CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MELISSA CORDEIRO CAVALCANTI

MODELO DE ROBÔ PARA EXTRAÇÃO DE DADOS NA ÁREA DE INOVAÇÃO DO ESTADO DE SÃO PAULO

Marília, 2016

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA

FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MODELO DE ROBÔ PARA EXTRAÇÃO DE DADOS NA ÁREA DE INOVAÇÃO DO ESTADO DE SÃO PAULO

Trabalho de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação

Orientador:

Prof. Fábio Dacêncio Pereira

Marília, 2016

CAVALCANTI, Melissa Cordeiro

Modelo de Robô para Extração de Dados na área de Inovação do Estado de São Paulo / Melissa Cordeiro Cavalcanti, orientador: Profº. MSc. Dr. Fábio Dacêncio Pereira, SP: [s.n.], 2016.

58 folhas

Monografia (Bacharelado em Ciência da Computação): Centro Universitário Eurípides de Marília.

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
MANTIDO PELA FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Melissa Cordeiro Cavalcanti

MODELO DE ROBÔ PARA EXTRAÇÃO DE DADOS NA ÁREA DE INOVAÇÃO DO ESTADO DE SÃO PAULO

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 9,0 (nove)

Orientador:Fábio Dacêncio Pereira

1º.Examinador:Emerson Alberto Marconato

2º.Examinador:Caio Saraiva Coneglian

Marília, 07 de dezembro de 2016.

À minha família, por acreditar e investir em mim. Mãe, Pai e Avó, seus exemplos, cuidado e dedicação foi que deram a esperança para seguir.

AGRADECIMENTOS

Este trabalho é fruto de um conjunto de esforços realizados por mim e por todos aqueles que contribuíram, de maneira direta ou indireta, para que eu alcançasse esse grande objetivo da minha vida.

Agradeço a Deus, a minha família, minha mãe e meu pai por terem proporcionado a realização dessa graduação, meu professor orientador Fabio Dacêncio Pereira por ter estimulado e fornecido as bases de minha pesquisa, jamais teria chegado tão longe sem o seu apoio. Um agradecimento especial aos meus amigos Valdir Junior, Jordan Saran e Livia Sad pelo apoio e estímulo e pelos momentos de conversas, descontração e brincadeiras.

Resumo

Diversos são os dados encontrados no cenário Web, ainda mais quando acontece o relacionamento entre os atores de inovação. Já que o relacionamento existente governo, empresas e universidades é dificultado por possuírem pontos de vista diferentes e as informações encontradas neste meio podem estar espalhadas e sem um formato específico. Assim, foi implementado um Robô Extrator capaz de retirar informações semiestruturadas da página de Chamadas Públicas Abertas do site do CNPq, este foi construído com o auxílio da biblioteca Jsoup e do Framework Jena. Essas informações extraídas da Web, são relacionadas com a parte governamental, são inseridas na estrutura RDF pré-definida, e podem ser realizadas consultas de acordo com a sintaxe das palavras e por um intervalo de tempo utilizando a linguagem SPARQL. Tendo como intuito traçar um caminho para melhorar a extração e esquematização dos dados semiestruturados que são extraídos e consultados pelo usuário, para que futuramente seja realizada a visualização das informações que possuem valor específico e relacionam os Atores de Inovação do Estado de São Paulo.

Palavras-chave: Cenário Web; Atores de Inovação; Dados Semiestruturados; RDF; Robô Extrator; Sintaxe;

Abstract

There is much data found in the Web scenario, even more when there is a relationship between the inovation actors. The relationship between government, companies and universities is hardened due to their different points of view and informations found in these mediums, that can be scattered and in no specific format. Because of this, a Extractor Robot capable of retrieving semi-structured information from the Chamadas Públicas Abertas page on CNPq's website was implemented, built with aid from the Jsoup library and the Jena Framework. These extracted informations from the Web are related to the governmental part, inserted into a pre-defined RDF structure, and queries can be done in accordance to the word sintax and a time interval using the language SPARQL. With the intent of delineating a path to improve the extraction and schematization of semi-structed data that are extracted and consulted by the user, so that in the future visualizations of the information with a specific value and related to the Inovation Actors in the state of São Paulo can be fulfilled.

Keywords: Web Scenario; Inovation Actors; Semi-structured data; RDF; Extractor Robot; Sintax;

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

API	*Application Programming Interface*
CIT	Centro de Inovação Tecnológica
CITec-Marília	Centro de Inovação Tecnológica de Marília
CITJUN	Incubadora Tecnológica de Jundiaí
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CSS	*Cascading Style Sheets*
CSV	*Comma Separated Values*
DAML	*DARPA Agent Markup Language*
DOM	*Document Object Model*
EBT	Empresa de Base Tecnológica
FPTS	Fundação Parque Tecnológico de Santos
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
JSON	*JavaScript Object Notation*
OIL	*Ontology Interchange Language*
OWL	*Ontology Web Language*
PICS	*Platform for Internet Content Selection*
PTS	Parque Tecnológico de Sorocaba
RDF	*Resource Description Framework*
RPCITec	Rede Paulista de Centros de Inovação Tecnológica
RPITec	Rede Paulista de Incubadoras de Empresas de Base Tecnológica
RPNIT	Rede Paulista de Núcleos de Inovação Tecnológica
SPAI	Sistema Paulista de Ambientes de Inovação
SPARQL	*Simple Protocol and RDF Query Language*
SPTEC	Sistema Paulista de Parques Tecnológicos
TSV	*Tab Separated Value*
URI	*Uniform Resource Identifier*
XML	*Extensible Markup Language*

Sumário

Introdução

O governo do estado de São Paulo busca incentivar o cenário de inovação estadual, com programas e leis que foram construídos e estabelecidos com a intenção de melhorar o relacionamento entre governo, empresas e universidades, já que para a cultura de inovação se desenvolver é necessário um bom envolvimento entre estes atores de inovação.

As leis e acordos criados e estabelecidos pelo governo estadual, a partir do início anos 2000, têm o objetivo de incentivar a relação e integração entre empresas, instituições governamentais e educacionais com o intuito melhorar e ampliar as informações e os produtos existentes que são relacionados ao conceito de inovação tecnológica. Assim, são capazes de estabelecer procedimentos, entre esses agentes, que apoiam projetos nos ambientes de desenvolvimento de inovação e tecnologia.

Os parques tecnológicos, as incubadoras de empresas de base tecnológica e os centros de inovação tecnológica são empreendimentos que procuram incentivar a relação entre os atores de inovação por meio de incentivos. Estes são capazes de disponibilizar suportes, de diversas formas, para que haja o desenvolvimento de projetos embasados nos conhecimentos tecnológicos e inovação e apoiando o desenvolvimento de empresas de tecnologia a curto e longo prazo. O núcleo de inovação tecnológica tem como finalidade gerir as políticas de inovação das instituições científicas e tecnológicas do estado de São Paulo.

O fato da Web ter crescido e se desenvolvido de maneira não centralizada, mostra que as informações existentes neste meio necessitam de técnicas que são capazes de classificar, organizar e estruturar os dados que são buscados e extraídos deste ambiente. A partir dos estudos realizados, ontologias são capazes de definir o contexto e os vocabulários em um domínio com múltiplos agentes e podem servir como base para comunicação entre eles na extração de informações.

Os robôs de busca e extração de dados são agentes computacionais capazes de percorrer um determinado cenário Web que é restringido a ele em sua implementação, procurando e decidindo as informações que possuem maior grau de importância e serão armazenadas de acordo com os padrões que são estabelecidos em seu desenvolvimento. Os agentes de buscas podem realizar as atividades destinadas a eles, e ao manipular as informações retiradas do meio são aptos a relacionar localidade, período de tempo e palavras, por exemplo.

A Fusão de Dados permite que aconteça a padronização e relação de informações que não possuem ligação direta e devem ser guardadas e usadas posteriormente. Essa padronização também pode ser realizada com auxílio de uma ontologia e é responsável por agregar valor aos dados que são recuperados, já que passa a existir um relacionamento entre atores na busca de informações.

A extração das informações semiestruturadas de uma página Web com auxílio do Jsoup[1], demonstrou que esta biblioteca, disponível para Java, é capaz de contornar essa limitação por não existir um padrão de formato pré-definido para os dados que são encontrados. Assim, como, a utilização do Jena[2] foi capaz de mostrar que esses dados extraídos podem ser reservados na estrutura RDF (*Resource Description Framework*) construída e manipulados de acordo com o que tiver importância para o usuário, com o uso de aplicações que podem ser disponibilizadas pelo próprio *framework*, como é o caso do SPARQL que foi usado para realizar as consultas de acordo com a sintaxe das palavras e por um período de tempo.

Com isso, o Robô Extrator construído tem como fonte informacional dados retirados do site do CNPq e demonstra a direção de um caminho para melhorar a extração e esquematização dos dados semiestruturados e não estruturados. De acordo com os resultados alcançados, também pode ser observado que, para trabalhos futuros, uma busca realizada pelo usuário pode ser objetivada no momento que for inserida uma ontologia específica e a fusão de dados, para construir o inter-relacionamento entre as informações ligadas aos Atores de Inovação, retiradas do cenário que é adotado como fonte informacional.

[1] Biblioteca Java, usada para extração de página da web.
[2] Framework Java gratuito para construção de Web Semântica e aplicações Linked Data.

Objetivos Gerais e Específicos

Este trabalho tem como objetivo projetar e desenvolver um robô de extração de dados, capaz de consultar dados semiestruturados, que são encontrados na página do CNPq e estruturá-los em um formato RDF. Essa estrutura é construída e preenchida com auxílio de aplicações disponibilizadas pelo *framework* Jena, sendo apresentado para o usuário apenas as informações encontradas de acordo com uma determinada consulta.

Objetivos Específicos

Para cumprimento dos objetivos gerais, os seguintes objetivos específicos foram ser realizados:

- Estudo do Cenário de Inovação do Estado de São Paulo
- Estudo do funcionamento do Jsoup e do SPARQL
- Estudo sobre estruturação e construção de RDF com utilização do Jena
- Estudo sobre Extração de informações com Jsoup
- Estudo sobre os padrões de consultas que são realizados pelo SPARQL
- Implementação de Robôs de Busca com utilização do Jsoup e de aplicações do Jena
- Análise de resultados apresentados

Metodologia

Para desenvolvimento do projeto, os seguintes passos foram realizados:

1. Estudo do campo de inovação tecnológica no estado de São Paulo, sendo observadas as atividades realizadas para melhorar o entrosamento entre os atores de inovação e como esse relacionamento funciona com auxílio de alguns programas realizados pelo Governo Estadual.

2. Estudo bibliográficos com os trabalhos correlatos de acordo com as semelhanças e diferenças, sendo observada a Arquitetura de Referência proposta e o que foi desenvolvido neste projeto.

3. Estudo da fusão de dados e informações: para entender o funcionamento da Fusão de dados e informações com o uso de Ontologias existentes.

4. Estudo de algumas técnicas para criação e desenvolvimento de robôs de buscas tendo como foco aplicabilidade no cenário Web, das técnicas para a extração de dados com uso da biblioteca Jsoup e para construção da estrutura RDF implementada com Jena, para guardar e estruturar as informações retiradas do ambiente.

5. Verificação do funcionamento do SPARQL para realizar as consultas usando essa linguagem e observando os padrões necessários para que seja retornado corretamente o que foi solicitado.

6. Implementação da extração de dados, realizando consultas específicas por palavras e por um intervalo de tempo, em cima das informações que foram salvas na estrutura RDF, com auxílio do SPARQL e análise dos resultados encontrados pelo robô de busca desenvolvido que são apresentados em uma interface simples.

Organização do Trabalho

Este projeto pode ser divido no Estudo do Cenário de Inovação do Estado de São Paulo, no Estudo dos conceitos relacionados às Tecnologias e Ferramentas para Extração e na Implementação e Verificação.

- **Estudo do Cenário de Inovação do Estado de São Paulo:** Consiste em realizar o estudo sobre a atual situação do ambiente de inovação encontrado no Estado de São Paulo, bem como a verificação das leis e ações que foram e estão sendo implantadas pelo Governo do Estado para melhorar o entrosamento entre os atores de inovação.
- **Estudo dos conceitos relacionados às Tecnologias e Ferramentas para Extração:** Fundamenta no estudo do funcionamento e da utilização das aplicações, ferramentas e bibliotecas usadas para a implementação do robô e no estudo dos conceitos que estão envolvidos com a construção de um robô de extração de informações.
- **Implementação:** Para construção do Robô Extrator o Java foi escolhido como linguagem de programação e, juntamente, com ele foi utilizado a biblioteca Jsoup e aplicações disponibilizadas pelo *framework* Jena para implementar a construção do RDF e realizar as consultas necessárias.
- **Verificação:** Foi feito uma série de testes em cima das consultas, no intuito de verificar o resultado obtido e se estes batem com as informações encontradas na página de chamadas abertas do CNPq.

1. Cenário de Inovação Tecnológica no Estado de São Paulo

É apresentado, neste capítulo, a ambientação do cenário de inovação tecnológica no estado de São Paulo. O objetivo desse capítulo é apresentar alguns ambientes e projetos que têm como propósito auxiliar o relacionamento entre os agentes de inovação, que também fazem parte de todo processo para a busca de informações.

1.1 Ambiente de Inovação Paulista

O campo de desenvolvimento tecnológico no Brasil tem evoluído pela modificação das relações existentes entre pesquisadores, indústrias e meio acadêmico, estes conhecidos como os atores de inovação. A partir do início dos anos 2000, os governos federais e estaduais brasileiros começaram a desenvolver maneiras para que a evolução da ciência, pesquisa e desenvolvimento tecnológico ocorresse de forma mais eficaz, e com a existência do relacionamento entre os atores.

Políticas públicas, programas e órgãos governamentais, em esfera nacional, estão sendo criados e reestruturados com a finalidade de unir e articular as iniciativas privadas, universidades e as diferentes esferas existentes no governo, seja em questão nacional, estatal ou municipal. Pereira (2016, p.4-5) destaca como alguns desses programas:

- Marco da Ciência Tecnologia e Inovação: regulamenta a Emenda Constitucional 85, definindo parcerias de longo prazo entre os setores público e privado, dá maior flexibilidade de atuação às Instituições Científicas, Tecnológicas e de Inovação (ICTs) e respectivas entidades de apoio.
- Programa Nacional de Apoio às Incubadoras de Empresas e aos Parques Tecnológicos (PNI) do Ministério da Ciência, Tecnologia e Inovação(MCTI).
- Financiadora de Estudos e Projetos (FINEP), empresa pública criada em julho de 1967 e vinculada ao MCTI. Tem como missão "Promover o desenvolvimento econômico e social do Brasil por meio do fomento público à Ciência, Tecnologia e Inovação em empresas, universidades, institutos tecnológicos e outras instituições públicas ou privadas."
- Política Industrial, Tecnológica e de Comércio Exterior – PITCE - A PITCE379 foi instituída com o objetivo de aumentar a competitividade das empresas brasileiras, mediante elevação dos níveis de eficiência e produtividade, fomento à capacidade inovadora e estímulo às exportações.
- Lei de Inovação e Lei do Bem: (a) incentiva parcerias em P&D entre universidades, instituições de pesquisa e empresas; (b) regula a transferência de tecnologia e a criação de incubadoras;(c) permite compartilhar equipamentos, infraestrutura e pessoal em atividades de desenvolvimento de novas tecnologias; e (d) estabelece subsídios e recursos para tais atividades.
- Conselho Nacional e Agência Brasileira de Desenvolvimento Industrial: Para aprimorar a coordenação institucional e, principalmente, incentivar a inovação e os gastos das empresas privadas em P&D, foram estabelecidos dois novos

> órgãos: o Conselho Nacional de Desenvolvimento Industrial (CNDI) e a Agência Brasileira de Desenvolvimento Industrial (ABDI).

Segundo Bruno Rondani[3] (apud MENDONÇA, 2012, p. 70) de acordo com a visão instituída até aquele momento "a ciência só é boa se for para gerar conhecimento. [. . .] O ideal é que haja todo um sistema de financiamento e cooperação entre os atores responsáveis pela inovação tecnológica [. . .]". Assim, o governo do estado de São Paulo desenvolveu e instalou programas e legislações como forma de incentivo ao relacionamento acadêmico, industrial e governamental, com a intenção de melhorar o desenvolvimento no campo de inovação estadual.

De acordo com a Secretaria de Desenvolvimento Econômico, Ciência, Tecnologia e Inovação do estado de São Paulo, o Centro de Inovação Tecnológica (CIT) é um local criado com o intuito de realizar o estímulo ao crescimento e à competitividade das micro e pequenas empresas com auxílio do avanço tecnológico que promove a interação entre empreendedores e pesquisadores; é capaz de oferecer mecanismos e serviços de suporte ao processo de inovação das empresas e teve como apoio à sua instalação o lançamento da Rede Paulista de Centros de Inovação Tecnológica (RPCITec).

Os objetivos da RPCITec são realizar ações estimulantes à cultura de inovação e aos centros de inovação tecnológica integrante dela a realização de pesquisa, desenvolvimento e engenharia de produtos e/ou processos; facilitar e estimular a consolidação de parcerias entre esses centros de inovação tecnológica com empresas e organizações que participam da área produtiva; e realizar todo o apoio necessário para o desenvolvimento, seja por meio de capacitações, treinamentos e eventos, como disponibilização de serviços tecnológicos.

A Rede Paulista de Incubadoras de Empresas de Base Tecnológica (RPITec), instituída pelo decreto Nº 56.424 de 23 de Novembro de 2010, foi criada com a intenção de apoiar, fortalecer e estimular a instalação de empresas inovadoras em desenvolvimento de produtos e/ou processos no Estado, já que incubadoras de empresas de base tecnológica, conhecidas como EBTs, são capazes de oferecer espaço físico por determinado período de tempo para empresas da área tecnológica que estão iniciando, oferecendo suporte gerencial e tecnológico; o que gera a interação entre essas empresas e, consequentemente, realiza a troca de informações e a difusão do conhecimento.

Em 25 de março de 2014 foi instituído o decreto Nº 60.286 que regulamenta e institui o Sistema Paulista de Ambientes de Inovação (SPAI). Este, por sua vez, abrange o

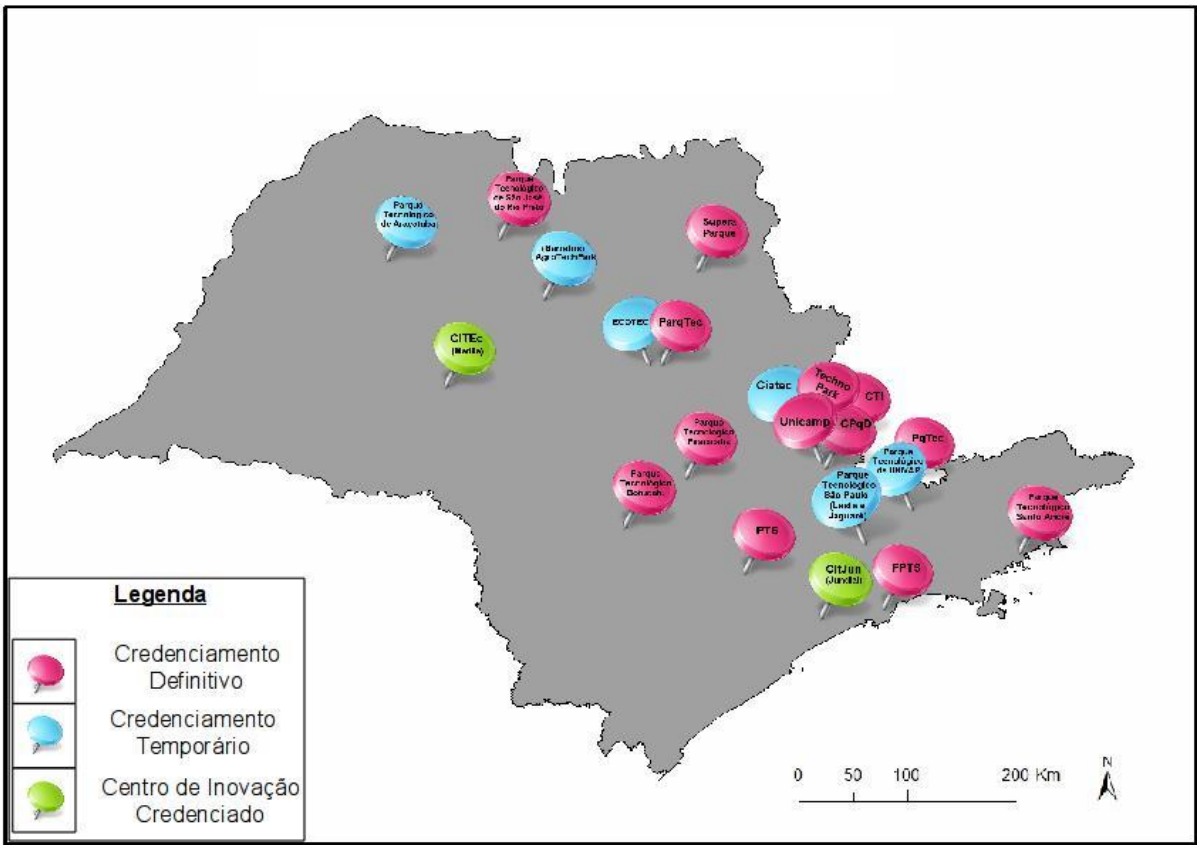
[3] Diretor do Centro de Pesquisa e Inovação Sueco-Brasileiro (Cisb) em 2012

Sistema Paulista de Parques Tecnológicos (SPTec), a RPITec, a RPCITec e a Rede Paulista de Núcleos de Inovação Tecnológica (RPNIT). Este decreto considera que parques tecnológicos, incubadora de empresas de base tecnológica, centro de inovação tecnológica e núcleo de inovação tecnológica devem apoiar projetos e/ou processos tecnológicos nos ambientes de inovação e tecnologia, incentivando a relação entre os atores de inovação.

O Sistema Paulista de Parques Tecnológicos tem como objetivo estimular o surgimento, desenvolvimento, competitividade e aumento da produtividade de empresas com atividades baseadas no conhecimento, na tecnologia e na inovação, incentivando a interação entre instituições de pesquisa, meio acadêmico, capital de oportunidade e investidores; realizar o desenvolvimento de São Paulo atraindo investimentos para as atividades baseadas no conhecimento e na inovação tecnológica.

Na Figura 1 é mostrada as disposições e localizações dos Parques Tecnológicos com Credenciamento Definitivo ou Temporário, assim como, os Centros de Inovação Tecnológicos Credenciados que participam do SPAI.

Figura 1 - Mapa do Estado de São Paulo com Parques Tecnológicos e Centros de Inovação do SPAI



Fonte: Autoria Própria

1.1.1 Parques Tecnológicos Credenciados Definitivamente

Os Parques Tecnológicos que participam do SPAI recebem inicialmente o credenciamento temporário, o qual permite a atuação destes no cenário tecnológico até que seja aprovado, pela organização regulamentadora, o credenciamento definitivo. Neste item, são apresentados alguns dos Parques que estão demonstrados na Figura 1.

O Parque Tecnológico de São José dos Campos possui Centros Empresariais que abrigam aproximadamente 60 empresas e oferece para essas empresas espaço físico e infraestrutura básica capazes de abrigar suas instalações e seu pessoal. Criado em 2009 por iniciativa da Prefeitura de São José dos Campos, foi o primeiro parque a ser credenciado definitivamente pelo SPTec no ano de 2010. (Disponível em: http://www.pqtec.org.br/conheca-o-parque/historico.php. Acesso em: Mar 10, 2016)

Localizado em São Carlos, o ParqTec tem como finalidade a promoção do desenvolvimento regional com otimização do custo da transação realizada por inovação tecnológica e mercado. Contribui de maneira significativa na construção de uma Região de Inovação constituída por universidades públicas e privadas, centros de pesquisas, órgãos de governo e por um conjunto de mais de 180 EBT's. (Disponível em: http://parqtec.com.br/quem-somos/instituicao/. Acesso em: Mar 10, 2016)

Responsável por atrair e reter empresas tecnológicas, com destaque para os setores de Saúde, Biotecnologia, Tecnologia da Informação e Bioenergia, o Parque Tecnológico localizado em Ribeirão Preto surgiu de uma parceria entre USP, Prefeitura Municipal e Secretaria de Desenvolvimento Econômico, Ciência, Tecnologia e Inovação do Estado de São Paulo. (Disponível em: http://superaparque.com.br/conheca-o-parque/. Acesso em: Mar 10, 2016)

Inserido no SPTec, o Parque Tecnológico de Piracicaba surgiu das diferentes visões de pessoas determinadas nos governos Estaduais e Municipais. Seus objetivos são promover a informação tecnológica, estimular que centros de pesquisa, universidades e empresas cooperem entre si e dar suporte ao desenvolvimento de atividades empresariais. (Disponível em: http://www.parquetecnologico.piracicaba.sp.gov.br/index.php/o-parque. Acessado em: Mar 10, 2016)

O Parque Tecnológico de Sorocaba (PTS) criado para atrair e acomodar empresas com base tecnológicas, instituições de ensino e pesquisa, assim como empresas de consultoria ou organizações, públicas e/ou privadas, que possam oferecer serviços de apoio técnico e de mercado, com o intuito de facilitar acesso ao conhecimento e ao mercado, pela aproximação

com possíveis desenvolvimentos e inovação tecnológica. (Disponível em: http://www.empts.com.br/parque. Acesso em: Mar 10, 2016)

Situado na cidade de Santos, o FPTS realiza a promoção da ciência e tecnologia e age aproximando os centros de conhecimento e o setor produtivo, oferece oportunidade para que as empresas do Estado possam transformar pesquisa em produto. Tem como intuito propagar a cultura da inovação e empreendedorismo com a finalidade de realizar o desenvolvimento sustentável na cidade e na região metropolitana da Baixada. (Disponível em: http://www.fpts.org.br/sobre.asp. Acesso em: Mar 10, 2016)

1.1.2 Centros de Inovação

Os Centros de Inovação possuem como alguns objetivos a promoção da competitividade local e regional, sediar incubadoras de empresas de base tecnológica e laboratórios específicos de acordo com a demanda regional. A Figura 1 mostra os dois Centros de Inovação Tecnológico Credenciados que fazem parte do Sistema Paulista de Ambientes de Inovação e são apresentados abaixo.

Localizado na cidade de Jundiaí, o CITJUN, tem como propósito primário facilitar o desenvolvimento tecnológico e atua agregando e incentivando as ações governamentais, acadêmicas e empresariais da região. Desenvolvido pela Prefeitura de Jundiaí, governo de São Paulo, pelo Sincomércio que é a gestora do Centro de Inovação e pela Companhia de Informática de Jundiaí. (Disponível em: http://www.incubadorajundiai.com.br/. Acesso em: Mar 10, 2016)

O Centro de Inovação Tecnológica de Marília (CITec-Marília) foi credenciado pelo governo do Estado em dezembro de 2015 na RPCITec. Esta tem como finalidade promover o fortalecimento e estimular os processos locais e regionais em benefício do desenvolvimento e da competitividade das empresas da região, também oferece um espaço adequado para a pesquisa, desenvolvimento e inovação (P&D&I) de empresas que tenham perfil inovador. (Disponível em: http://www.inovamarilia.com.br/citec-marilia/. Acesso em: Mar 10, 2016)

Portanto, percebe-se que os programas, leis e atividades políticas de modo geral, realizadas pelo governo estadual com a intenção de melhorar o seu cenário de inovação tenológica, tiveram um retorno positivo e facilitam o relacionamento entre os atores de inovação. Entretanto, mesmo com os diversos tipos de incentivos, pelo fato de empresas, centros acadêmicos e agentes governamentais possuírem objetivos muito distintos, a

existência da relação entre eles ainda é de grande dificuldade. Assim, faz-se necessária a continuidade de implementações políticas que desenvolvam esses projetos e ações, com a finalidade de melhorar a interação entre os atores e o desenvolvimento da inovação tecnológica no estado.

2. Conceitos, Tecnologias e Ferramentas

Para realização da Extração de Dados no meio Web, por meio de um robô de extração de dados, são utilizados softwares, aplicações e codificações específicas para extração de informações. A seguir é apresentado informações sobre o cenário, métodos para a estruturação das informações acessadas e biblioteca existente para a busca de dados encontrados em Páginas de sites.

2.1 Web e Web Semântica

O ambiente Web se desenvolveu de maneira difusa, tendo como prioridade inicial para seu desenvolvimento a construção da rede, fazer com que esta fosse acessível e capaz de ser comercializada. Por conta da descentralização gerada durante toda a sua construção e evolução, surge a necessidade de encontrar e interpretar conteúdos específicos para a recuperação das informações.

No resultado de uma busca realizada por um usuário pode estar contido inúmeras informações, sendo estas, relevantes ou não para ele, por conta da vasta quantidade de informações que estão disponíveis neste meio. Esse acontecimento permite que o próprio usuário tenha o poder de decidir e verificar as informações resultantes que tem importância real para serem usadas por ele.

> O principal propósito da Web Semântica é de atribuir significado ou sentido a qualquer conteúdo publicado na internet, através da utilização de metadados, de maneira a tornar as informações na Web interpretáveis por computadores, alcançando assim resultados mais rápidos, inteligentes, eficientes e precisos no compartilhamento de informações. (PRYBECS; JÚNIOR; MENDES, 2013, p. 6)

Assim, percebe-se que a Web Semântica aparece como um auxílio para que o desenvolvimento de aplicações que sejam capazes de dar como resposta ao usuário informações realmente relevantes para sua busca e que melhor se relacionam com o que é explicitado, e não toda a vasta quantidade de dados que são encontrados de maneira descentralizada que podem não ter valor associado ao que foi especificado.

Segundo Bräscher (2007), a Web Semântica é uma plataforma acessível e universal capaz de permitir que os dados possam ser compartilhados e processados tanto por ferramentas automáticas como por pessoas, são agentes capazes de realizar busca, filtragem e preparação de dados encontrados para o usuário.

De acordo com Dias e Santos (2003), a proposta da Web Semântica não é uma separação da Web atual, mas sim uma extensão dela, baseada em ontologias capazes de descrever os relacionamentos entre os objetos e conter suas informações semânticas para acontecer automatização do processamento pelas máquinas, que não acontecem no momento.

Coneglian (2014, p. 35-36) fala que uma das maneiras que a Web Semântica pode ser realizada é fazendo uma divisão por camadas para que ela seja aplicada, sendo estas:

- URI (*Uniform Resource Identifier* – Identificador de Recursos Uniforme): conjunto de caracteres para a identificação de um recurso (W3C, 2014b);
- Unicode: define um conjunto e padrão universal de codificação (UNICODE, 2008);
- XML (*Extensible Markup Language* – Linguagem de Marcação Extensível): é um sistema de representação de informação estruturada (W3C, 2014c);
- *Namespace*: um conjunto de nomes, identificada por uma referência URI.
- XML *Schema*: expressam os vocabulários compartilhados e permitem que as máquinas vejam as regras feitas pelas pessoas (W3C, 2014d);
- RDF M&S: um modelo para intercâmbio de dados na web, e tem características que facilitam a fusão de dados (W3C, 2014e);
- RDF *Schema*: um vocabulário para fazer a modelagem de dados de RDF (W3C, 2014f);
- *Ontology*: será tratado com mais clareza ainda neste capítulo;
- *Rules*: nela é feita a conversão das informações que estão dentro de um documento para outro, criando regras de inferência (PRADO, 2004).
- *Logic*: tem a intenção de transformar o documento em uma linguagem lógica, fazendo inferências e funções, para que duas aplicações de RDF sejam conectadas.
- *Proof*: pode-se depois de passar por várias camadas, fazer uma prova deste documento, ou seja, pode-se provar hipóteses a partir das informações.
- *Sig*: assinatura, para verificar a autonomia do documento.
- *Trust*: tendo a assinatura do documento, pode-se saber a confiança nesta informação.

É observado que para realizar buscas mais eficazes e capazes de conseguir resultados com grande valor de pertinência, podem ser utilizadas ontologias quem organizam os dados da Web de forma que a base de dados seja persistida com dados sobre a relação feita entre os dados extraídos de Fontes Informacionais que não se relacionam, baseando-se na interpretação das respostas realizada por um aplicativo que é usado.

2.2 Ontologias

O conceito de ontologias não existe apenas no campo de Tecnologia da Informação, na realidade, um dos cenários pioneiros na classificação de ontologia foi a Filosofia, que usou desse termo para explicar o ser. Baseado nos apontamentos do curso de Formação de Gestores do Conhecimento da UFBA (2007), Platão foi responsável pelo

primeiro modelo de representação do conhecimento baseado nas questões que eram conhecidas até aquele momento sobre os seres vivos.

Para a Computação o interesse na ontologia teve o intuito de garantir o conhecimento sobre informações relacionadas ao cenário que é usado pelo mecanismo de representação escolhido por um usuário. É observado que a Inteligência Artificial usa desse conceito para realizar descrição de domínios conhecidos e pode servir de auxílio para reuso e compartilhamento das informações que são utilizadas tanto por usuários como por máquinas.

De acordo com Gruber (apud OLIVEIRA e WERNECK, 2003, p. 2) “ontologia é uma especificação explícita de uma conceituação. A conceituação é a organização do conhecimento em forma de entidades e a especificação é a representação dessa conceituação em uma forma concreta”.

Ontologias podem ser classificadas e utilizadas de diversas maneiras, tanto como por nível de generalização como por categorias ou tipos de utilização, dependendo então de como a informação deve ser observada.

Conforme a Formação de Gestores do Conhecimento da UFBA (2007), a classificação de ontologias desenvolvida por Guarino foi construída de acordo com o nível de generalização, podendo elas serem classificadas como: genérica que descrevem dados gerais como objetos, funções, eventos, tempo, etc; tarefas que descreve um vocabulário de termos que tem relação com atividades, independente do domínio relacionado; domínio é capaz de especificar um vocábulo pertencente ao domínio desejado; aplicação descreve informações necessárias para uma aplicação que dependem tanto de um domínio em específico quanto de uma atividade pertencente a este domínio.

Outra classificação que pode ser encontrada foi criada por Uschold, construída tendo como base o tipo de conhecimento pode ser dividida em ontologias de representação, domínio e tarefas, estes definem fundamentos que embasam a representação do conhecimento, domínios específicos e conceituam a resolução de problemas que não dependem do domínio o qual aconteçam, respectivamente. Neste tipo de ontologia, elas também podem ser classificadas quanto ao grau de formalidade que podem ser: altamente informal, é expressa de maneira livre em linguagem natural; estruturada informal, encontrada em linguagem natural mas expressa de maneira restrita; semiformal, definida formalmente e é expressada em uma linguagem artificial; e rigorosamente formal, é expressa por meio de semântica formal, teoremas e provas.

2.2.1 OWL

Desenvolvida pelo W3C a linguagem de Web Semântica conhecida como *Web Ontology Language* (OWL) tem como intuito resolver as limitações encontradas com RDF e RDF *Schema*, desenvolvida baseada nessas duas linguagens e DAML+OIL.

> O DAML+OIL (DARPA Agent Markup Language – Ontology Interchange Language) é uma linguagem baseada no XML, desenhada para possuir muito mais capacidade que este na descrição de objetos e no seu relacionamento; para expressar semântica e criar um alto grau de interoperabilidade entre sites Web. (SOUZA; ALVARENGA, 2004, p.137)

Por ter sido baseada em linguagens que foram construídas com base no XML, também possui como base e tem como finalidade atender aos requisitos da Web Semântica disponibilizando algumas características com melhor descrição quando acontece o relacionamento e as definições dos e entre os recursos. Pelo fato de oferecer a criação de um vocabulário adicional para descrição de propriedades e classes possui a expressividade necessária para representar ontologias mais complexas.

De acordo com Moraes (2007), a linguagem OWL consegue suprir as restrições existentes para RDF e RDF *Schema* como: método usado para indicar que os valores de determinada classe são instâncias de uma ou mais classes gera limitação desses valores que podem ser aplicados a uma certa propriedade; a não identificação de classes que tem uma ligação com uma mesma subclasse sejam distintas; não conseguir criar classes a partir de outras usando operações booleanas como intersecção, união e complemento; não oferecer suporte para realizar a definição da quantidade de valores que uma propriedade pode ter; e não conseguir rotular as propriedades como transitivas, únicas ou inversa de outra, fazendo com que não consiga ser aplicada a dedução a partir dos indícios sobre as classes de acordo com suas propriedades.

Para inteirar outras restrições existentes, a linguagem OWL possui três sublinguagens que foram construídas com a derivação da sua antecessora, são essas:

- A OWL-Lite é a versão mais simplificada e tem seu propósito na descrição de restrições e da hierarquia de classes simples, esta é mais simples de implementar e consequentemente possui melhor desempenho mas tem pouca expressividade da linguagem.

- OWL-DL baseia-se em lógica descritiva o que adiciona a possibilidade de raciocínio automatizada, impondo restrições quanto ao uso dos recursos e melhorando a expressividade da linguagem.
- OWL-Full que para a utilização desta, não é possível realizar deduções em uma ontologia; essa linguagem também não impõe restrição sintática e garante que qualquer documento RDF que seja válido é um documento OWL-Full válido.

2.3 Fusão de Dados

Na fusão de informação os dados encontrados em um determinado cenário são correlacionados e unidos com auxílio de uma ontologia específica, assim podem ser persistidos de maneira padronizada, fazendo com que o usuário final, seja ele máquina ou não, possa utilizar da informação conforme desejado.

Segundo Botega (apud PEREIRA, 2016, p. 19) fusão de dados e informações é a rotina de transformação de dados e informações com a finalidade de produzir estimativas e predições de estados de entidades, tendo como o objetivo a maximização do valor da informação que é adquirida e o estímulo da consciência situacional de analistas em relação a um ambiente desejado.

Pelo fato dos dados serem encontrados de maneira descentralizada e sem uma padronização pré-definida na Web a fusão de informação nesse cenário pode utilizar de Agentes de Extração de dados juntamente com uma ontologia para realizar a extração de informações neste meio, conseguindo unir dados que tenham real valor para o usuário final.

A informação extraída com o uso da união dos Agentes de extração, que no projeto geral são o Governo, Instituições Educacionais e Empresas, e da ontologia pode ser verificada por um algoritmo que tem como funcionalidade, observar se a informação que foi retirada do cenário Web está de acordo com o contexto pedido pela ontologia usada e se será útil para quem for usá-la, assim sendo persistida.

Junior (2008) fala que a fusão de dados é a capacidade que os sensores computacionais possuem para juntar os dados que foram coletados, consequentemente reduzindo a quantidade de informações e o tamanho destas mensagens que trafegam pela Web.

2.4 JSOUP

Conforme Hedley (2016), Jsoup é uma biblioteca java usada para trabalhar com HTML e fornece uma API que usa do CSS, do *Document Object Model* (DOM) e de métodos *jQuery-like* para trabalhar com HTML para realizar a extração e manipulação de dados. Este implementa a especificação para Html5 (WHATWG HTML5) e é capaz de analisar o HTML para o mesmo DOM como fazem os navegadores mais novos e criar uma árvore construída com uma análise sensata.

Essa biblioteca possibilita algumas funcionalidades como: apurar e analisar o HTML de uma URL, um arquivo ou *string*; manipular elementos, atributos e textos HTML; limpar o conteúdo enviado pelo usuário de encontro a uma lista vazia segura, com o intuito de evitar ataques XSS que é uma vulnerabilidade causada por falha na validação de parâmetros de entrada de um usuário e resposta do servidor na aplicação.

2.5 RDF

Segundo Ferreira e Santos (2013), na década de 1990 foi criado um grupo de trabalho, pelo W3C, intitulado como *Resource Description Framework* (RDF) que buscava discutir uma estrutura de recursos que atingisse as necessidades de diferentes comunidades de descrição que se interessassem, pois percebeu-se que apenas a classificação e a descrição do conteúdo de páginas da Web realizado pelo padrão *Platform for Internet Content Selection* (PICS) era insuficiente, contendo limitações nas especificações.

De acordo com a especificação do W3C (1999), o RDF tem como fundamentação o processamento de metadados que fornece interoperabilidade entre as aplicações que trocam informações e podem ser compreendidas por máquinas, realça facilidades para realizar o processamento automatizado de recursos na Web e realiza o uso do padrão XML para especificação da semântica dos dados; realizando, assim, apenas a representação dos metadados sobre os recursos.

A atualização de 2004 pelo W3C trouxe a possibilidade de descrição dos recursos encontrados na Web, representa os metadados, que são conjunto de atributos e informações sobre os dados referidos no *World Wide Web*, também pode ser usado para representar todas as informações ou objetos que podem ser identificados neste meio, mesmo quando esses dados não podem ser recuperados.

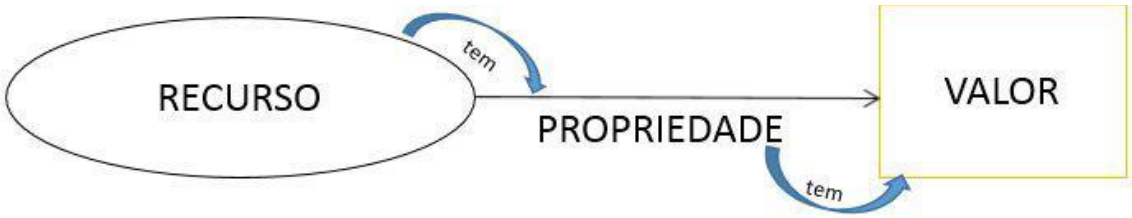
> O RDF é destinado para aquelas situações que as informações precisam ser processadas por aplicativos, e que não são apenas exibidas para pessoas. RDF fornece uma estrutura comum para que essas informações sejam expressas e possam realizar intercâmbios entre aplicações sem perder significado. [. . .] A capacidade de trocar informações entre aplicações diferentes mostra que esta informação pode estar disponível para finalidades diferentes da qual foi gerada inicialmente. (W3C, 2004)

Esta ferramenta é uma estrutura para expressar recursos, estes sendo documentos, pessoas, objetos físicos, conceitos abstratos e outras informações encontradas na Web. Pode ser usado para publicar e interligar dados neste cenário, o que permite que uma pessoa ou processo automatizado possa seguir essas ligações e agregar os dados sobre estes recursos. O RDF é capaz de realizar ligação dos dados que fazem parte de uma organização, e permite consultas cruzadas deste conjunto de dados com utilização do SPARQL; enriquece o *dataset* por meio de interligação com conjunto de dados que possuem recursos vinculados.

A sintaxe deste modelo de dados pode ser realizada por declarações, conhecidas como Triplas compostas por propriedade, recurso e valor que permitem declarações sobre recursos; conforme é mostrado na Figura 2, essas declarações expressam a relação entre dois recursos, um sujeito e um objeto.

Segundo Santarem Segundo (2014) o recurso, ou sujeito, é todo elemento que tenha identidade, pode ser serviço, imagem e outros, nem todo ele pode ser recuperável já que pode ser identificado por uma URI e é considerado o mapeamento conceitual para um conjunto de entidades ou uma única, e a propriedade, por sua vez, possui um respectivo valor (objeto) e caracteriza um recurso, um único recurso pode ter mais de uma propriedade.

Figura 2 - Modelo de estrutura das Triplas



Fonte: Autoria Própria

A Figura 2 apresenta a estrutura de triplas usada pelo RDF, para ela considera-se que um recurso (sujeito) tem um valor ligado a ele por meio de uma propriedade. Na propriedade de conexão entre sujeito e valor é utilizado um '*link*' de ligação, este é capaz de conectar de traçar um caminho entre os elementos.

Dias e Santos (2003) falam que a função exercida pelo Esquema RDF ou RDFS é "permitir a criação de classes de tipos de recursos e propriedades, descrições dessas classes, combinações possíveis de classes, propriedades e valores e restrições entre relacionamentos, definindo assim esquemas que podem ser utilizados em conjunto com vocabulários descritivos".

Para ser utilizada a API conhecida como Jena necessita que o desenvolvedor tenha familiaridade tanto com a linguagem de programação Java como com XML. Possui métodos para ler e escrever RDF e XML, e é capaz de armazenar um RDF em um arquivo e carregá-lo em outro.

2.6 SPARQL

De acordo com Elias e Holanda (2016) o *Simple Protocol and RDF Query Language* (SPARQL) é uma linguagem construída para consulta da Web semântica, capaz de permitir a recuperação de valores de dados estruturados e semiestruturados, a exploração dos dados quando realizada consultas com relações desconhecidas e uniões complexas de diferentes *datasets* em uma consulta única e simples.

Conforme a recomendação disponibilizada em Janeiro de 2008 pelo W3C essa linguagem de consultas contém capacidades necessárias para requisitar o que foi solicitado, pode ser usada para expressar buscas através de dados com diferentes fontes, estando eles armazenados nativamente como RDF ou vistos como RDF por meio de *middleware* e seus resultados podem ser tanto um conjunto de resultados quanto de grafos RDF. Realiza consultas de padrões em RDF, as buscas são enviadas por HTTP e os seus resultados podem ser disponibilizados tanto em formato XML quanto em formato JSON.

Nas versões atualizadas da consulta e do protocolo SPARQL foi especificado pela W3C em março de 2013 a possibilidade: de inserção, remoção a modificação dos dados RDF por meio do SPARQL1.1 *Update*; combinação de inferência pelo *Entailments*; consulta de uma única vez de muitos *endpoints*; resultados com formato de CSV/TSV (*Comma Separated Values*/*Tab Separated Value*) e outros.

Elias e Holanda (2016) dizem que a estrutura desta consulta é composta por declarações de prefixos que tem o intuito de abreviar URIs, definição do conjunto de dados informando os grafos RDF que são consultados, identificação da informação que deve retornar a partir da consulta pela cláusula de resultado, o padrão de consulta que está sendo usado que indica o que deve ser consultado dentro do conjunto de dados e os modificadores

de consulta, limites, ordenação, e outros que tem poder de modificar o resultado final da busca.

Observa-se que o uso de ontologias junto com a realização da fusão de dados e informações são capazes de objetivar e restringir os resultados selecionados por agentes extratores. Ferramentas e tecnologias como JSOUP, RDF e SPARQL são aptas para realizar captura, estruturação e consultas das informações obtidas, que no caso deste trabalho são retiradas da Web.

3. Trabalhos Correlatos

Arquiteturas de Extração e Recuperação de Informações com o uso de agentes já foram propostas por outras pesquisas, as quais realizam a extração da informação para a utilização em algum cenário posteriormente.

Em sua pesquisa Coneglian (2015) implementou um Agente Semântico de Extração, que contempla informações recuperadas da Web e das bases de dados internas que são usadas como fonte para a extração realizada pelo robô que foi construído. Essas informações que são recuperadas são persistidas em um Banco de Dados, formando o ambiente de Big Data.

O projeto referenciado acima teve como objetivo a criação dessa plataforma de Recuperação de Informações de toda Web, usando o contexto semântico das informações extraídas. Sendo capaz de permitir a localização, armazenamento, tratamento e das informações que são inseridas no contexto de Big Data.

Teve como intuito a transformação de informações que não são relacionadas em um ambiente de conhecimento estratégico, relevante, preciso e utilizável, permitindo que o usuário tenha acesso aos dados apenas quando este se encontrar com um maior valor agregado. Satisfazendo as necessidades informacionais desejadas pelo usuário quando adicionada uma semântica neste processo de Recuperação da Informação.

Esta proposta é de grande relevância pois é possível recuperar informação de uma maneira eficiente, sendo apresentado para o usuário apenas aquelas informações que passaram por todo o processo e tiveram valores agregadas. O fato da extração acontecer com auxílio da verificação semântica, acabada não agregando valor para resolver a dificuldade na realização da extração, pela forma estrutural, dos dados não estruturados ou semiestruturados existentes na internet.

Este projeto foi selecionado como trabalho correlato pelo fato de apresentar semelhanças com o projeto geral, proposto por Pereira (2016), que utiliza de elementos como Web Semântica, Fusão de Dados e Ontologias e ter como cenário um dos seus cenários para extração de dados e informações o meio Web, que foi o ambiente utilizado para captura dos dados neste trabalho.

Assim, pode ser observado que com relação com este projeto, o trabalho correlato tem como um dos ambientes de extração o meio Web que é o mesmo usado para o presente trabalho, porém usa de metodologias e ferramentas que não foram utilizadas no desenvolvimento deste, como é o caso da Web Semântica, Big Data e Banco de Dados.

4. Arquitetura de Referência do Projeto

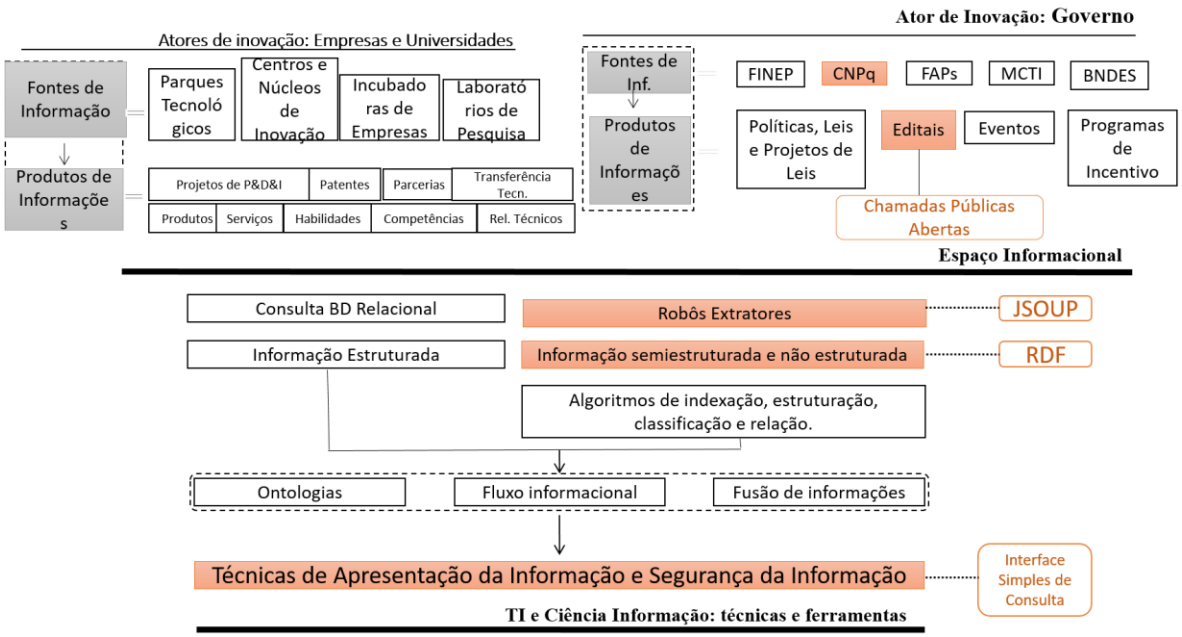
Já que não existe um padrão específico para os dados que são colocados no meio Web, são encontrados dados de diferentes lugares e com diversos formatos e estruturas, muitos destes podem ser semiestruturados ou até mesmo não-estruturados, não possuindo um formato padrão previamente definido.

Na recuperação de dados em uma consulta, uma alta quantidade de informações e dados podem não ter relevância real, sendo o próprio usuário o responsável por verificar as informações de maior importância.

Neste capítulo é apresentada a arquitetura de referência do projeto para extração de dados na Web, proposta por Pereira (2016) em seu projeto de pós-doutorado, esta é composta pelo conjunto dos elementos que participam de toda a estrutura. Essa arquitetura pode ser subdividida em duas, como mostrado na Figura 3, são essas subdivisões: os Atores de Inovação e o Espaço Informacional que é utilizado.

Essa subdivisão apresentada tem como intenção mostrar que as Fontes e os Produtos de Informações possuem uma grande variedade de informações que são relacionadas aos produtos desenvolvidos pelos Atores de Inovação e podem ser retiradas de locais diferentes. Também pode ser observado que para realização da automação de um robô extrator é necessário uma série de procedimentos deve acontecer com os dados que são retirados do cenário.

Figura 3 - Modelo da Arquitetura de Referência do Projeto



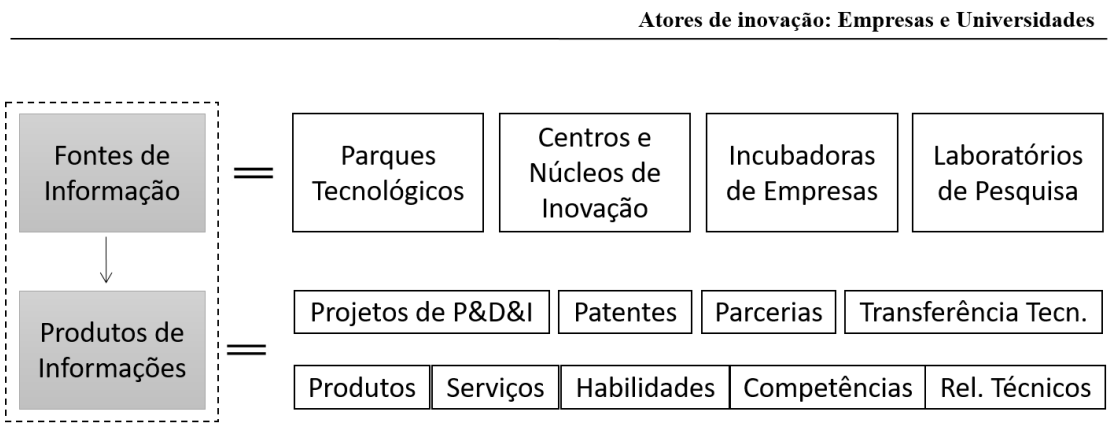
Fonte: Adaptado de Pereira (2016, p. 8)

A Figura 3 mostra as Fontes e os Produtos de Informações dos Atores de Inovação Tecnológica e no Espaço Informacional é encontrado o modelo com os elementos necessários para a construção de um Robô Extrator.

A arquitetura demonstrada na Figura 3 mostra o que é utilizado para realizar a retirada das informações que são usadas na construção da estrutura RDF, montada com auxílio do *Framework* Jena.

Os Produtos de Informações são os objetos produzidos pelas Fontes de Informação como mostram as Figuras 4 e 5, estas mostram os Atores de Inovação: Empresas, Universidades e Governo. A Figura 4 apresenta os ambientes tecnológicos e o que estes produzem no cenário correspondente.

Figura 4 - Estrutura dos Atores de Inovação: Empresas e Universidades

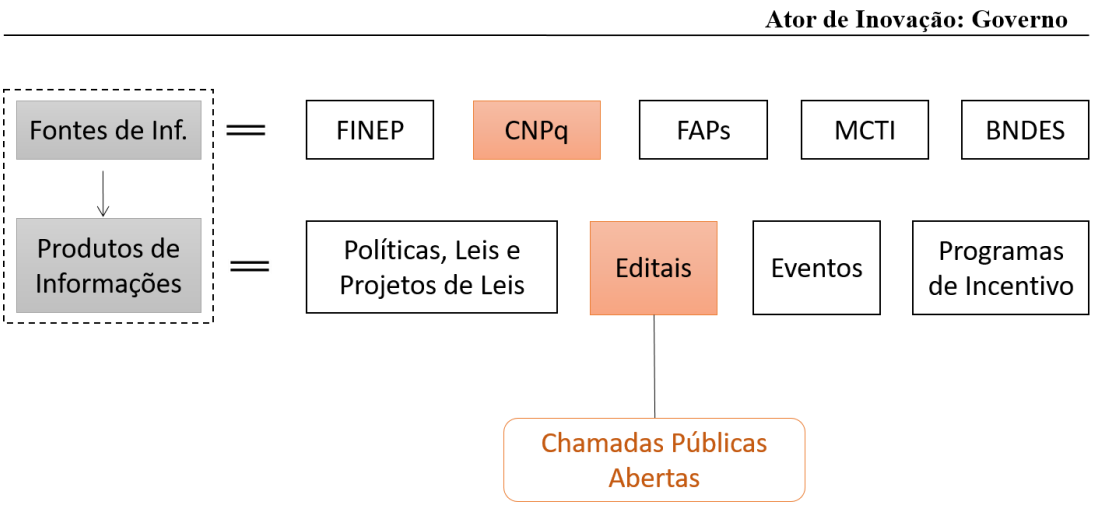


Fonte: Adaptado de Pereira (2016, p. 8)

Pode ser observada na Figura 5 que com relação as Fontes Informacionais disponíveis pelo Governo, os dados que foram extraídos são obtidos pelo Edital de Chamadas Públicas Abertas disponibilizado pelo CNPq. Estas informações são encontradas de maneira semiestruturada e são extraídas da internet pelo agente extrator.

As Figuras 4 e 5 mostram que não existe um único tipo de informação que são disponibilizadas e encontradas, mas sim uma vasta variedade de dados, sendo eles como forma de produtos, patentes, projetos, leis, eventos e outros.

Figura 5 - Estrutura do Ator de Inovação: Governo

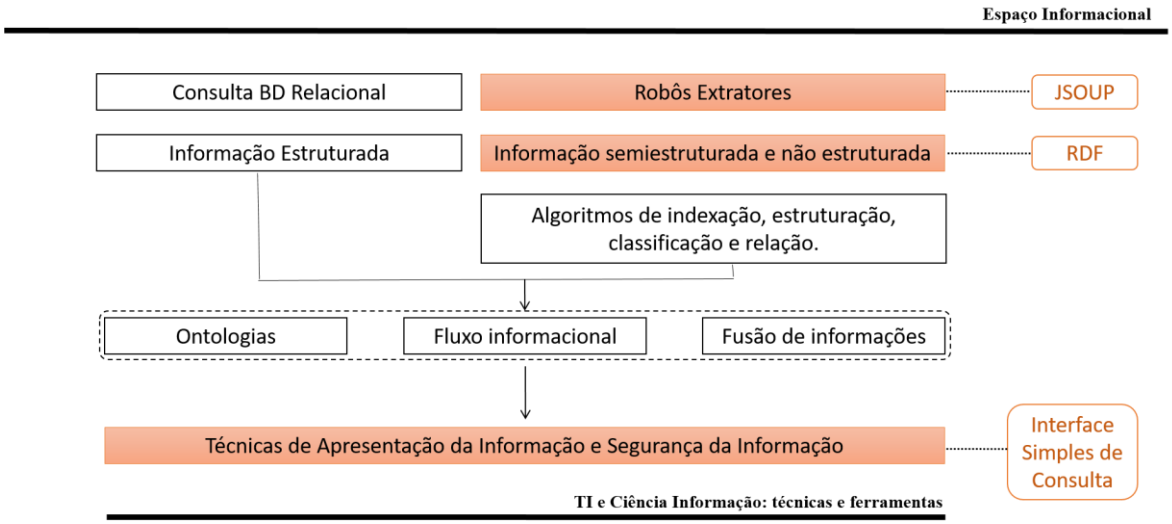


Fonte: Adaptado de Pereira (2016, p. 8)

Neste projeto apenas alguns destes elementos são utilizados, já que as informações extraídas podem ser semiestruturadas ou não-estruturadas, dependendo do local da extração.

A Figura 6 mostra que o Robô Extrator é desenvolvido com auxílio da biblioteca java Jsoup, capaz de realizar a leitura das informações semiestruturadas encontradas e retiradas da página do CNPq e estas são modeladas pelas triplas de RDF, com auxílio do *Framework* Jena e são apresentadas para o usuário final por meio de uma interface simples.

Figura 6 - Estrutura do Espaço Informacional



Fonte: Adaptado de Pereira (2016, p. 8)

O Espaço Informacional é composto por todos os elementos existentes para a realização da Extração de Informações. Para este projeto, é realizada a extração das informações desejadas com auxílio do JSOUP; a estruturação dessas informações no formato RDF; a consulta que é realizada pela linguagem SPARQL e a visualização das informações encontradas que são solicitadas pelo usuário.

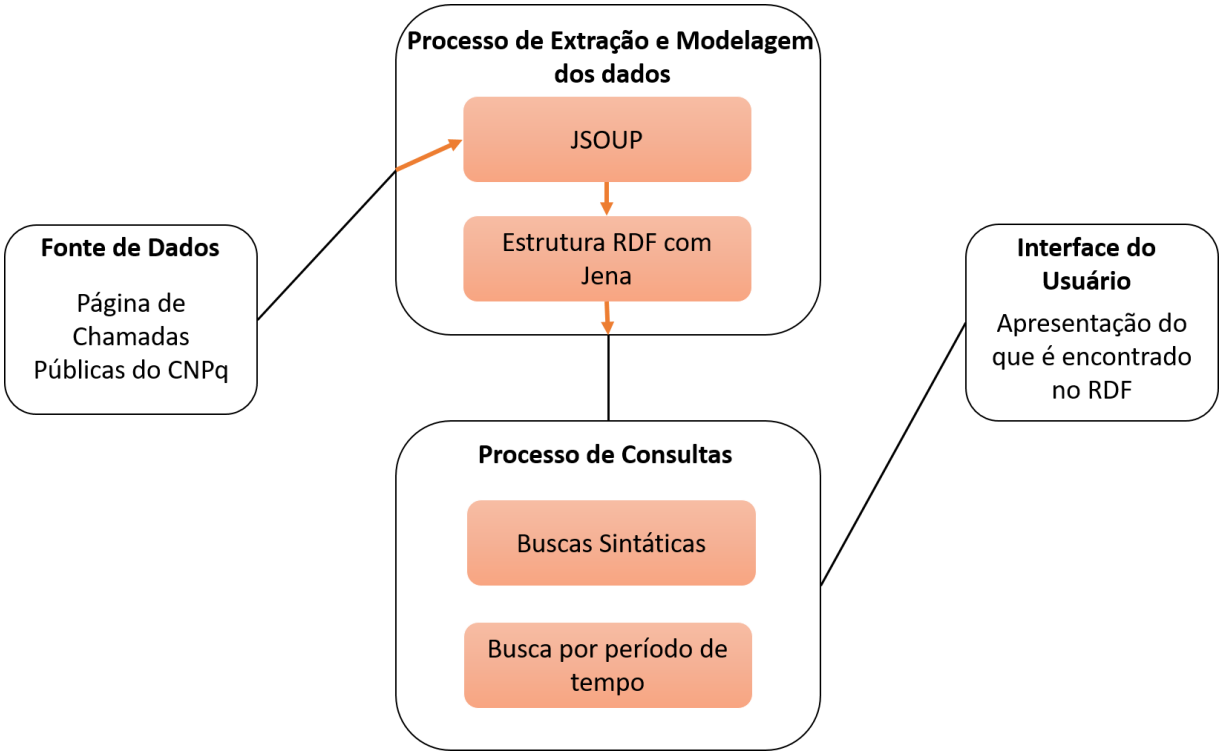
É observado que para a automação dos Robôs Extratores, desde o momento da leitura das páginas Web até o momento da apresentação dessas informações para o usuário final, que os dados recuperados passam por uma série de processos internos até que sejam apresentados ao usuário.

5. Extração e Modelagem dos Dados

A extração e estruturação das informações desejadas é realizada na parte do Espaço Informacional. Esse estágio do projeto é o responsável pela formatação das informações que são retidas e utilizadas posteriormente.

Os dados capturados são retirados da página de Chamadas Públicas Abertas do CNPq e estão presentes na página de forma semiestruturada, já que ela possui um formato de apresentação para o usuário que é pré-definido.

Figura 7- Diagrama do Processo de Extração e Modelagem



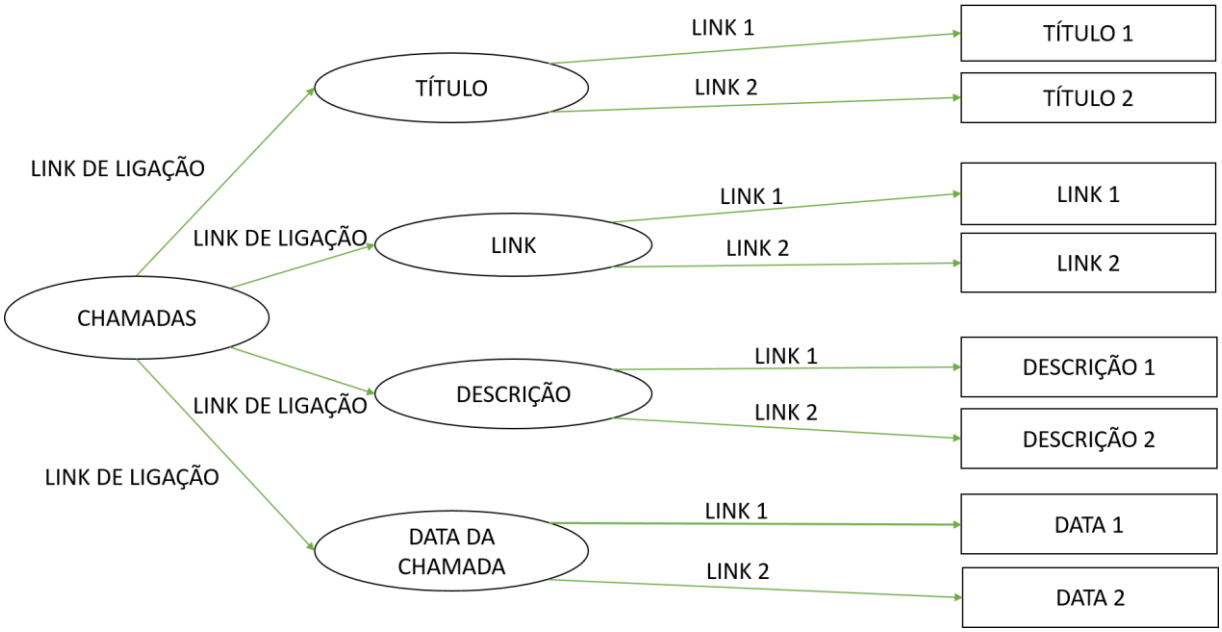
Fonte: Autoria Própria

O diagrama apresentado na Figura 7 mostra o caminho que é percorrido por uma determinada informação quando realizada uma consulta pelo usuário do sistema.

A Figura 7 mostra que os dados obtidos são encontrados na página de Chamadas Públicas Abertas do CNPq. Esses valores encontrados na Web passam pelo Processo de Extração e Modelagem, seguido pelo Processo de Consultas, até o determinado momento da visualização da informação que é solicitada pelo usuário.

Para conectar com a página da internet e obter os dados que são guardados no grafo RDF, foi utilizado o método .connect(this.getHtml()).get(), disponibilizado pelo Jsoup. Neste método é necessário o uso do link da página a qual se quer realizar essa conexão.

Figura 8 - Exemplificação Estrutura RDF Inicial



Fonte: Autoria Própria

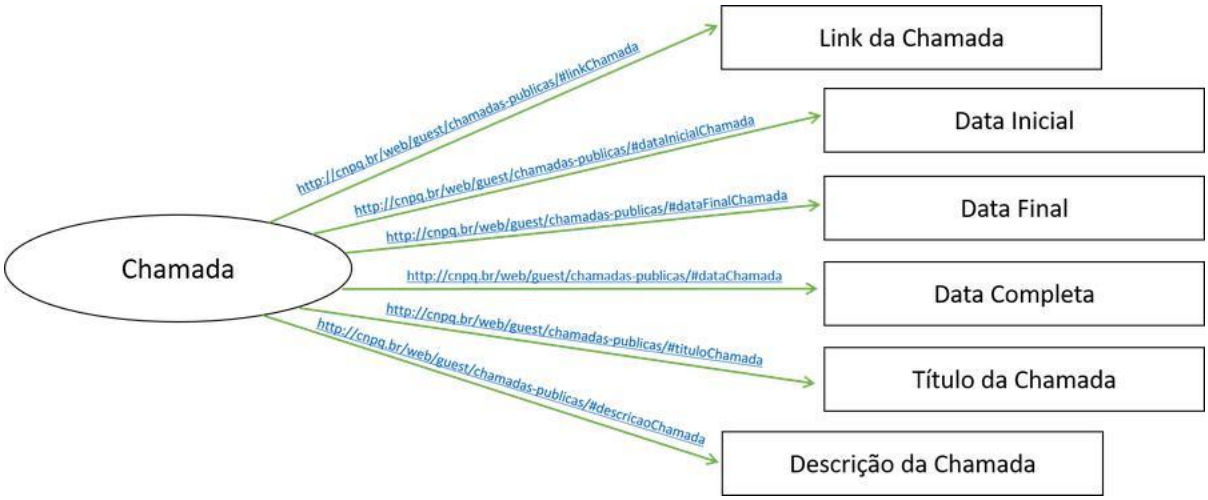
A Figura 8 exemplifica, de maneira genérica, como foi realizada a implementação da primeira estrutura RDF construída. Esta possuía um atributo genérico para chamada que era ligado por triplas com os elementos título, descrição, link e data que são guardados. Todos os valores que deveriam ser atribuídos a cada um dos atributos eram separados pela sua classificação e a ligação realizada entre o atributo e o valor era realizado pelo respectivo link que foi extraído.

A estrutura mostrada na Figura 8 foi considerada inviável para a realização das consultas que são realizadas. A consulta realizada pelo SPARQL no momento em que foi feita observando a estrutura demonstrada na Figura 8 não retornou nenhum resultado, fosse ele valor encontrado ou ocorrência de erro. Assim, pelo fato de se perder no percurso, em busca dos objetos, foi definido e implementado um novo modelo de RDF, conforme exemplificado na Figura 9.

De cada item existente na lista, que é apresentada, são retiradas as informações de link, data completa, título e descrição de cada chamada aberta disponibilizada. Para cada chamada existente nesta lista é gerado uma estrutura de triplas no RDF que contém as

informações que foram recuperadas e os elementos de Data Inicial e Data Final da chamada, como demonstrado na Figura 9.

Figura 9 - Estrutura RDF de Tópicos



Fonte: Autoria Própria

Nesse projeto foi definida a estrutura RDF apresentada acima. Esta é construída com o auxílio da biblioteca Jena, usada na linguagem Java. O recurso Chamada possui os literais (valores) de link, data inicial, data final, data completa, título e descrição que são conectados por links estáticos e específicos para cada valor, com o intuito de realizar a consulta das informações posteriormente.

Na Figura 10 é apresentado, um exemplo da estrutura de uma das triplas RDF, existente para cada recurso, que são construídas para estruturação das informações desejadas.

Figura 10 - Exemplo de tripla RDF para uma informação específica

Fonte: Autoria Própria

Para realizar a obtenção das informações desejadas é realizado uma conexão com a página especificada, esta ligação é realizada através de um método existente e disponibilizado pela biblioteca Jsoup.

Além da conexão com a página Web, o Jsoup é capaz de realizar uma varredura em toda ela e capturar qualquer informação que nela esteja disponibilizada. Por possuir métodos específicos para tais tarefas foi possível especificar os elementos da lista principal que deveriam ser capturados, usando principalmente o método getElementsByClass(), já que esses elementos são encontrados em classes específicas existentes no html.

Após a retirada das informações da página do CNPq, essas passam pelo processo de inserção no RDF que já teve suas propriedades e recursos criados anteriormente. Esse procedimento acontece com a adição dos valores dessas propriedades com auxílio do addProperty(), método do *Framework* aberto Jena.

Desse modo, foi construída a estrutura RDF a qual as informações capturadas são inseridas, possibilitando a realização das consultas realizadas de acordo com a necessidade do usuário.

6. Agente de Consulta

Assim como a extração das informações e a esquematização destas, as consultas feitas em cima da estrutura criada são realizadas na parte de Espaço Informacional deste projeto.

Durante o processo de decisão foi definido que as consultas que seriam realizadas pelo usuário são capazes de extrair informações, contidas no RDF, baseadas em palavras e num determinado período de tempo.

Para fazer as buscas desejadas é usada a linguagem para consultas de Web Semântica conhecida como SPARQL, que é disponibilizada pelo framework Jena. As *querys* usadas por essa linguagem fazem as buscas especificadas observando o grafo RDF construído, para que seja retornado as informações encontradas na estrutura é necessário que esta esteja modelada contendo as especificações separadas por módulos, conforme exemplificado na Figura 9.

Figura 11 - Exemplo da Consulta no Título

```
String queryStr
= "SELECT ?title ?descricao ?link ?data WHERE "
 + "{?x <http://cnpq.br/web/guest/chamadas-publicas/#tituloChamada> ?title FILTER (regex(?title, '"+testeTitulo1+"', 'i')   "
 + " || regex(?title, '"+testeTitulo2+"', 'i') "
 + " ). "
 + "?x <http://cnpq.br/web/guest/chamadas-publicas/#descricaoChamada> ?descricao . "
 + "?x <http://cnpq.br/web/guest/chamadas-publicas/#linkChamada> ?link . "
 + "?x <http://cnpq.br/web/guest/chamadas-publicas/#dataChamada> ?data }";
```

De acordo com o mostrado na Figura 11, é necessário que o item de retorno desejado seja referenciado pelo link de ligação, ou seja, a propriedade usada para ligação entre o recurso e o valor deve ser usado para que o SPARQL consiga realizar e retornar os valores encontrados durante a busca na estrutura. Em muitos casos, para realizar essa ligação entre o recurso e o valor, também é usada a declaração de um prefixo com a intenção de realizar a abreviação das URIs.

No *select* construído para realização da leitura por palavras é necessário o uso de um *filter* com a função *regex*. A expressão construída varre todo o RDF à procura dos elementos que são solicitados na busca feita pelo usuário pela interface. Para cada palavra informada deve ser construído e adicionado uma nova função *regex* no *filter* feito para consulta.

Figura 12 - Exemplo da Consulta por Data

```
String queryDataUma
                = "SELECT ?dataFim ?dataInicio ?title ?descricao ?link ?data WHERE "
                        + "{?x <http://cnpq.br/web/guest/chamadas-publicas/#dataFinalChamada> ?dataFim ; "
                        + " <http://cnpq.br/web/guest/chamadas-publicas/#dataInicialChamada> ?dataInicio . "
                        + " FILTER ( ?dataFim >= '2016-12-17' || ?dataInicio <= '2016-09-18' ) . "
                        + "?x <http://cnpq.br/web/guest/chamadas-publicas/#tituloChamada> ?title . "
                        + "?x <http://cnpq.br/web/guest/chamadas-publicas/#descricaoChamada> ?descricao . "
                        + "?x <http://cnpq.br/web/guest/chamadas-publicas/#linkChamada> ?link . "
                        + "?x <http://cnpq.br/web/guest/chamadas-publicas/#dataChamada> ?data }";
```

Já para as pesquisas realizadas pelas datas é necessário que os elementos usados na leitura das informações estejam em um formato específico, como mostrado na Figura 12.

Para essa consulta, a expressão implementada contém apenas a função *filter*, nela foi feita uma comparação entre as datas com o intuito de recuperar todas as chamadas públicas abertas existentes no intervalo de tempo especificado e que foram inseridas na estrutura RDF.

A implementação das consultas que são realizadas pelo SPARQL mostra que, para recuperar as informações corretas e desejadas, os agentes de busca utilizados devem estar previamente formatados, de acordo com o que é aceitado pela linguagem e dentro das funções especificadas para cada tipo de consulta que pode ser efetuada.

7. Resultados

Para verificação das informações que são encontradas de acordo com as consultas, dois tipos de pesquisa foram realizadas: uma busca em cima das palavras desejadas e outra para um determinado período de tempo.

Atualmente, estão disponibilizadas poucas chamadas na página Web do CNPq selecionada. Com isso, a estrutura RDF montada foi gerada com 2 itens, estes compostos com todas as informações desejadas.

O exemplo demonstrado na Tabela 1 representa a consulta por palavras. Nesse caso, a busca realizada foi pela palavra ‘apoio’.

Tabela 1 - Verificação das Chamadas Recuperadas quando a leitura por palavra

Título	Descrição	Período da Chamada	Possui a palavra solicitada?
Chamada CNPq/MCTIC Nº 016/2016 - SEGURANÇA ALIMENTAR E NUTRICIONAL NO ÂMBITO DA UNASUL	Objeto Apoiar projetos de pesquisa científica e tecnológica que visem contribuir significativamente para o desenvolvimento científico (...)	20/09/2016 a 16/12/2016	Não
APOIO À PESQUISA E À INOVAÇÃO EM CIÊNCIAS HUMANAS, SOCIAIS E SOCIAIS APLICADAS	A presente chamada tem por objetivo apoiar atividades de pesquisa de excelência, inovadoras e criativas, nos temas elencados nas Linhas de Pesquisa, com foco (...)	12/09/2016 a 23/01/2017	Sim

Na tabela 1, são apresentadas as informações de título, descrição e data de todas as chamadas que foram extraídas da página do CNPq e são encontradas no RDF, também pode ser encontrado se a chamada pública em questão possui a palavra usada na consulta.

As informações apresentadas na Tabela 2 são referentes a uma consulta que é realizada com um conjunto de palavras. Neste exemplo, a consulta foi realizada pelas palavras ‘apoio’ e ‘segurança’.

Tabela 2 - Verificação das Chamadas Recuperadas quando a leitura por um conjunto de palavras

Título	Descrição	Período da Chamada	Possui alguma das palavras solicitadas?	Palavra Encontrada
Chamada CNPq/MCTIC Nº 016/2016 - SEGURANÇA ALIMENTAR E NUTRICIONAL NO ÂMBITO DA UNASUL	Objeto Apoiar projetos de pesquisa científica e tecnológica que visem contribuir significativamente para o desenvolvimento científico (...)	20/09/2016 a 16/12/2016	Sim	Segurança
APOIO À PESQUISA E À INOVAÇÃO EM CIÊNCIAS HUMANAS, SOCIAIS E SOCIAIS APLICADAS	A presente chamada tem por objetivo apoiar atividades de pesquisa de excelência, inovadoras e criativas, nos temas elencados nas Linhas de Pesquisa, com foco (...)	12/09/2016 a 23/01/2017	Sim	Apoio

A Tabela 2 apresenta as informações de título, descrição e data de todas as chamadas que foram extraídas da página do CNPq e são encontradas no RDF, assim como se a chamada pública em questão possui alguma das palavras que foi usada na consulta e qual a palavra encontrada em cada título que foi retornado.

Para que as chamadas sejam mostradas na interface, quando é realizada uma consulta por um intervalo de tempo, é preciso que elas estejam no intervalo de tempo informado pelo usuário.

Para a demonstração da busca apresentada na tabela abaixo é usado o intervalo de tempo que possui data inicial em 23/09/2016 e data final em 18/12/2016.

Tabela 3 - Verificação das Chamadas Recuperadas por consulta por intervalo de tempo

Título	Descrição	Período da Chamada	Presente no Intervalo?
Chamada CNPq/MCTIC Nº	Objeto Apoiar projetos de pesquisa	20/09/2016 a 16/12/2016	Sim

016/2016 - SEGURANÇA ALIMENTAR E NUTRICIONAL NO ÂMBITO DA UNASUL	científica e tecnológica que visem contribuir significativamente para o desenvolvimento científico (...)		
APOIO À PESQUISA E À INOVAÇÃO EM CIÊNCIAS HUMANAS, SOCIAIS E SOCIAIS APLICADAS	A presente chamada tem por objetivo apoiar atividades de pesquisa de excelência, inovadoras e criativas, nos temas elencados nas Linhas de Pesquisa, com foco (...)	12/09/2016 a 23/01/2017	Sim

A tabela 3 mostra as mesmas informações da tabela 1 e 2 quanto ao que é apresentado sobre as chamadas, porém, a coluna relacionada a consulta exemplifica a leitura realizada no intervalo de tempo pré-definido.

Assim, é entendido que para as informações das chamadas serem retornadas de acordo com a busca realizada por um período de tempo é preciso que elas estejam, em algum momento, dentro do intervalo que foi solicitado. Caso dentro do período pedido não tenha informações de chamadas abertas, nenhum item será recuperado.

No decorrer da implementação do projeto, foram encontradas diversas dificuldades, principalmente relacionadas as consultas que deveriam ser realizadas. Com essas dificuldades, pôde-se observar que a consulta realizada com a linguagem SPARQL apenas é capaz de retornar os valores corretos se a estrutura montada para guardar as informações desejadas seja encontrada no padrão mostrado na Figura 9.

O esboço implementado neste projeto não possui a integração entre as informações que são extraídas da página por meio da Fusão de Informações, assim como também não são realizadas buscas observando a relação semântica das palavras consultadas. Também não foi realizado a modelagem de apresentação das informações, sendo, estes, apresentado para o usuário em uma interface simples, sem a informação de geolocalização dos dados.

Este trabalho é parte de um projeto geral que é composto em diversas etapas. O presente esquema tem como função contribuir com a validação de tecnologias, como o Jena e

Jsoup, que podem ser usadas no processo de extração e armazenamento dos dados que são usados por mecanismos no momento da recuperação pela linguagem SPARQL.

8. Conclusões

Esse projeto tem como objetivo o desenvolvimento de um robô de extração de dados semiestruturados capaz de extrair, estruturar e filtrar informações encontradas em uma página Web, utilizando bibliotecas e softwares como Jena, Jsoup e SPARQL. Com o intuito de traçar um caminho que melhore a extração e esquematização destes dados e objetive uma busca realizada pelo usuário.

Para comprovar este trabalho foi usada a estrutura em RDF, preenchida com as informações das chamadas públicas abertas. O autômato construído se depara com o problema de captura das informações ao encontrá-las semiestruturadas e em alguns casos não-estruturadas.

No momento em que a estrutura demonstrada na Figura 8 estava sendo utilizada, não era obtido nenhum tipo de retorno, fosse ele um resultado ou ocorrência de erro, nas tentativas de consultas que eram realizadas. Mostrando que existe uma estrutura padronizada para que o SPARQL seja capaz de realizar as consultas sob as informações que são salvas no RDF.

Para a realização dos testes, o robô de buscas foi implementado com a capacidade de extrair informações da página do CNPq e a consulta que pode ser realizada foi construída utilizando funções disponibilizadas pelo SPARQL. Para que a leitura do RDF aconteça da maneira correta, é necessário que tanto a estrutura gerada como os dados enviados pelo usuário, estejam construídos e formatados dentro dos padrões correspondentes para cada tipo de consulta que é construída.

Depois da realização dos testes, foi observado que o uso da consulta para a estrutura gerada é uma maneira eficaz para se obter informações pesquisadas, atendendo o que foi solicitado pelo usuário.

Grande parte das páginas são criadas para serem lidas apenas pelo homem, sem uma estrutura e formato que agentes computacionais consigam realizar a extração dos dados ali contidos dentro de um contexto. Assim, pode ser concluído que no momento que for inserida uma ontologia o autômato terá a capacidade de realizar as tomadas de decisões de acordo com valor semântico dos dados que são recuperados, caso aquela informação esteja dentro do contexto desejado pelo usuário.

Como não foi usada uma ontologia na construção deste projeto, os dados obtidos não possuem uma verificação baseada no significado e no contexto de termos buscados, sendo

o próprio usuário o responsável por decidir a relevância das informações apresentadas na interface.

O agente de extração construído realiza a leitura e captura dos elementos das chamadas públicas abertas disponibilizadas no site do CNPq que devem ser guardados, enquanto, a implementação realizada com o Jena é capaz de estruturar esses dados no RDF e disponibiliza a linguagem SPARQL para fazer a busca desejada pela estrutura, e assim apresentar os resultados encontrados na interface.

Desta forma, os resultados obtidos com a utilização do protótipo desenvolvido conseguem apresentar aos usuários os dados, filtrados pelas consultas por palavras e datas, que são obtidos de uma página semiestruturada.

Portanto, a consulta sobre as informações retiradas das páginas ocorre de maneira sintática, e a partir do que foi extraído, pode ocorrer uma análise das informações baseada na semântica no momento que for inserido o uso de uma ontologia nesse processo. Este método se mostrou muito eficiente, pois consegue realizar a extração e estruturação dos dados da página do CNPq e consegue fazer uma busca observando a sintaxe dos dados, assim apresenta para o usuário aquelas chamadas que possuem a palavra desejada no título e/ou que estão no período de tempo solicitado.

Para trabalhos futuros, uma busca realizada pelo usuário pode ser objetivada no momento que for inserida uma ontologia específica diretamente relacionada com a estrutura RDF, a recuperação e a fusão de dados e a Web Semântica com a intenção de construir o inter-relacionamento entre as informações ligadas aos Atores de Inovação, retiradas do cenário que é adotado como fonte informacional e a automação do Robô Extrator para as várias fontes de informações existentes e que disponibilizam dados neste ambiente.

Referências

APACHE. **The core RDF API**. Acessado em Mai 03, 2016. Disponível em: <https://jena.apache.org/documentation/rdf/index.html>.

APACHE. **Uma introdução a RDF e à API RDF de Jena**. Acessado em Mai 02, 2016. Disponível em: <https://jena.apache.org/tutorials/rdf_api_pt.html>.

BRÄSCHER, M. **WEB SEMÂNTICA**. 2007. Acessado em Abr 31, 2016. Disponível em: <http://www.stf.jus.br/arquivo/sijed/16.pdf>.

CITJUN. **A trajetória do Sistema de Inovação de Jundiaí**. Jundiaí, 2016. Acessado em: Mar 10, 2016. Disponível em: <http://www.incubadorajundiai.com.br/hist%C3%B3ria.html>.

CONEGLIAN, C. S. **Agente Semântico de Extração Informacional no Contexto de Big Data**. Marília, 2014. Acessado em Nov 13, 2016. Disponível em: <http://aberto.univem.edu.br/bitstream/handle/11077/997/Caio%20Saraiva%20Coneglian.pdf?sequence=1>.

DIAS, T. D.; SANTOS, N. **Web Semântica**: Conceitos Básicos e Tecnologias Associadas. Cadernos do IME: Série Informática, Rio de Janeiro, v. 14, p. 79 – 92, Junho 2003. Acessado em Mai 02, 2016. Disponível em: <http://www.e-publicacoes.uerj.br/index.php/cadinf/article/viewFile/6619/4734>.

ELIAS, E.; HOLANDA, O. **SPARQL**: Linguagem de Consulta em Ontologias. 2016. Acessado em Mai 13, 2016. Disponível em: <http://www.egov.ufsc.br/portal/sites/default/files/sparqlrevisado.pdf>.

FERREIRA, J. A.; SANTOS, P. L. V. A. da C. **O MODELO DE DADOS RESOURCE DESCRIPTION FRAMEWORK (RDF) E O SEU PAPEL NA DESCRIÇÃO DE ECURSOS**. Informação & Sociedade: Estudos, João Pessoa, v. 23, n. 2, p. 13 – 23, maio/agosto 2013. ISSN 1809-4783. Disponível em: <http://www.ies.ufpb.br/ojs/index.php/ies/article/view/15436/9681>.

FREITAS, F. L. G. **Ontologias e a Web Semântica**. Acessado em Abr 31, 2016. Disponível em: <http://www.inf.ufsc.br/~fernando.gauthier/EGC6006/material/Aula%203/Ontologia_Web_semantica%20Freitas.pdf>.

HEDLEY, J. **Jsoup**: Java HTML Parser. 2009-2016. ed. [S.l.], 2016. Acessado em Abr 27,2016. Disponível em: <https://jsoup.org/>.

INOVA MARÍLIA. **Centro de Inovação Tecnológica de Marília (CITec - Marília)**. Marília, 2016. Acessado em: Mar 10, 2016. Disponível em: <http://www.inovamarilia.com.br/citec-marilia/>.

JUNIOR, B. R. B. **FUSÃO DE DADOS PARALELA EM REDES DE SENSORES SEM FIO DENSAS UTILIZANDO ALGORITMO GENÉTICO**. Florianópolis 2008. Acessado em Mai 01, 2016. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/90842/254976.pdf?sequence=1>.

MORAES, M. O. **REENGENHARIA DO *ONTOCOVER***: UMA BIBLIOTECA JAVA PARA MANIPULAR ONTOLOGIAS EM APLICAÇÕES DA WEB SEMÂNTICA. Florianópolis 2007. Acessado em Mai 03, 2016. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_574/tccMarcelo.pdf>.

NAVARRO, M. B. M. de A. et al. **Inovação Tecnológica e as questões reflexivas do campo da biossegurança**. *Estudos Avançados,* São Paulo, v. 28, n. 80, Jan/Apr 2014. ISSN 0103-4014. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-40142014000100019>

OLIVEIRA, A. B. F. de; WERNECK, V. M. B. **Ontologias**. Cadernos do IME: Série Informática, Rio de Janeiro, v. 15, p. 7 - 13, dezembro 2003. ISSN: 1413-9014. Disponível em: <http://www.e-publicacoes.uerj.br/index.php/cadinf/article/viewFile/6384/4547>.

PARQTEC. **A instituição**. São Carlos, 2016. Acessado em: Mar 10, 2016. Disponível em: <http://www.pqtec.org.br/conheca-o-parque/historico.php>.

PARQUE TECNOLÓGICO PIRACICABA. **O Projeto**. Piracicaba, 2016. Acessado em: Mar 10, 2016. Disponível em: <http://www.parquetecnologico.piracicaba.sp.gov.br/index.php/o-parque>.

PARQUE TECNOLÓGICO DE SANTOS. **Conheça o Parque**. Santos, 2016. Acessado em Mar 10, 2016. Disponível em: <http://www.fpts.org.br/sobre.asp>.

PARQUE TECNOLÓGICO DE SOROCABA. **O Parque**. Sorocaba, 2016. Acessado em Mar 10, 2016. Disponível em: <http://www.empts.com.br/parque>.

PARQUE TECNOLÓGICO SÃO JOSÉ DOS CAMPOS. **Vocação para Ciência e Tecnologia**. São José dos Campos, 2016. Acessado em: Mar 10, 2016. Disponível em: <http://www.pqtec.org.br/conheca-o-parque/historico.php>.

PEREIRA, F. D. **Automação do fluxo Informacional entre atores de inovação no Brasil para processos de tomada de decisão**. Acessado em Fev 08, 2016. 2016.

PRYBECZ, B. H.; JÚNIOR, J. I. G.; MENDES, T. R. **Web Semântica**. Curitiba, 2013. Acessado em Mai 01, 2016. Disponível em: <http://www.inf.ufpr.br/bmuller/TG/TG-BTJ.pdf>.

São Paulo. **DECRETO Nº 56.424, DE 23 DE NOVEMBRO DE 2010**. Acessado em Fev 17, 2016. Disponível em: <http://dobuscadireta.imprensaoficial.com.br/default.aspx?DataPublicacao=20101124&Caderno=DOE-I&NumeroPagina=1>.

SANTAREM SEGUNDO, J. E. **Web Semântica**: introdução a recuperação de dados usando SPARQL. In: ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO: além das nuvens, expandindo as fronteiras da Ciência da Informação, 2014. Belo Horizonte. Anais... Belo Horizonte: UFMG/ ECI, 2014. p. 3863-3882.

São Paulo. **DECRETO Nº 60.286, DE 25 DE MARÇO DE 2014**. Acessado em Fev 17, 2016. Disponível em: <http://www.al.sp.gov.br/repositorio/legislacao/decreto/2014/decreto-60286-25.03.2014.html>.

SECRETARIA DE DESENVOLVIMENTO ECONÔMICO, CIÊNCIA, TECNOLOGIA E INOVAÇÃO. **CENTRO DE INOVAÇÃO**. Acessado em Fev 16, 2016. Disponível em: <http://www.desenvolvimento.sp.gov.br/centros-de-inovacao>.

SECRETARIA DE DESENVOLVIMENTO ECONÔMICO, CIÊNCIA, TECNOLOGIA E INOVAÇÃO. **REDE PAULISTA DE INCUBADORAS**. Acessado em Fev 16, 2016. Disponível em: <http://www.desenvolvimento.sp.gov.br/centros-de-inovacao>.

SEGUNDO, J. E. S. **Web Semântica:** conceitos e tecnologias. Grupo de Pesquisa Novas Tecnologias em Informação, Marília. Acessado em: Mai 03, 2016.

SILVA, G. C. **RDF e RDFS na Infra-estrutura de Suporte à Web Semântica**. Acessado em Mai 04, 2016. Disponível em: <http://www2.ic.uff.br/~gsilva/slreic.pdf>.

SOUZA, R. R.; ALVARENGA, L. A **Web Semântica e suas contribuições para a ciência da informação**. Ciência da Informação, Brasília 33.1. 132-141. 2004. Acessado em Mai 02, 2016.

SUPERA PARQUE DE INOVAÇÃO E TECNOLOGIA DE RIBEIRÃO PRETO. **Conheça o Parque**. Ribeirão Preto, 2016. Acessado em: Mar 10, 2016. Disponível em: <http://superaparque.com.br/conheca-o-parque/>.

UFBA. **Ambiente Virtual de Aprendizagem**. Universidade Federal da Bahia, 2007. Acessado em Fev 22, 2016. Disponível em: <http://www.moodle.ufba.br/mod/book/view.php?id=10902&chapterid=9850>.

W3C. **RIF RDF and OWL Compatibility (Second Edition)**. 2003. Acessado em Mai 03, 2016. Disponível em: < https://www.w3.org/TR/2013/REC-rif-rdf-owl-20130205/>.

W3C. **Resource Description Framework (RDF) Model and Syntax Specification**. 1999. Acessado em Mai 02, 2016. Disponível em: <https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>.

W3C. **RDF Primer**. 2004. Acessado em Mai 03, 2016. Disponível em: <https://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.

W3C. **RDF 1.1 Primer**. 2004. Acessado em Mai 03, 2016. Disponível em: <https://www.w3.org/TR/rdf11-primer/>

W3C. **RDF and SPARQL**: Using Semantic Web Technology to Integrate the World's Data. 2007. Acessado em Mai 05, 2016. Disponível em: <https://www.w3.org/2007/03/VLDB/>.

W3C. **SPARQL 1.1 Protocol**. 2013. Acessado em Mai 06, 2016. Disponível em: <https://www.w3.org/TR/sparql11-protocol/>.

W3C. **SPARQL Query Language for RDF**. 2008. Acessado em Mai 06, 2016. Disponível em: < https://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>.

W3C. **SPARQL Protocol for RDF**. 2008. Acessado em Mai 5, 2016. Disponível em: <https://www.w3.org/TR/rdf-sparql-protocol/>.

Apêndice A: CÓDIGO DE ESTRUTURAÇÃO E CONSULTAS DO RDF E LIGAÇÕES USANDO JSOUP, JENA E SPARQL

```
package model;

import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.apache.jena.query.Query;
import org.apache.jena.query.QueryExecution;
import org.apache.jena.query.QueryExecutionFactory;
import org.apache.jena.query.QueryFactory;
import org.apache.jena.query.QuerySolution;
import org.apache.jena.query.ResultSet;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class Call extends URL {

    private Model model;
    private Subject subject;
    private Predicate predicate;

    private String description;
    private String title;
    private String date;
    private String linkURL;
    private String dateStart;
    private String dateEnd;
    private String html;
    private Document doc = null;
    private Elements elements;
    private final SimpleDateFormat dateFormate;

    public Call() {
        subject = Subject.getInstance();
        predicate = Predicate.getInstance();

        subject.setCallURL("chamada");
        subject.setTitleURL("tituloChamada");
        subject.setDescriptionURL("descricaoChamada");
        subject.setDateURL("dataChamada");
```

```
    subject.setDateStartURL("dataInicialChamada");
    subject.setDateEndURL("dataFinalChamada");
    subject.setLinkURL("linkChamada");

    predicate.setHas("tem");

    model  = new Model();

    this.setDoc();
    this.setElements();

    dateFormate = new SimpleDateFormat("yyyy-MM-dd");
}
private String getDescription() {
    return description;
}

private void setDescription(String description) {
    this.description = description;
}

private String getTitle() {
    return title;
}

private void setTitle(String title) {
    this.title = title;
}

private String getDate() {
    return date;
}

private void setDate(String date) {
    this.date = date;
}

private String getLinkURL() {
    return linkURL;
}

private void setLinkURL(String linkURL) {
    this.linkURL = linkURL;
}

private String getDateStart() {
    return dateStart;
}

private void setDateStart(String dateStart) {
```

```
    try {
      Date dateSt = new SimpleDateFormat("dd/mm/yyyy").parse(dateStart);
      dateStart   = this.dateFormate.format(dateSt);
      this.dateStart = dateStart;
    } catch (ParseException ex) {
       dateStart = "";
       System.out.println("DateStart empty ");
       Logger.getLogger(Call.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public String getDateEnd() {
    return dateEnd;
}

public void setDateEnd(String dateEnd) {
    try {
       Date dateEn = new SimpleDateFormat("dd/mm/yyyy").parse(dateEnd);
       dateEnd     = this.dateFormate.format(dateEn);
       this.dateEnd = dateEnd;
    } catch (ParseException ex) {
       dateEnd = "";
       System.out.println("DateEnd empty");
       Logger.getLogger(Call.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private String getHtml() {
    return html;
}

private void setHtml() {
    this.html = "http://CNPq.br/web/guest/chamadas-publicas";
}

private Document getDoc() {
    return doc;
}

private void setDoc() {
   this.setHtml();
    try {
       Thread.sleep(5000);
       this.doc = Jsoup.connect(this.getHtml()).get();
    } catch (IOException ex) {
       this.doc = null;
       System.out.println("Comming here while NULL");
       Logger.getLogger(Call.class.getName()).log(Level.SEVERE, null, ex);
    } catch (InterruptedException ex) {
       Logger.getLogger(Call.class.getName()).log(Level.SEVERE, null, ex);
```

```
    }
  }
  private Elements getElements() {
    return elements;
  }

  private void setElements() {

    if ( this.getDoc() == null ) {
      this.elements = null;
    } else {
      this.elements = this.getDoc().getElementsByClass("content");
    }
  }

  private Boolean createResource() {
    if ( this.getElements() != null ) {
      for (Element element : this.getElements()) {
        //System.out.println("Response ->"+this.getElements());

this.setLinkURL(element.getElementsByClass("resultadosChamada").select("a").attr("href"));
        this.setTitle(element.select("h4").text());
        this.setDescription(element.select("p").text());
        this.setDate(element.getElementsByClass("datas").select("li").text());
        String[] dateArray = this.getDate().split(" ");
        this.setDateStart(dateArray[0]);
        this.setDateEnd(dateArray[2]);

        System.out.println("-----------------");
        System.out.println("Date       -> "+this.subject.getDateURL());
        System.out.println("-----------------");
        System.out.println("Title      -> "+this.getTitle());
        System.out.println("Date Start -> "+this.getDateStart());
        System.out.println("Date Link  -> "+this.subject.getDateStartURL());
        System.out.println("Date End   -> "+this.getDateEnd());
        System.out.println("Date Link  -> "+this.subject.getDateEndURL());
        System.out.println("-----------------");

        this.model.getModel().createResource(this.getLinkURL())
            .addProperty(this.model.getTitleProperty(), this.getTitle())
            .addProperty(this.model.getDescriptionProperty(), this.getDescription())
            .addProperty(this.model.getDateProperty(), this.getDate())
            .addProperty(this.model.getDateStartProperty(),this.getDateStart())
            .addProperty(this.model.getDateEndProperty(),this.getDateEnd())
            .addProperty(this.model.getLinkProperty(),this.getLinkURL());
      }
      return true;
    } else {
      return false;
    }
```

```
    }

    public ArrayList returnData (String word, String dateStart, String dateEnd, int type) {

        ArrayList<CNPq> listTitle = null;
        listTitle = new ArrayList<>();

        if ( this.createResource() ) {
            String queryStr;

            switch(type) {
                case 1:
                  queryStr = "SELECT ?title ?descricao ?link ?data WHERE "
                      + "{?x <"+this.subject.getTitleURL()+"> ?title FILTER ( "
                      +" "+word+" ) ."
                      + "?x <"+this.subject.getDescriptionURL()+"> ?descricao . "
                      + "?x <"+this.subject.getLinkURL()+"> ?link . "
                      + "?x <"+this.subject.getDateURL()+"> ?data }";
                   break;
                case 2:
                   queryStr = "SELECT ?dataFim ?dataInicio ?title ?descricao ?link ?data WHERE
"
                        + "{?x <"+this.subject.getDateEndURL()+"> ?dataFim ; "
                        + " <"+this.subject.getDateStartURL()+"> ?dataInicio . "
                        + " FILTER ( ?dataInicio >= '"+dateStart+"' || ?dataFim <= '"+dateEnd+"' ) .
"
                        + "?x <"+this.subject.getTitleURL()+"> ?title . "
                        + "?x <"+this.subject.getDescriptionURL()+"> ?descricao . "
                        + "?x <"+this.subject.getLinkURL()+"> ?link . "
                        + "?x <"+this.subject.getDateURL()+"> ?data }";
                   break;
                case 3:
                   queryStr = "SELECT ?dataFim ?dataInicio ?title ?descricao ?link ?data WHERE
"
                        + "{?x <"+this.subject.getDateEndURL()+"> ?dataFim ; "
                        + " <"+this.subject.getDateStartURL()+"> ?dataInicio . "
                        + " FILTER ( ?dataInicio >= '"+dateStart+"' || ?dataFim <= '"+dateEnd+"' ) .
"
                        + "?x <"+this.subject.getTitleURL()+"> ?title . "
                        + "?x <"+this.subject.getDescriptionURL()+"> ?descricao . "
                        + "?x <"+this.subject.getLinkURL()+"> ?link . "
                        + "?x <"+this.subject.getDateURL()+"> ?data }";
                   break;
                default:
                   queryStr = "SELECT ?title ?descricao ?link ?data WHERE "
                      + "{?x <"+this.subject.getTitleURL()+"> ?title FILTER
regex(?title,'"+word+"', 'i') . "
                      + "?x <"+this.subject.getDescriptionURL()+"> ?descricao . "
                      + "?x <"+this.subject.getLinkURL()+"> ?link . "
                      + "?x <"+this.subject.getDateURL()+"> ?data }";
```

```
                break;
            }

            Query query = QueryFactory.create(queryStr);

            try (QueryExecution qexec =
QueryExecutionFactory.create(query,this.model.getModel()))
            {
               ResultSet results = qexec.execSelect();

               while(results.hasNext())
               {
                  QuerySolution solution = results.nextSolution();

                  listTitle.add(new CNPq(solution.get("title").toString(),
                                 solution.get("descricao").toString(),
                                 solution.get("data").toString(),
                                 solution.get("link").toString()
                                )
                           );
               }
            }
         }

         return listTitle;
      }
}
```