificativa, basta escolher uma política no campo de seleção “Política” (somente terá políticas que possuem instâncias não conformes), depois escolher uma instância não conforme na caixa de seleção “Instancia não conforme”, preencher um nome para a justificativa na caixa de entrada “Justificativa” e depois adicionar uma descrição na caixa de texto “Descrição” e pressionar o botão “Incluir justificativa”.

Pode-se observar também que caso uma instância não conforme já esteja justificada para uma determinada política, não será possível cadastrar outra justificativa para esta mesma instância nesta mesma política. Sendo assim, não será possível selecionar esta instância não conforme na caixa de seleção “Instância não conforme”.

Na tabela de justificativas, que está presente na mesma página, é possível visualizar todas as justificativas que foram cadastradas. Para cada justificativa presente na tabela, existe as opções editar e excluir. A Figura 16 ilustra a tabela de justificativas com alguns exemplos.

JUSTIFICATIVA	POLÍTICA	INSTÂNCIA	DESCRIÇÃO		
Justificativa_1.0	CompliantThingWithPolicy1	COMPANY_1.0	Ainda não tem responsável.	Editar	Excluir
Justificativa_2.0	CompliantThingWithPolicy2	EMRG	Ainda não tem um contrato emitido pelo responsável do serviço.	Editar	Excluir

**Figura 16** - Tabela de justificativas

Ao clicar em “Editar”, somente será possível editar o campo “Descrição” da justificativa clicada, não sendo possível editar os campos “Política”, “Instância não conforme” e “Justificativa”.

Ao clicar em “Excluir”, a justificativa será removida da lista.

## 4.6 Manter ações

A página “Manter ações” tem o propósito de adicionar, editar e remover as ações utilizadas para resolver as não conformidades. A página possui 6 campos, que devem ser preenchidos para adicionar uma ação, e uma tabela com as ações que foram criadas, permitindo editar ou cancelar estas ações pela tabela. A interface é apresentada na Figura 17.

## Adicionar ação

Política

CompliantThingWithPolicy1

Instancia não conforme

COMPANY\_1.0

Ação

Cadastrar\_responsavel\_COMPANY\_1.0

Responsável

TIC/PG/EP

Prazo

11/14/2013

Descrição

Deverá ser adicionado uma propriedade de objeto "ManagedBy" com um "HumanActor" para esta instância.

Incluir ação

**Figura 17** - Exemplo para incluir uma ação

Para adicionar uma ação, basta escolher uma política no campo de seleção “Política” (somente terá políticas que possuem instâncias não conformes), depois escolher uma instância não conforme na caixa de seleção “Instancia não conforme”, preencher um nome para a ação na caixa de entrada “Ação”, adicionar um responsável para a ação, uma prazo e depois adicionar uma descrição na caixa de texto “Descrição” e pressionar o botão “Incluir ação”.

Uma ação pode ter 4 possíveis situações que são: aberta, atrasada, encerrada e cancelada. A suas mudanças e condições estão explicadas no Apêndice IV – Diagrama de estados de uma ação.

Na tabela de ações, é possível visualizar todas as ações que foram cadastradas. Para cada ação presente na tabela, existe as opções editar e cancelar. A Figura 18 ilustra a tabela de ações com alguns exemplos.

AÇÃO	POLÍTICA	INSTÂNCIA	RESPONSÁVEL	DESCRIÇÃO	PRAZO	SITUAÇÃO
Cadastrar_emissor_1710	CompliantThingWithPolicy2	1710	TIC/CPTC/PPT	Deverá ser cadastrado uma emissor para o contratado permitindo o uso do serviço.	11/14/2013	Aberta
						Editar
						Cancelar
Cadastrar_responsavel_COMPANY_1.0	CompliantThingWithPolicy1	COMPANY_1.0	TIC/PG/EP	Deverá ser adicionado uma propriedade de objeto ManagedBy com um HumanActor para esta instância.	11/14/2013	Aberta
						Editar
						Cancelar

Figura 18 - Tabelas de ações

Ao clicar em “Editar”, somente será possível editar os campos “Responsável”, “Prazo”, “Descrição” e “Data da execução” da ação clicada, não sendo possível editar os campos “Política”, “Instância não conforme” e “Ação”. Não é possível editar ações que foram canceladas.

Ao clicar em “Cancelar”, a ação será cancelada, atualizando a sua situação para “Cancelada”. Não é possível cancelar ações que estão com a situação “Encerrada”.

## **Capítulo 5: Avaliação da Ferramenta**

Esta seção apresenta uma análise da ferramenta desenvolvida com o objetivo de verificar se consegue ser utilizada de forma prática e fácil por usuários, utilizando como base, os questionários que foram aplicados na avaliação de Célia Bastos (2010).

### **5.1 Avaliação utilizada**

Para realizar a avaliação da ferramenta, foram selecionados 4 participantes de diferentes áreas de atuação em TI, e para cada um, foi apresentada a ferramenta, explicando o contexto em que ela foi desenvolvida e qual é o seu objetivo.

Após a apresentação, cada participante foi instruído a seguir um roteiro de testes para avaliação que simulava um cenário real para avaliar o uso da ferramenta (roteiro localizado no Apêndice V – Roteiro de testes para avaliação).

Em seguida, o participante respondia um questionário sobre as funcionalidades implementadas e outro sobre a usabilidade (os modelos dos questionários estão no Apêndice VI – Questionários de avaliação).

A primeira etapa dos questionários (funcionalidades implementadas) visava verificar se cada funcionalidade estava realizando as suas tarefas propostas adequadamente e se cada funcionalidade é fácil e prática de utilizar. A segunda etapa (usabilidade) foi necessária para verificar se a ferramenta, como um todo, era prática para ser utilizada pelos usuários, com perguntas sobre a navegação, disposição das informações, feedbacks, estabilidade da ferramenta, satisfação do usuário e outras.

Após todos os questionários serem respondidos, foi realizada uma análise sobre os resultados coletados (respostas localizadas no Apêndice VII – Resposta dos participantes).

## 5.2 Análise dos resultados

Após acompanhar os testes com cada participante e analisar os resultados dos questionários, considerando que a resposta “concordo completamente” tenha pontuação 5 e a resposta “discordo completamente” tenha pontuação 1, foi verificado que as funcionalidades “Obter lista de não conformidades”, “Manter justificativas” e “Manter ações” são de fácil utilização, obtendo médias de pontuação 4.7, 4.6 e 4.6 respectivamente.

Já na funcionalidade “Manter políticas”, os participantes tiveram dificuldades em escrever uma nova política, por considerarem as opções para a adição de regras SWRL não tão práticas e intuitivas, mesmo com a imagem interativa da ontologia para SOA disponível para dar suporte. Os participantes conseguiam interpretar a imagem sem problemas, porém, por não conhece-la previamente, se perdiam ao tentar encontrar as classes e propriedades para clicar. Sendo assim, esta funcionalidade obteve média 4 em sua pontuação.

Quanto as perguntas de usabilidade, a média de pontuação das respostas foi 4.5 e foi verificado que os participantes tinham dificuldades de navegação no primeiro contato, porém após se familiarizarem, a navegação fluiu naturalmente.

## Capítulo 6: Conclusão

SOA é uma abordagem que está sendo bastante utilizada por corporações nos dias atuais para ajudar a atingir os seus objetivos. Porém a sua implementação não é simples e, sendo assim, é necessária uma governança SOA adequada para não comprometer a obtenção dos seus benefícios. No entanto, existem diversas propostas de modelos para governança SOA que são diferentes em aspectos e formas.

Nos trabalhos de Teixeira Filho e Azevedo (2012), foram realizadas análises sobre estas propostas para elaborar uma nova abordagem consolidada chamada CommonGOV. Porém, algumas atividades ainda representam grandes desafios mesmo utilizando uma governança SOA. Um exemplo é a verificação de conformidades que, segundo Niemann *et al.* (2010), é um dos fatores mais críticos para se obter os benefícios SOA.

O objetivo deste trabalho foi desenvolver uma ferramenta que viabilizasse esta atividade de verificação de conformidades, se baseando em uma aplicação prática feita por Teixeira Filho *et al.* (2013a) e no trabalho de Teixeira Filho *et al.* (2013b) utilizando ontologias e regras semânticas. O grande diferencial de ontologias como uma base de conhecimento é a possibilidade de interpretação por agentes computacionais, agilizando a atividade de verificação de conformidades.

A ferramenta desenvolvida ajuda a identificar as instâncias na ontologia que não estão conformes com as regras SWRL especificadas. Com essas instâncias não conformes, é possível adicionar ações e justificativas, agilizando o processo de validação destas, para assim, prover uma melhor governança e, conseqüentemente, ajuda a se realizar a verificação de conformidade da governança SOA.

Esta primeira versão da ferramenta é limitada para funcionar apenas em um arquivo OWL com apenas uma única URI unificada para todas as classes, entidades e propriedades. Após realizar a avaliação, pode-se concluir, em geral, que a ferramenta é prática e fácil de utilizar, porém é necessário aprimorar a funcionalidade “manter política” para facilitar a inclusão de novas políticas SWRL.

Este trabalho possibilita a continuidade de diversos trabalhos futuros. Dentre alguns que podem ser listados que irão aprimorar a gestão de uma governança SOA são:

- Melhorias na funcionalidade “Manter políticas” para ajudar o usuário a adicionar novas políticas com mais facilidade.
- Melhorias da interface front-end da ferramenta para aperfeiçoar o design e usabilidade.
- Implementação dos casos de usos restantes.
- Desenvolvimento das interfaces de coleta de dados em uma arquitetura SOA.

## Referências

CÉLIA BASTOS. **Avaliação de usabilidade em softwares educativos**. Fortaleza, 2010.

CHANDRASEKARAN, B.; JOHNSON, T. R.; BENJAMINS, V. R. **Ontologies: what are they? why do we need them?**. *IEEE Intelligent Systems*, 1999.

ERL. **Service-Oriented Architecture (SOA): Concepts, Technology, and Design**. Upper Saddle River, NJ, USA: Prentice Hall, 2005.

ERL. **SOA: Principles of Service Design**. Prentice Hall. T. Service Oriented ArchiOpen tecture, 2007.

GRUBER. **A translation approach to portable ontologies**. In Knowledge Acquisition, 1993.

GUARINO. **Formal Ontology, Conceptual Analysis and Knowledge Representation**. In. J. Hum.-Comput. Stud. 42, 6 (June 1995).

GU, Q., LAGO, P. **A stakeholder-driven service life cycle model for SOA**. In 2nd international Workshop on Service Oriented Software Engineering: in Conjunction with the 6th ESEC/FSE Joint Meeting, pp 1-7, Croatia, 2007.

HOJAJI, F., SHIRAZI, M.R. **A comprehensive SOA Governance Framework based on CobiT**. In: SERVICES-1, 6th Wold Congress on Services, Miami, EUA, 2010.

JANIESCH, C., KORTHAUS, A. e ROSEMANN, M. **Conceptualisation and facilitation of SOA governance**. In Proceedings of: ACIS 2009: 20th Australasian Conference on Information Systems, Melbourne, 2009.

JOHN F. SOWA. **"Semantic Networks"**. In Stuart C Shapiro. Encyclopedia of Artificial Intelligence. Retrieved 2008-04-29, 1987.

JOSUTTIS, N. **SOA in practice: The Art of Distributed System Design**, Beijing;Cambridge. O'Reilly, 2007.

KONTOGIANNIS, Kostas et al. **The landscape of service-oriented systems: A research perspective**. In: **Systems Development in SOA Environments**. SDSOA'07: ICSE Workshops 2007. International Workshop on. IEEE, 2007. p. 1-1.

MARKS, E. A.; BELL, M. **Service-Oriented Architecture: a planning and implementation guide for business and techonology**, John Willey & Sons Inc, 2006.

NIEMANN, M., MIEDE, A., JOHANNSEN, W., REPP, N. e Steinmetz, R. **Structuring SOA Governance**. International Journal of IT/Business Alignment and Governance, vol. 1, no. 1, 2010.

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

SCHEPERS, T. G. J., IACOB, M. E., ECK, P. A. T. Van. **A Lifecycle Approach To Soa Governance**. In: The 2008 ACM Symposium on Applied Computing. Fortaleza, Brasil, 2008.

SOWA, JOHN F. **Knowledge Representation: Logical, Philosophical, and Computational Foundations**, Brooks/Cole Publishing Co., Pacific Grove, CA, 2000.

TEIXEIRA FILHO, H. M., AZEVEDO, L. G. **CommonGOV: A Consolidate Approach for Governance of Service-Oriented Architecture**, 2012.

TEIXEIRA FILHO, H. M., AZEVEDO, L. G., SIQUEIRA, S. W. M. **Análise da Aplicabilidade do Uso de Ontologias e Regras Semânticas para Apoiar a Verificação de Conformidade no Contexto de Arquiteturas Orientadas a Serviço**. IX Simpósio Brasileiro de Sistemas de Informação (SBSI 2013), João Pessoa, PB, Brasil, 2013a.

TEIXEIRA FILHO, H. M., AZEVEDO, L. G., SIQUEIRA, S. W. M **Um Método Centrado em Ontologias para Implantação de Governança de Arquitetura Orientada a Serviço**. Workshop de Teses e Dissertações do IX Simpósio Brasileiro de Sistemas de Informação (WTDSI 2013), João Pessoa, PB, Brasil, 2013b.

THE OPEN GROUP, “**SOA Governance Framework**”, 2009. [Online]. Available: <https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?catalogno=c093>. [Accessed: 14-abr-2012].

## **Apêndice I – Casos de uso**

### **1. Caso de uso: Manter políticas**

**Atores:** Analista de Governança

**Descrição:** O usuário adiciona novas políticas SWRL

**Pré-condições:** N/A

**Pós-condições:** N/A

**Início:** Este caso de uso se inicia quando o usuário acessa a página “Manter políticas”.

#### **Fluxo principal:**

1. O usuário digita um nome para a política na caixa de entrada “política”
2. O usuário escolhe uma classe que a política irá validar da lista de seleção “Classe que será validada”
3. Sistema preenche um trecho da regra SWRL na caixa de texto “Política SWRL”
4. Usuário preenche a caixa de texto “Política SWRL” manualmente ou clicando nas classes e propriedades de objetos na imagem ou selecionando algum dos valores que estão nas listas de seleção “propriedades de dados”, “Valores da propriedade selecionada” e “Operadores”
5. Sistema retorna se a política foi inserida com sucesso ou não. Caso não tenha sido inserida com sucesso, o sistema retorna a razão de não ter sido inserida  
Fim do caso de uso

## 2. Caso de uso: Obter lista de não conformidades

**Atores:** Analista de Governança

**Descrição:** Emitir relatório das instâncias que não estão conformes com as políticas definidas.

**Pré-condições:** N/A

**Pós-condições:** N/A

**Início:** Este caso de uso se inicia quando o usuário acessa a página “Obter lista de não conformidades”.

### **Fluxo principal:**

#### **-Executar reasoner-**

1. O usuário clica no botão “Executar reasoner”
2. Sistema retorna se a execução do reasoner foi um sucesso ou não. Caso não tenha sido executada com sucesso, o sistema retorna a razão de não ter sido executada.

#### **-Visualizar lista de não conformidades-**

1. Estará disponível uma tabela com todas as instâncias não conformes encontradas no sistema com os seguintes dados.

-A primeira coluna representa as instâncias.

-A segunda coluna representa a qual política essa instancia não está de acordo.

-A terceira coluna representa a data em que essa instancia foi indentificada que não estava de acordo.

-A quarta coluna representa se a instância foi resolvida após ter sido indentificada como não conforme.

-A quinta coluna representa a justificativa dada a instância, caso possua.

-A sexta coluna representa a descrição da justificativa, caso possua.

Fim do caso de uso.

### **3. Caso de uso: Manter justificativas**

**Atores:** Analista de Governança

**Descrição:** Este caso de uso é utilizado para inserir, visualizar, editar e remover justificativas.

**Pré-condições:** N/A

**Pós-condições:** N/A

**Início:** Este caso de uso se inicia quando o usuário acessa a página “Manter justificativas”

#### **Fluxo principal:**

##### **-Inserir justificativa-**

1. O usuário escolhe uma das política, que somente estarão disponíveis na lista de seleção “política”, após executar o reasoner na página “Obter lista de não conformidades”
2. Sistema retorna todas as instancias não conformes com a política selecionada para a lista de seleção “Instancia”.
3. O usuário escolhe uma instancia da lista de seleção “Instancia”.
4. Usuário preenche os campos “Justificativa”, “Descrição”.
5. Usuário clica no botão “Inserir Justificativa”.
6. Sistema retorna se a justificativa foi inserida com sucesso ou não. Caso não tenha sido inserida com sucesso, o sistema retorna a razão de não ter sido inserida.

### **-Visualizar justificativas-**

1. Estará disponível uma tabela com todas as justificativas cadastradas no sistema com os seguintes dados.

- A primeira coluna representa o nome da justificativa.

-A segunda coluna representa a qual política essa justificativa referencia.

-A terceira coluna representa a instancia que essa justificativa referencia.

-A quarta coluna representa a descrição dessa justificativa.

-A quinta coluna apresenta as opções para excluir ou editar a justificativa.

### **-Editar justificativa-**

1. O usuário clica no botão “Editar” da justificativa que pretende editar, que está localizado na quinta coluna da tabela, que apresenta todas as justificativas cadastradas no sistema.

2. Sistema direcionada para uma interface de edição com os dados da respectiva justificativa selecionada.

3. Usuário edita o campo “Descrição”.

4. Usuário clica no botão “Editar justificativa”.

5. Sistema retorna se a justificativa foi editada com sucesso ou não. Caso não tenha sido editada com sucesso, o sistema retorna, a razão de não ter sido editada.

### **-Remover justificativa-**

1. O usuário clica no botão “Excluir” da justificativa que pretende remover, que está localizado na quinta coluna da tabela, que apresenta todas as justificativas cadastradas no sistema.

2. Sistema retorna se a justificativa foi removida com sucesso ou não. Caso não tenha sido removida com sucesso, o sistema retorna, a razão de não ter sido removida.

Fim do caso de uso.

### **4. Caso de uso: Manter ações**

**Atores:** Analista de Governança

**Descrição:** Este caso de uso é utilizado para inserir, visualizar, editar e remover ações.

**Pré-condições:** N/A

**Pós-condições:** N/A

**Início:** Este caso de uso se inicia quando o usuário acessa a página “Manter ações”

### **Fluxo principal:**

#### **-Inserir ação-**

1. O usuário escolhe uma das política, que somente estarão disponíveis na lista de seleção “política”, após executar o reasoner na página “Obter lista de não conformidades”

2. Sistema retorna todas as instancias não conformes com a política selecionada para a lista de seleção “Instancia”.

3. O usuário escolhe uma instancia da lista de seleção “Instancia”.

4. Usuário preenche os campos “Ação”, “Responsável”, “Prazo” e “Descrição”.

5. Usuário clica no botão “Inserir Ação”.

6. Sistema retorna se a ação foi inserida com sucesso ou não. Caso não tenha sido inserida com sucesso, o sistema retorna, a razão de não ter sido inserida.

### **-Visualizar ação-**

1. Estará disponível uma tabela com todas as ações cadastradas no sistema com os seguintes dados.

-A primeira coluna representa o nome da ação.

-A segunda coluna representa a qual política essa ação referencia.

-A terceira coluna representa a instancia que essa a ação referencia.

-A quarta coluna representa o responsável da criação desta ação.

-A quinta coluna representa a descrição da ação.

-A sexta coluna representa o prazo que essa ação deverá ser atendida.

-A sétima coluna representa a situação que essa ação se encontra.

-A oitava coluna apresenta as opções para editar ou cancelar a ação.

### **-Editar ação-**

1. O usuário clica no botão “Editar” da ação que pretende editar, que está localizado na oitava coluna da tabela que apresenta todas as ações cadastradas no sistema.

2. Sistema direcionada para uma interface de edição com os dados da respectiva ação selecionada.

3. Usuário edita os campos “Responsável”, “Prazo” e “Descrição” ou somente alguns destes campos a sua preferência.

4. Usuário clica no botão “Editar ação”.

5. Sistema retorna se a ação foi editada com sucesso ou não. Caso não tenha sido editada com sucesso, o sistema retorna, a razão de não ter sido editada.

### **-Cancelar ação-**

1. O usuário clica no botão “Cancelar” na ação que pretende cancelar, que está localizado na oitava coluna da tabela, que apresenta todas as ações cadastradas no sistema.

2. Sistema retorna se a ação foi cancelada com sucesso ou não. Caso não tenha sido cancelada com sucesso, o sistema retorna, a razão de não ter sido removida.

Fim do caso de uso

## Apêndice II – Unificando URIs

Para unificar as URIs, deveremos importar para o Protégé todos os arquivos OWL que iremos utilizar e depois clicar na aba “Refactor” e depois na opção “Rename multiple entities...” como está demonstrado na Figura 19.

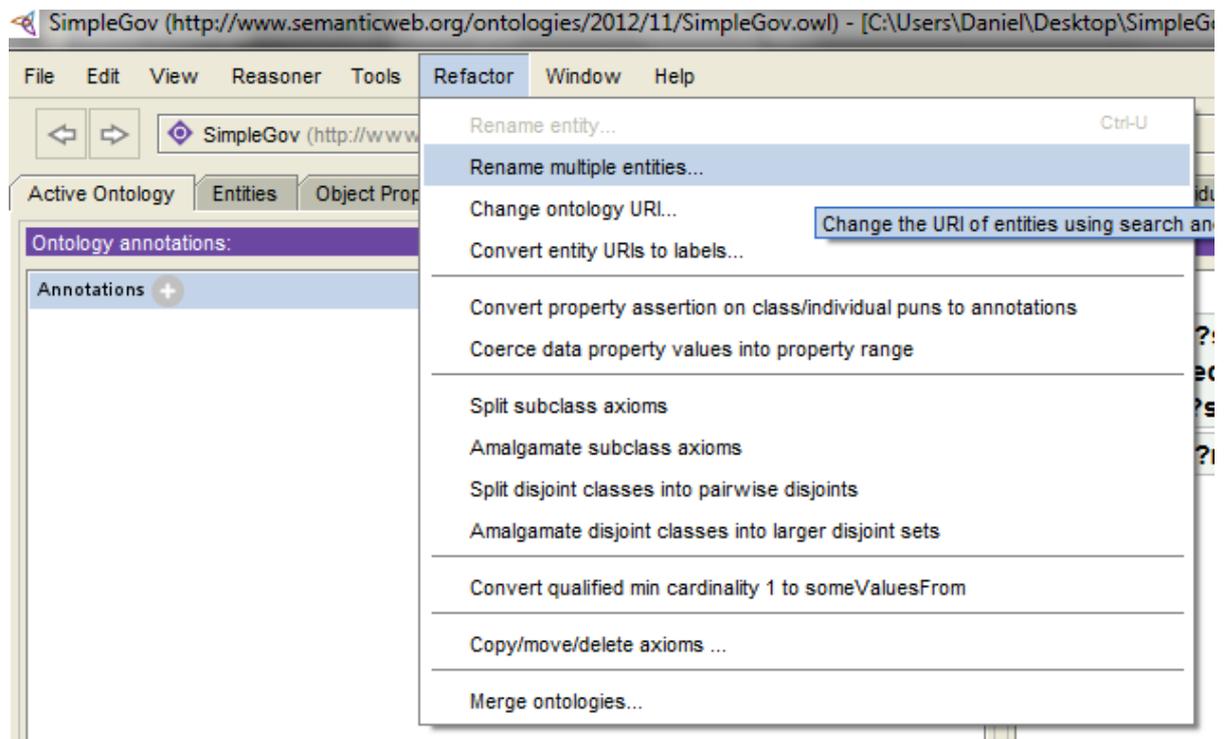


Figura 19 - Unificar diversas URI

O resultado, após essas operações, será um único arquivo OWL com todas as classes e propriedades com apenas um URI, tornando viável a utilização deste arquivo para a versão atual da ferramenta desenvolvida.

## Apêndice III – Preparando o ambiente

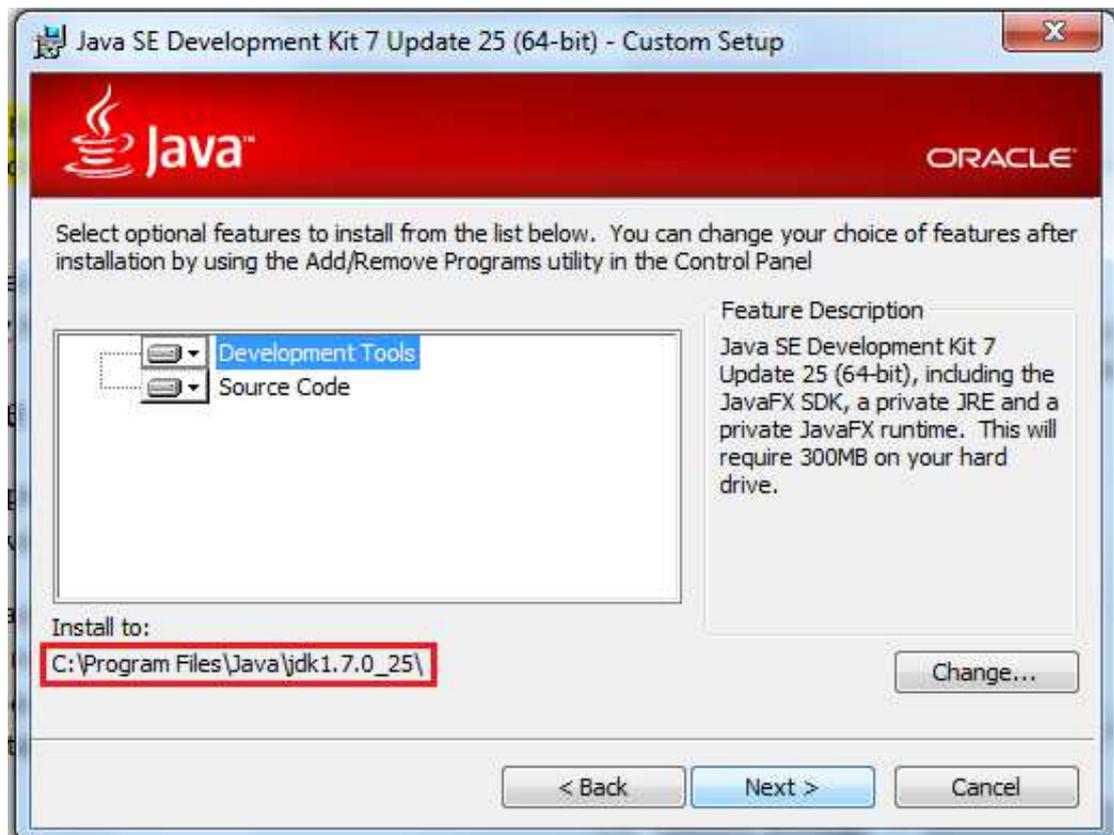
Antes de iniciar o desenvolvimento desta ferramenta, foram necessários os seguintes componentes listados abaixo. No caso, foi utilizado o sistema operacional Windows 7 x64

- JDK (Java development kit)  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html?ssSourceSiteId=otnpt>  
Contém os módulos necessários para o desenvolvimento de uma aplicação Java, também inclui o JRE (Java Runtime Environment) necessário para rodar as aplicações Java. Para este trabalho, foi utilizado o **JDK Java SE Development Kit 7u25**.
- Apache Tomcat  
<http://tomcat.apache.org/download-70.cgi>  
Servidor Web Java. Para este trabalho, foi utilizado a versão **7.0.42**. Para facilitar, foi realizado o download do instalador (32-bit/64-bit Windows Service Installer)
- Eclipse IDE for Java EE Developers  
<http://www.eclipse.org/downloads/>  
Ambiente necessário para o desenvolvimento em JAVA.
- Pellet  
<http://clarkparsia.com/pellet/download/pellet-2.3.1>  
“Reasoner” utilizado para realizar as inferências em uma ontologia a partir das “propriedades” e regras SWRL.
- Jena  
(Como iremos utilizar o Pellet, o mais aconselhável será utilizar o pacote Jena que vem junto com o download do pacote Pellet)  
Framework Java que possui um conjunto de ferramentas e bibliotecas Java para desenvolvimento de aplicações semânticas Web. É utilizado neste trabalho por facilitar a manipulação de ontologias através código Java, porém não possui a capacidade de adicionar regras SWRL.
- Protege-OWL API  
Outro conjunto de bibliotecas JAVA utilizado para desenvolvimento de aplicações semânticas Web. É utilizado neste trabalho por permitir a adição de regras SWRL.
- Arquivos .OWL  
Arquivos que possuem as descrições de classes e suas respectivas

propriedades e relacionamentos no formato OWL (Ontology Web Language).

Após estar com todos os componentes disponíveis, será necessário realizar as instalações e configurações necessárias que possibilitaram o desenvolvimento desta ferramenta.

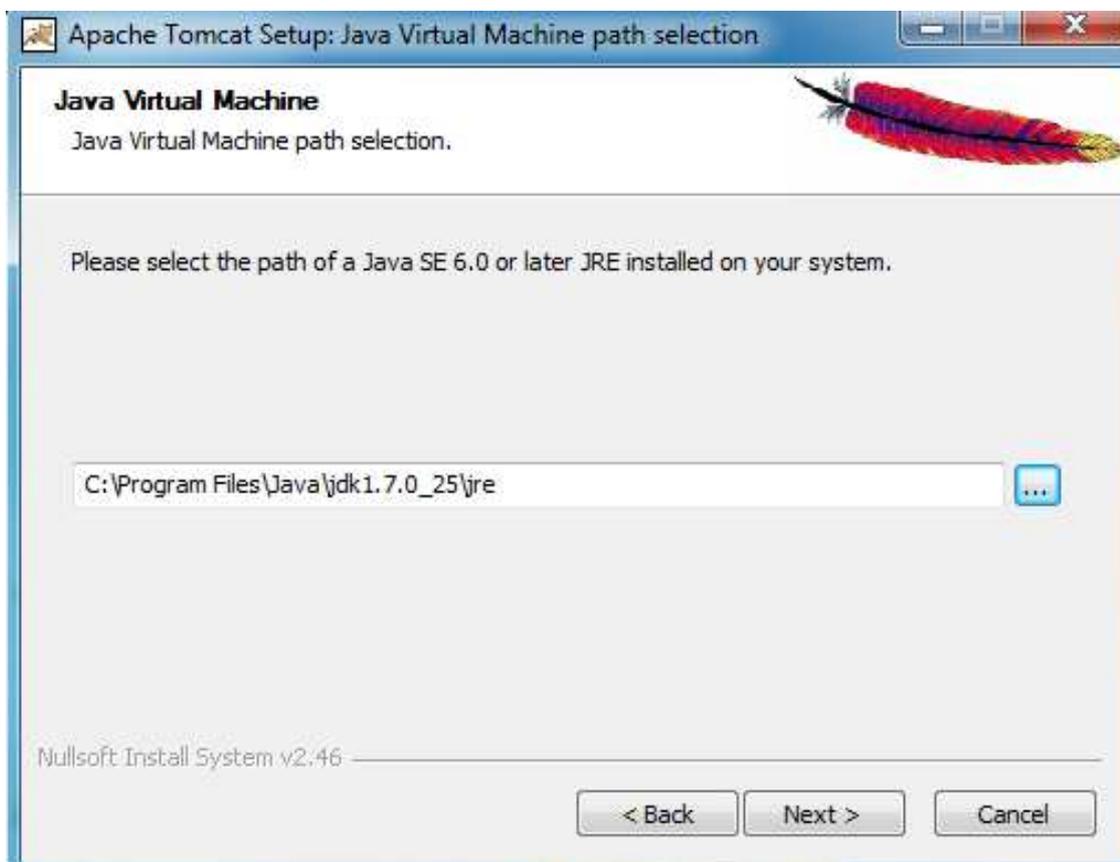
Primeiramente iremos instalar o JDK (Java development kit) que irá possibilitar o desenvolvimento JAVA (como explicado anteriormente no capítulo 4.1). Sua instalação é simples, e não será necessário realizar alterações no “setup” de instalação do JDK. Porém é necessário anotar o caminho físico da sua instalação, pois será necessário para o próximo passo. O caminho está destacado na Figura 20 dentro do retângulo vermelho.



**Figura 20** - Exemplo de caminho onde JDK será instalado

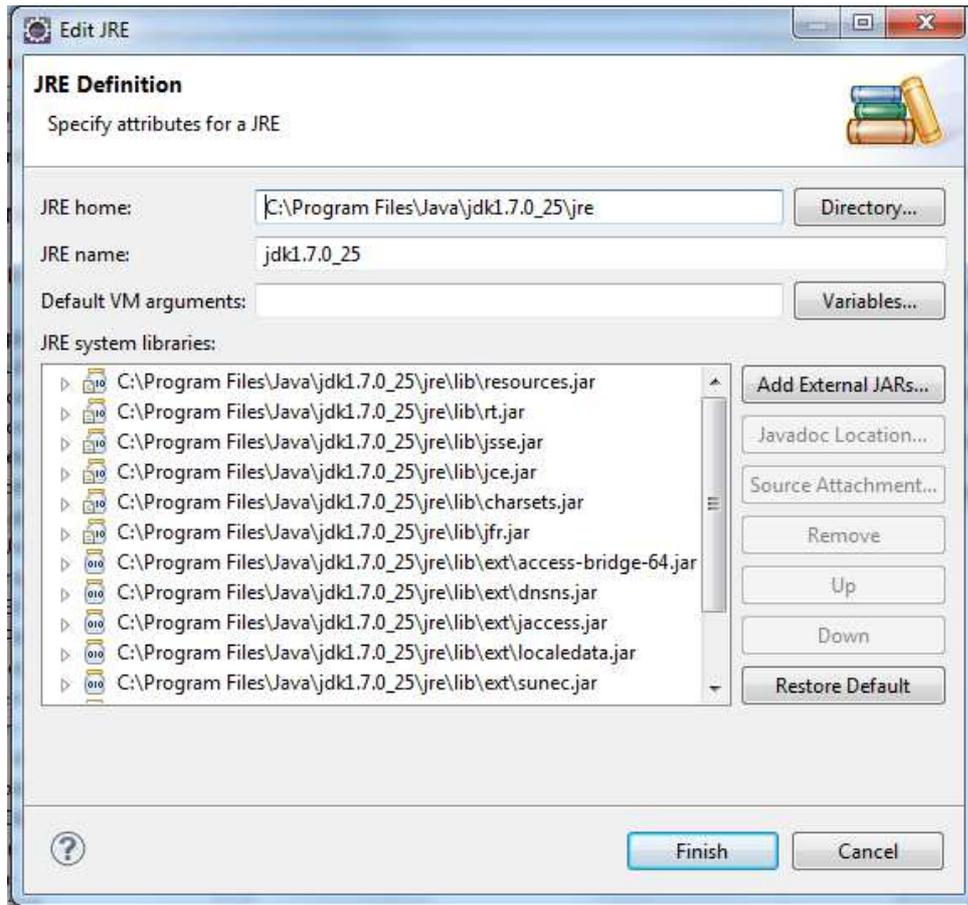
Após realizar a instalação do JDK, iremos instalar o apache tomcat. A instalação também é simples, porém quando o instalador pedir para selecionar uma JRE, será aconselhável utilizar a JRE que está dentro da pasta do JDK que foi

instalada no passo anterior. A Figura 21 demonstra o momento que o setup pede para selecionar o caminho da JRE, o caminho que deverá ser colocado, caso o JDK esteja instalado no caminho default, será “C:\Program Files\Java\jdk1.7.0\_25\jre”.



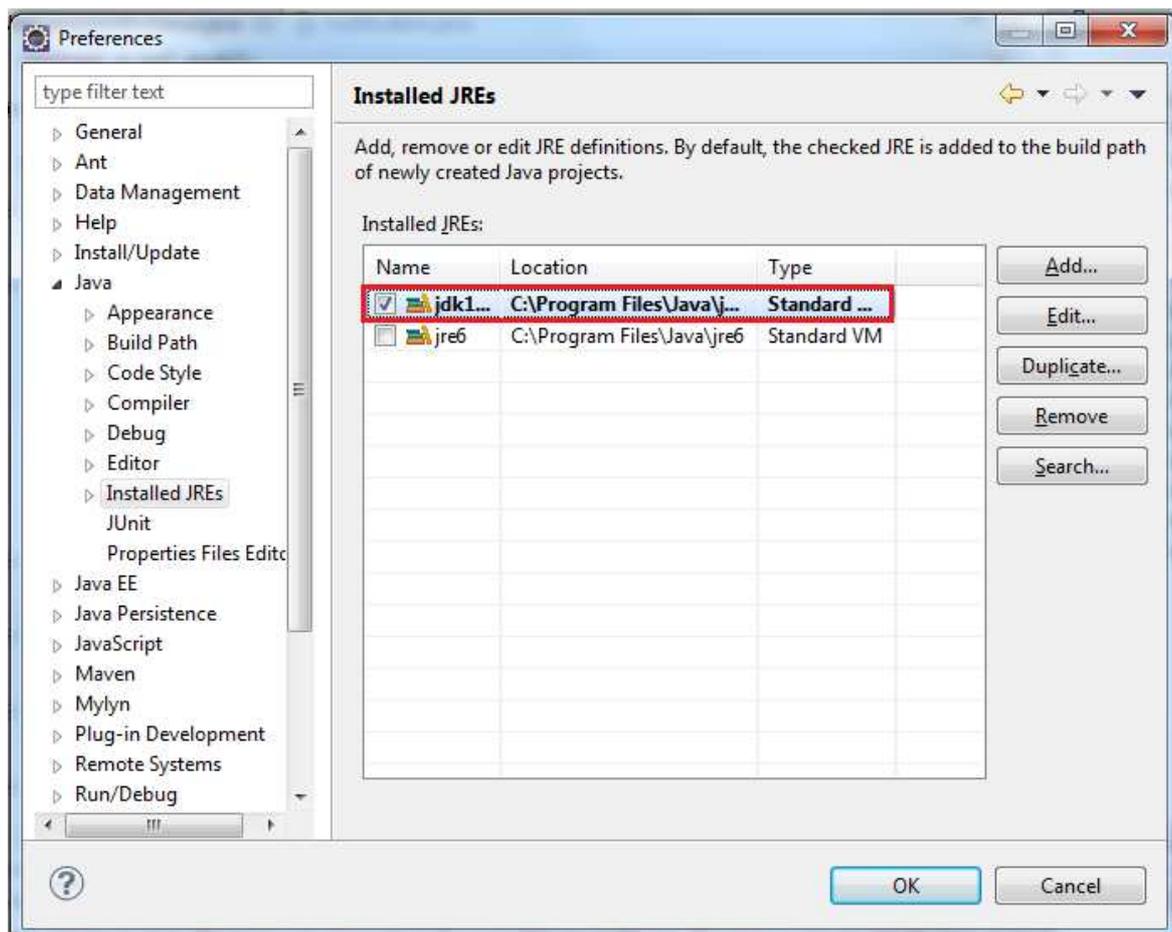
**Figura 21** - Especificação do caminho de instalação da JRE do JDK

Após realizar as instalações, será necessário configurar o eclipse para utilizar a JRE do JDK e o Apache Tomcat. Após descompactar a pasta “eclipse” e iniciá-lo, será necessário adicionar a JRE, que está na pasta do JDK, clicando na aba “Windows” e depois em “Preferences” e quando uma nova janela abrir, clicar na seta ao lado de “Java” para expandir as sub opções, e depois “Installed JREs” para abrir a listagem dos JREs. Quando estiver visualizando a listagem, deverá clicar em “Add” e na nova janela que abrir, selecionar “Standard VM” e clicar em “Next >”. Na nova janela que abrir, deverá ser colocada o caminho da JRE que está dentro da pasta do JDK, como está ilustrado na Figura 22 e, após configurar, clicar em “Finish”.



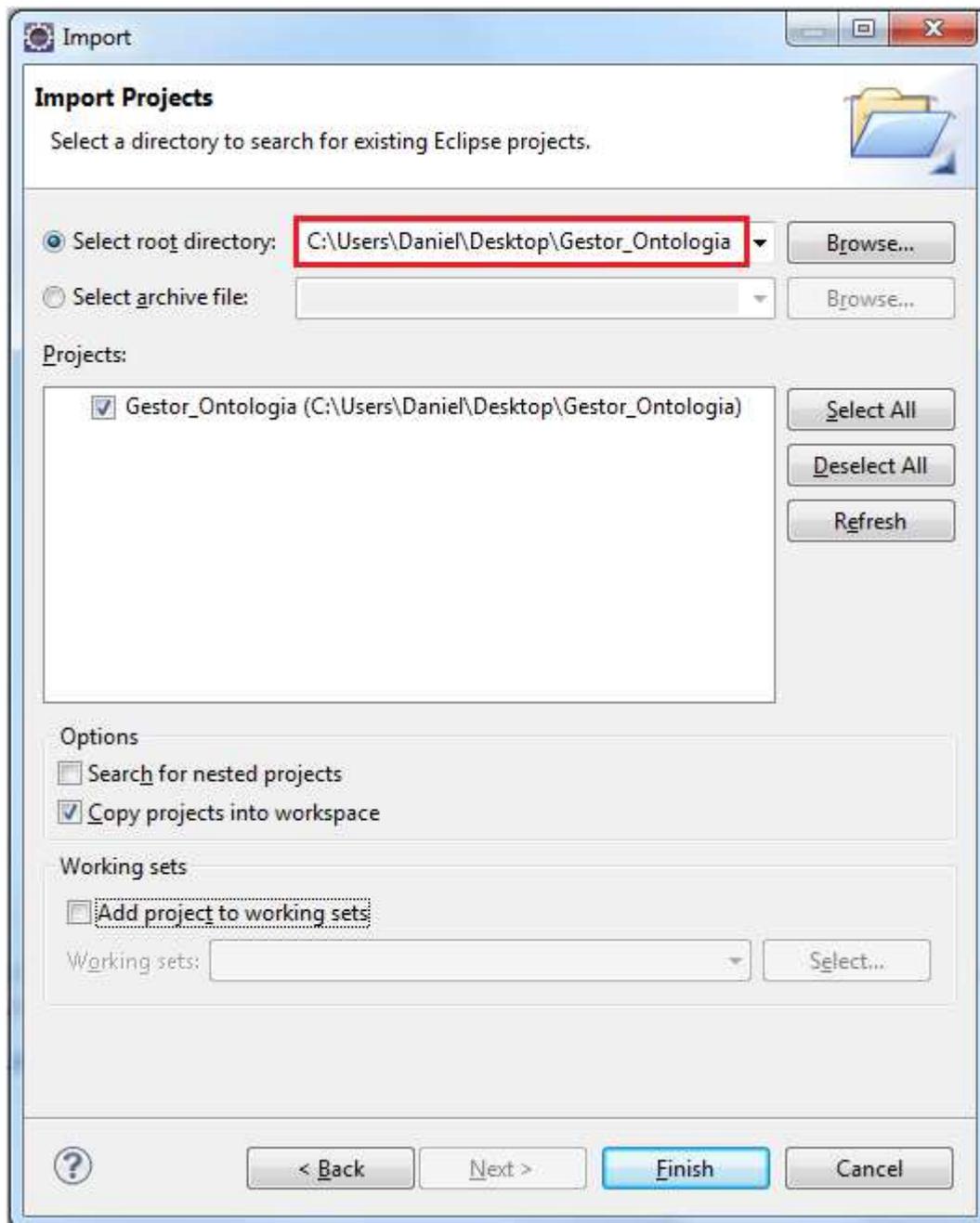
**Figura 22** - Configuração da JRE no eclipse

Após ter adicionado uma nova configuração do JRE, iremos marcar a checkbox do nome da nova configuração adicionada. A Figura 23 mostra, com um retângulo vermelho, como deve ser feito.



**Figura 23** - Marcar check-box da nova JRE configurada

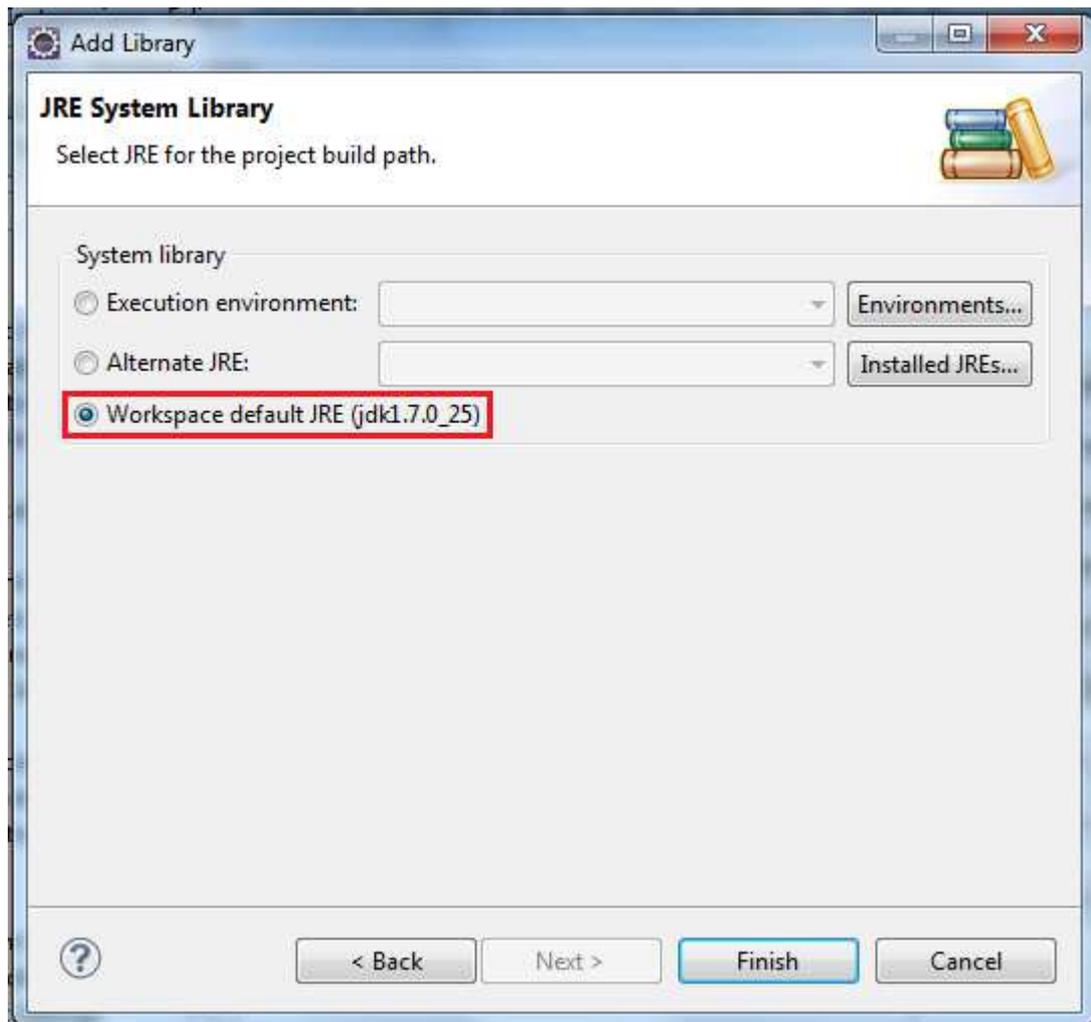
Próximo passo será realizar a importação da pasta que contém o projeto “Gestor\_Ontologia” com seus códigos, libs, .JSPs e demais arquivos necessários para seu funcionamento. Com o eclipse aberto, clique na aba “File” e depois “Import...”. Na janela que abre, expanda a pasta “General”, selecione a opção “Existing Projects into Workspace” e clique em “Next >”. Na próxima janela que abrir, clique em “Browse...” e indique o caminho do projeto “Gestor\_Ontologia” usando a Figura 24 abaixo como o exemplo (o caminho pode ser diferente dependendo de onde esteja localizado o projeto. A Figura 24 somente ilustra um exemplo).



**Figura 24** - Exemplo de caminho do projeto "Gestor\_Ontologia"

Quando a importação estiver completa, iremos configurar o nosso build path para utilizar o JRE que configuramos, anteriormente, no eclipse. Para isso, clicar com o botão direito no projeto importado "Gestor\_Ontologia" e no sub menu que abrir, clicar em "Build Path" e depois em "Configure Build Path". Na nova janela que abrir, clicar na aba "Libraries" e depois em "Add Library..." que está localizado juntos com os outro botões a direita. Na próxima janela que abrir, selecionar "JRE

System Library” e clicar em “Next >”. Irá abrir outra janela, e deve ser verificado se está selecionada a configuração do JRE que foi realizado anteriormente. Se estiver correta, deve manter ela selecionada e clicar no botão “Finish”. A Figura 25 abaixo mostra a tela e a opção, com retângulo vermelho, que (caso esteja com caminho default correto) deve ser selecionada.



**Figura 25 - Caminho default da JRE selecionada**

Após adicionar o JRE ao projeto, iremos adicionar o servidor web para executar o projeto. Para adicionar um servidor web, o eclipse deve estar na perspectiva “Java EE” como está demonstrado na Figura 26, no primeiro retângulo vermelho que se encontra no canto direito acima. Após selecionar a perspectiva,

iremos clicar na aba "Servers" e, em seguida, clicar no link escrito "No servers are available. Click this link to create a new server..." (Dentro do segundo retângulo vermelho da Figura 26). Após clicar no link, irá abrir uma nova janela para escolher o tipo de servidor que irá executar o projeto. Para este projeto, deve ser escolhido o servidor apache com a versão que está instalada na máquina. Após escolher, clicar em "Next >".

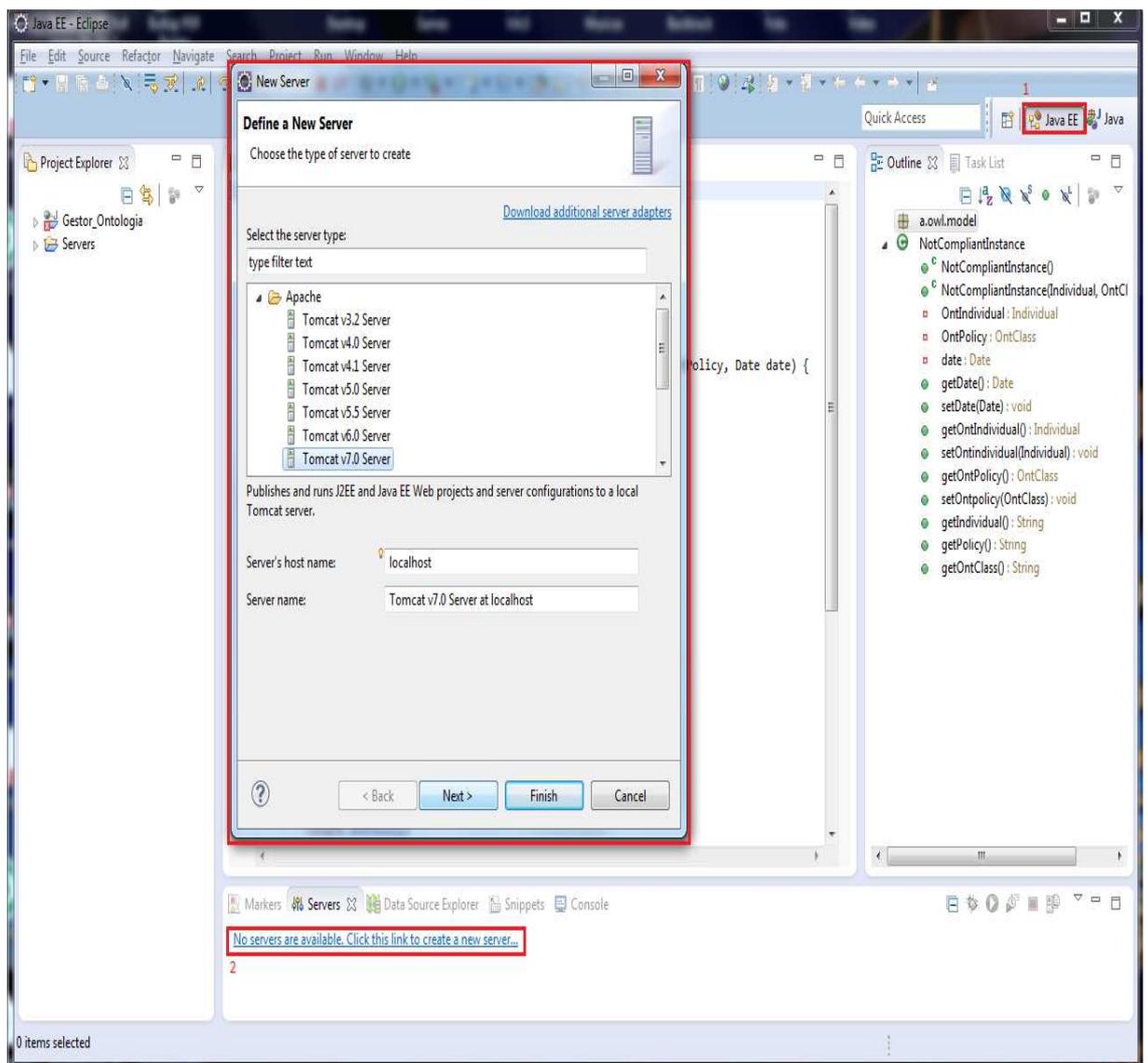
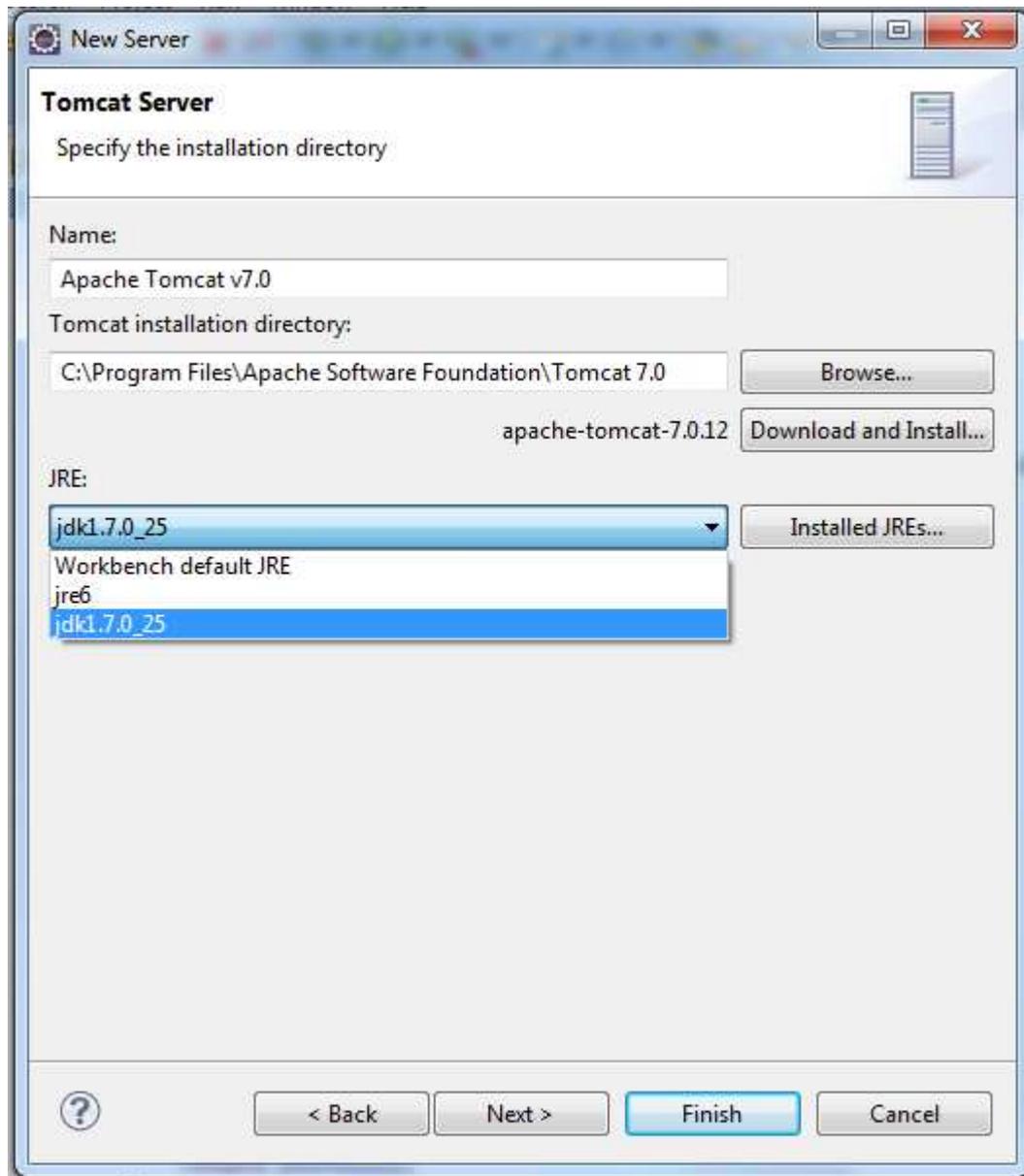


Figura 26 - Adicionando servidor web (parte 1)

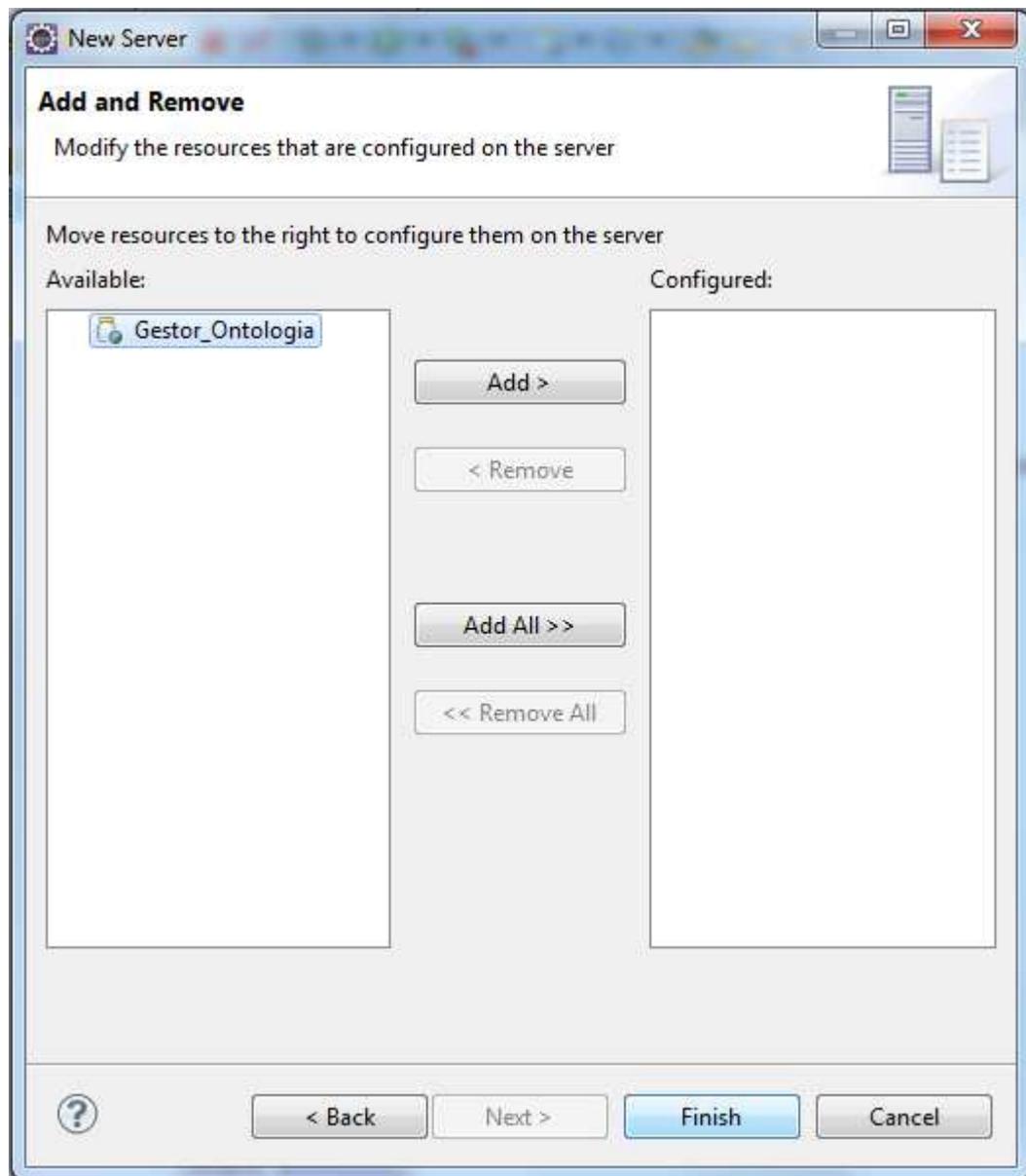
Após clicar em “Next >”, iremos para outra janela que irá apresentar novas opções. Nela será necessário colocar a localização de instalação do servidor tomcat e selecionar a JRE. A Figura 27 mostra um exemplo de como realizar esta configuração. Após selecionar as opções, deverá clicar no botão “Next >”.



**Figura 27** - Adicionando servidor web (parte 2)

Após clicar no botão “Next >”, irá aparecer uma nova janela. Nesta janela, deve-se selecionar o projeto que o servidor apache irá rodar, neste caso será o “Gestor\_Ontologia” como está demonstrando na Figura 28. Após selecionar o projeto,

deverá clicar no botão “Add >” para que o projeto passe para a aba ao lado. Após essas ações, clicar no botão “Finish”.

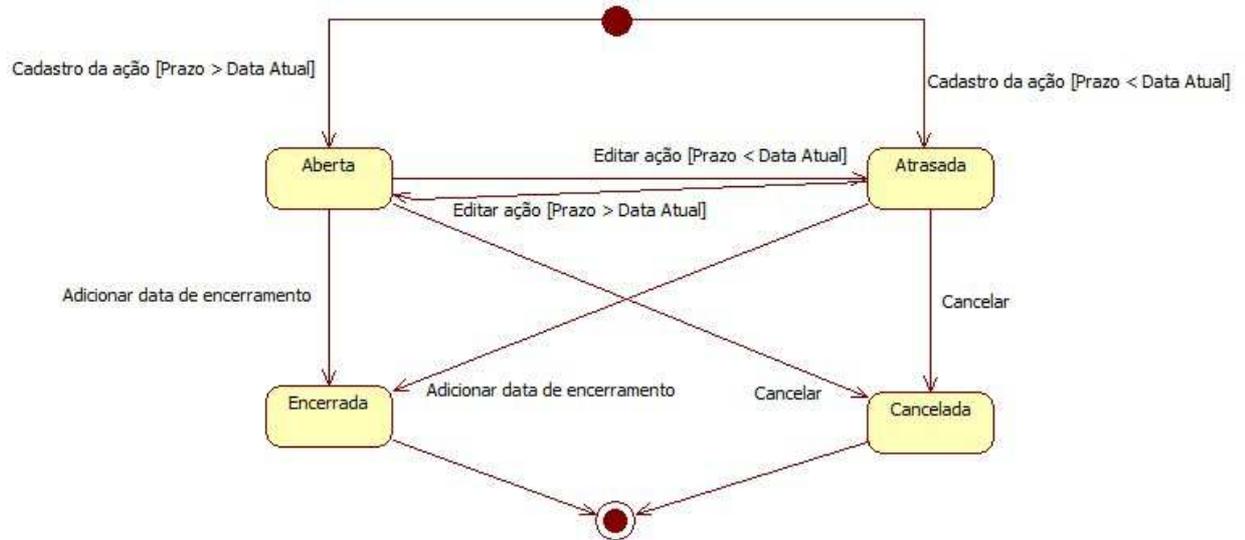


**Figura 28** - Adicionando servidor web (parte 3)

Após todas as configurações estarem completas, o projeto estará pronto para ser executado! Para verificar se tudo foi configurado corretamente, o servidor irá iniciar automaticamente após a configuração do passo anterior. Quando o servidor estiver iniciado, abra qualquer navegador e digite na barra de endereços (URL)

“http://localhost:8080/Gestor\_Ontologia” . Se tudo estiver correto, a página irá abrir na página inicial “index.jsp”.

#### Apêndice IV – Diagrama de estados de uma ação



## **Apêndice V – Roteiro de testes para avaliação**

### **1- Cadastrar políticas**

Política CompRule1 - Todo serviço deve ter um contrato que especifica os efeitos ocasionados pelo serviço:

$Service(?s) \wedge ServiceContract(?sc) \wedge Effect(?e) \wedge isContractFor(?sc, ?s) \wedge specifies(?sc, ?e) \rightarrow CompRule1(?s)$

Política CompRule2 - Todo serviço deve dispor de um contrato que contenha alguma área da organização (Human Actor) como parte envolvida:

$Service(?s) \wedge ServiceContract(?sc) \wedge HumanActor(?ha) \wedge isContractFor(?sc, ?s) \wedge involvesParty(?sc, ?ha) \rightarrow CompRule2(?s)$

Política CompRule3 - Todo serviço deve conter em sua interface pelo menos uma informação de entrada:

$Service(?s) \wedge ServiceInterface(?si) \wedge InformationType(?it) \wedge hasInterface(?s, ?si) \wedge hasInput(?si, ?it) \rightarrow CompRule3(?s)$

### **2- Visualizar lista de não conformidades**

Executar o reasoner na página “obter lista de não conformidades” e explicar os dados que são apresentados na tela.

### **3- Manipulação de justificativas**

Adicionar 3 justificativas nas instâncias não conformes encontradas, sendo uma delas com o nome “Justificativa\_COMPANY\_1.0” para a instância “COMPANY\_1.0” da política “CompRule2” com a descrição “Ainda não foi definido uma área responsável da organização para o contrato ‘Service\_Contract\_1.0’ ”.

Visualizar se as justificativas aparecem na tabela de justificativas da página atual e na tabela da página “Obter lista de não conformidades”.

Editar a descrição de uma justificativa e verificar o resultado na tabela de justificativas da página atual e na tabela da página “Obter lista de não conformidades”.

Remover a justificativa da instancia “COMPANY\_1.0” da política “CompRule2” e verificar o resultado na tabela de justificativas da página atual e na tabela da página “Obter lista de não conformidades”

#### **4- Manipulação de ações**

Adicionar 3 ações nas instâncias não conformes encontradas, sendo uma delas com o nome “Acao\_1\_COMPANY\_1.0” para a instância “COMPANY\_1.0”, da política “CompRule2”, com responsável “TIC/CPIE/SID”, prazo “12/10/2013” descrição “Adicionar a propriedade de objeto “involvesParty” para a instância ‘Service\_Contract\_1.0’ com o valor ‘RH’ ”.

Visualizar se as ações aparecem na tabela de ações da página atual.

Editar a descrição, responsável e prazo (para uma data anterior da atual) de uma ação e verificar o resultado na tabela de ações da página atual e explicar por que a situação mudou de estado

Editar a data de execução de uma ação e verificar o resultado na tabela de ações da página atual e explicar por que a situação mudou de estado e, após a explicação, clicar na opção “Cancelar” desta mesma ação e explicar por que não foi cancelada.

Cancelar uma ação e verificar o resultado na tabela de ações da página atual e, após cancelar, clicar na opção “Editar” desta mesma ação e explicar por que não pode ser editada.

## **Apêndice VI – Questionários de avaliação**

### **-Manter políticas-**

- 1 - A adição de um nome no campo "política" é fácil e prático.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 2 - O campo "Operadores" é prático e fácil de usar.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 3 - A imagem está fácil e prática de usar.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 4 - A edição do campo "Política SWRL" está fácil e prática.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 5 - Em geral, a criação da política é prática e intuitiva.  
 concordo completamente  concordo  neutro  discordo  discordo completamente

### **-Obter lista de não conformidade-**

- 1 - Ao executar o reasoner, a listagem foi apresentada de forma apropriada e esperada.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 2 - Os filtros estão funcionando de forma apropriada e esperada.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 3 - Em geral, a listagem de não conformidades é prática e intuitiva de se usar.  
 concordo completamente  concordo  neutro  discordo  discordo completamente

### **-Manter justificativas-**

- 1 - As listagem de seleção "Política" e "Instâncias não conforme" são fáceis e práticas de usar.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 2 - A adição de um nome no campo "justificativa" é fácil e prático.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 3 - A tabela está intuitiva e as opções de editar e excluir são fáceis de se utilizar.  
 concordo completamente  concordo  neutro  discordo  discordo completamente
- 4 - Em geral, a adição, edição e exclusão de justificativas são fáceis e práticas de se utilizar.  
 concordo completamente  concordo  neutro  discordo  discordo completamente

### **-Manter ações-**

1 - As listagem de seleção "Política" e "Instâncias não conforme" são fáceis e práticas de usar.

concordo completamente  concordo  neutro  discordo  discordo completamente

2 - A adição de um nome no campo "Ação" é fácil e prático.

concordo completamente  concordo  neutro  discordo  discordo completamente

3 - A adição de uma data no campo "Prazo" é fácil e prático.

concordo completamente  concordo  neutro  discordo  discordo completamente

4 - A tabela está intuitiva e as opções de editar e cancelar são fáceis de se utilizar.

concordo completamente  concordo  neutro  discordo  discordo completamente

5 - Em geral, a adição, edição e cancelamento de ações são fáceis e práticas de se utilizar.

concordo completamente  concordo  neutro  discordo  discordo completamente

### **-Usabilidade-**

1 - A navegação pelo site é fácil e prático.

concordo completamente  concordo  neutro  discordo  discordo completamente

2 - Os links das páginas são consistentes com os títulos das páginas as quais estão vinculadas

concordo completamente  concordo  neutro  discordo  discordo completamente

3 - É fácil identificar onde se está localizado e quais opções deve seguir.

concordo completamente  concordo  neutro  discordo  discordo completamente

4 - AS informações correlacionadas estão agrupadas de forma correta.

concordo completamente  concordo  neutro  discordo  discordo completamente

5 - As informações estão organizadas em uma ordem natural e lógica.

concordo completamente  concordo  neutro  discordo  discordo completamente

6 - A entrada de informações no sistema é fácil e prática.

concordo completamente  concordo  neutro  discordo  discordo completamente

7 - Consigo ver, com precisão, os resultados de cada ação que realizo através de feedbacks.

concordo completamente  concordo  neutro  discordo  discordo completamente

8 - O sistema reage da forma esperada, sem surpreender com comportamentos inesperados.

concordo completamente  concordo  neutro  discordo  discordo completamente

9 - O sistema desempenha as tarefas apropriadamente.

concordo completamente  concordo  neutro  discordo  discordo completamente

10 - É rápido de se trabalhar utilizando a ferramenta.

concordo completamente  concordo  neutro  discordo  discordo completamente

11 - Uma vez que aprendeu a utilizar o sistema, será fácil utiliza-lo na próxima vez.

concordo completamente  concordo  neutro  discordo  discordo completamente

12 - Eu fiquei satisfeito com o sistema.

concordo completamente  concordo  neutro  discordo  discordo completamente

## **Apêndice VII – Resposta dos participantes**

### Participantes e áreas de atuação em TI

Geraldo Crelier – Administração de ambiente de TI

Guilherme Albuquerque – Desenvolvimento / Testes

Hiran Azevedo – Desenvolvedor da área de integração

Julio Cesar – Banco de dados e BI

### **-Manter políticas-**

1 - A adição de um nome no campo "política" é fácil e prático.

[Hiran/Julio] concordo [Geraldo/Guilherme] neutro

2 - O campo "Operadores" é prático e fácil de usar.

[Hiran] concordo [Geral/Guilherme/Julio] neutro

3 - A imagem está fácil e prática de usar.

[Geraldo/Guilherme/Julio] concordo completamente [Hiran] concorda

4 - A edição do campo "Política SWRL" está fácil e prática.

[Guilherme/Hiran] concordo completamente [Geraldo] concordo [Julio] neutro

5 - Em geral, a criação da política é prática e intuitiva.

[Hiran] concordo completamente [Geraldo/Guilherme/Julio] concordo

### **-Obter lista de não conformidade-**

1 - Ao executar o reasoner, a listagem foi apresentada de forma apropriada e esperada.

[Geraldo/Julio] concordo completamente [Guilherme/Hiran] concordo

2 - Os filtros estão funcionando de forma apropriada e esperada.

[Geraldo/Guilherme/Hiran/Julio] concordo

3 - Em geral, a listagem de não conformidades é prática e intuitiva de se usar.

[Geraldo/Hiran] concordo completamente [Guilherme/Julio] concordo

### **-Manter justificativas-**

1 - As listagem de seleção "Política" e "Instâncias não conforme" são fáceis e práticas de usar.

[Guilherme] concordo completamente [Geraldo/Hiran/Julio] concordo

2 - A adição de um nome no campo "justificativa" é fácil e prático.

[Geraldo/Guilherme/Hiran/Julio] concordo completamente

3 - A tabela está intuitiva e as opções de editar e excluir são fáceis de utilizar.

[Geraldo/Guilherme/Hiran/Julio] concordo completamente

4 - Em geral, a adição, edição e exclusão de justificativas são fáceis e práticas de se utilizar.

[Guilherme] concordo completamente [Geraldo/Hiran/Julio] concordo

### **-Manter ações-**

1 - As listagem de seleção "Política" e "Instâncias não conforme" são fáceis e práticas de usar.

[Guilherme] concordo completamente [Geraldo/Hiran/Julio] concordo

2 - A adição de um nome no campo "Ação" é fácil e prático.

[Geraldo/Guilherme/Hiran/Julio] concordo completamente

3 - A adição de uma data no campo "Prazo" é fácil e prático.

[Guilherme/Julio] concordo completamente [Geraldo/Hiran] concordo

4 - A tabela está intuitiva e as opções de editar e cancelar são fáceis de utilizar.

[Geraldo/Guilherme/Hiran/Julio] concordo completamente

5 - Em geral, a adição, edição e cancelamento de ações são fáceis e práticas de se utilizar.

[Guilherme] concordo completamente [Geraldo/Hiran/Julio] concordo

## **-Usabilidade-**

1 - A navegação pelo site é fácil e prático.

[Geraldo] concordo completamente [Guilherme/Hiran/Julio] concordo

2 - Os links das páginas são consistentes com os títulos das páginas as quais estão vinculadas

[Geraldo] concordo completamente [Guilherme/Hiran/Julio] concordo

3 - É fácil identificar onde se está localizado e quais opções deve seguir.

[Hiran] concordo completamente [Geraldo/Guilherme/Julio] concordo

4 - AS informações correlacionadas estão agrupadas de forma correta.

[Guilherme] concordo completamente [Geraldo/Hiran/Julio] concordo

5 - As informações estão organizadas em uma ordem natural e lógica.

[Julio] concordo completamente [Geraldo] concordo [Guilherme/Hiran] neutro

6 - A entrada de informações no sistema é fácil e prática.

[Julio] concordo completamente [Geraldo/Guilherme/Hiran] concordo

7 - Consigo ver, com precisão, os resultados de cada ação que realizo através de feedbacks.

[Geraldo/Guilherme/Hiran] concordo completamente [Julio] concordo

8 - O sistema reage da forma esperada, sem surpreender com comportamentos inesperados.

[Geraldo/Guilherme/Hiran] concordo completamente [Julio] concordo

9 - O sistema desempenha as tarefas apropriadamente.

[Guilherme] concordo completamente [Geraldo/Hiran/Julio] concordo

10 - É rápido de se trabalhar utilizando a ferramenta.

[Guilherme/Hiran] concordo completamente [Geraldo/Julio] concordo

11 - Uma vez que aprendeu a utilizar o sistema, será fácil utiliza-lo na próxima vez.

[Geraldo/Guilherme/Hiran/Julio] concordo completamente

12 - Eu fiquei satisfeito com o sistema.

[Guilherme/Hiran/Julio] concordo completamente [Geraldo] concordo

# Anexo I – Documento de visão

## Anexo A – DOCUMENTO DE VISÃO

**Nome da Aplicação = Gestão de Governança Corporativa de Serviços**

### Histórico de Revisões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
<b>18/02/2013</b>	1.0	Versão inicial	Haroldo Teixeira Filho

### Objetivo

O objetivo deste documento é contextualizar e descrever o problema, informar como o mesmo está sendo abordado atualmente e propor uma solução, definindo o seu escopo e identificando as restrições que a limitam.

Relaciona ainda os envolvidos e usuários, assim como descreve os requisitos de negócio que devem ser atendidos. Dessa forma, este documento fornece insumos para o acordo com o cliente, a estimativa de esforço e o desenvolvimento da solução.

### Visão Geral do Problema

O centro de competência de integração da TIC (CCI) é a área responsável pela disponibilização de soluções para integração de sistemas online na TIC da organização XPTO<sup>13</sup>, utilizando como principais ferramentas de trabalho o barramento corporativo e o repositório de ativos de integração. Esta equipe recebeu em 2011 a responsabilidade de manter e promover a arquitetura orientada a serviços na área de tecnologia de informação da XPTO.

---

<sup>13</sup> Organização descaracterizada

Foi realizado projeto de estruturação da área, nos quais foram definidos métodos para construção de soluções de integração, políticas de governança de serviços, papéis e responsabilidades e ferramentas de apoio. Neste trabalho foram identificadas necessidades de trabalho em conjunto com o centro de provimento de infraestrutura (equipe de INFRA), responsável pela implantação e monitoração das soluções nos ambientes de homologação e produção, e com a equipe de Gerência de Configuração de Software (GCS), visando o correto versionamento e controle de liberações dos produtos gerados pelo CCI.

Uma vez dominados os artifícios técnicos para desenvolvimento de soluções usando o barramento de serviços, o CCI resolveu iniciar a implantação de mecanismos de governança mais refinados, com o objetivo de agregar mais valor as soluções SOA entregues para a organização.

Para tal, optou-se pela implantação de um ciclo iterativo de implantação de governança de serviços, baseado em quatro etapas: planejamento (definição do escopo de cada iteração), definição (especificação e configuração de ferramentas para suportar novas políticas ou processos), execução (liberação para o uso e monitoração) e avaliação (análise de conformidade, lições aprendidas, análise crítica do processo e recomendações de novas ações). Desta maneira, novas políticas e processos de governança podem ser incluídos de maneira gradual, com os benefícios sendo medidos ao término de cada iteração.

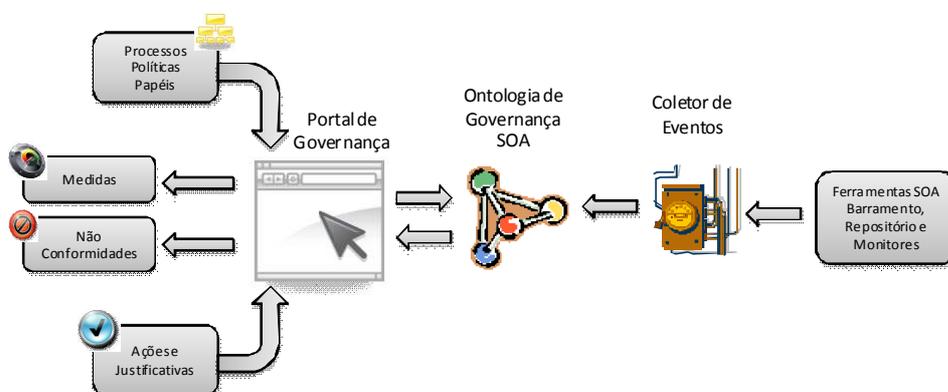
Porém, as ferramentas disponíveis não permitem a criação, gerenciamento e verificação de conformidade de acordo com este modelo. Torna-se necessário um software que permita a gestão dos objetos relacionados a este modelo, facilitando o trabalho das equipes de governança, infraestrutura e GCS.

## Visão Geral da Solução Proposta

A solução proposta visa apoiar o processo de gestão do modelo de governança da XPTO, provendo uma plataforma na qual as informações sobre os elementos que constituem a arquitetura são consolidadas e processos e políticas podem ser definidos e verificados de maneira automatizada.

Como boa parte das políticas envolve convergência de informações relativas ao contexto no qual a TIC opera, abrangendo diversos campos de conhecimento existentes na companhia, optou-se por utilizar uma base de dados descrita sob a forma de uma ontologia, visando representar conhecimento tácito envolvido e facilitar a avaliação de conformidade através do uso de inferências.

Uma visão geral do ambiente proposto é apresentado na figura a seguir.



Um portal de governança operará como ferramenta para o cadastro de processos, políticas, artefatos e métricas a serem consideradas para governança SOA. Este mesmo ambiente permitirá a visualização de relatórios para acompanhar a evolução do modelo de governança. Todos os dados serão armazenados na ontologia. Adicionalmente, o portal permitirá o cadastro das ações

e justificativas a serem executadas caso sejam detectadas não conformidades.

Em paralelo, coletores de eventos obterão informações sobre os ambientes que contém os objetos governados - barramento de serviços, repositório e planilhas de controle.

Os requisitos específicos serão detalhados na próxima seção deste documento.

## **Requisitos de Negócio**

### **Funcionais**

#### **Manter Modelo de Maturidade de Governança**

O analista de governança mantém modelos de maturidade, indicando nome, descrição, níveis e, para cada nível, uma descrição.

#### **Manter modelo de governança**

O analista de governança pode manter um modelo de governança, composto por uma lista de processos, papéis e políticas a serem aplicados. Cada um destes elementos deve ser relacionado a um ou mais níveis do modelo de governança.

#### **Definir ciclo de governança**

O analista de governança define o escopo de um ciclo de governança, informando que novas políticas, papéis ou processos serão implantados, alterados ou descontinuados e informando data de início e término previstos para o ciclo e modelo de governança a ser aplicado. Os elementos a serem incluídos no ciclo de governança devem constar do modelo de governança selecionado.

#### **Manter Processo**

O analista de governança mantém a definição de um processo, informando seu nome, papéis associados, estados possíveis e artefatos que evidenciam sua execução.

### **Manter Papel**

O analista de governança define um papel, relacionando com as competências que demandadas para sua execução.

### **Manter Competência**

O analista mantém a definição de competências, informando uma descrição textual e áreas relacionadas.

### **Manter KPI**

O analista de governança informa uma KPI, que pode ser associada a um processo ou política, e que pode utilizar uma regra para cálculo, utilizando como parâmetro qualquer elemento existente na ontologia.

### **Manter Política**

O analista de governança informa uma política, através da definição de uma regra, utilizando como parâmetros quaisquer elementos da ontologia.

### **Atualizar Estágio do Ciclo de Governança**

O analista de governança atualiza os estágios do ciclo de governança, conforme sua evolução.

### **Obter lista de não conformidades**

O usuário obtém uma lista de não conformidades de políticas e processos, podendo filtrar por período, política ou processo específico e por ciclo de governança.

### **Manter ações e justificativas**

O usuário poderá cadastrar ações ou justificativas para as não conformidades, permitindo definir para as ações título, descrição, prazo e status, podendo ser este em aberto, encerrada, atrasada (gerado automaticamente comparando a data atual com o prazo) ou cancelada; e para as justificativas descrição e responsável.

### **Manter ações e justificativas**

O usuário obtém uma lista de ações e justificativas, podendo filtrar por ciclo, processo, política, período, status e categoria (ação/justificativa).

### **Exibir evolução de não conformidades**

O usuário obtém um quantitativo de conformidade, mostrando a evolução ao longo de diversos ciclos de governança. Podem ser aplicados filtros por período, ciclo de governança, política, processo ou nível de maturidade.

### **Exibir painel de indicadores**

Exibe os indicadores relacionados ao modelo de governança, podendo ser filtrado por período, processo ou KPI.

### **Não Funcionais**

#### **Registo das transações**

Todas as operações executadas serão registradas possibilitando diagnóstico de problemas e auditoria. Também será possível buscar as transações por ID da alteração e por período.

#### **Atributos dos elementos**

Para fins de cadastro, os atributos dos elementos devem ser obtidos dinamicamente através de consulta ao modelo da

ontologia, visando mais flexibilidade e possibilidade de implantação em outras organizações.

### **Carga das Informações**

As informações referentes a processos e incidentes serão carregadas a partir das planilhas de controle utilizadas pelo CCI. As informações referentes a serviços e componentes da arquitetura serão carregadas a partir de interface com as ferramentas de SOA existentes no ambiente.

### **Perfilação**

O sistema contará com perfis configuráveis de usuários, permitindo selecionar que objetos poderão ser mantidos ou consultados por um perfil. Caso um usuário tenha permissão de visualizar apenas um conjunto restrito de elementos, ao consultar relatórios e indicadores, deverá poder visualizar apenas informações relativas ao universo ao qual tem acesso.