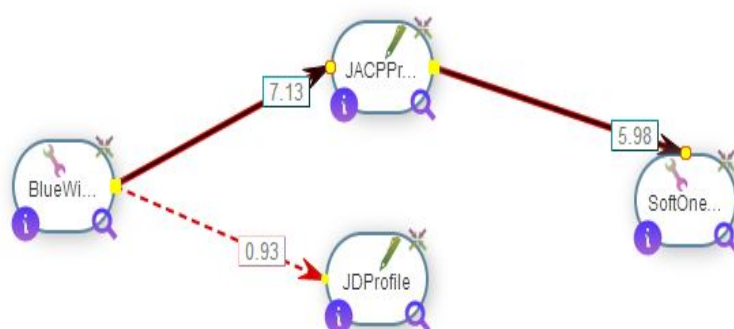




INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Eletrónica e Telecomunicações e de Computadores



e-Recruitment – Descrição e Inferência sobre Oferta e Procura

BRUNO MIGUEL ALVES CARRIÇO

(Bacharelato)

Trabalho de Projeto para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientador : Prof. Doutor Paulo Trigo Silva

Júri:

Presidente: Doutor Manuel Martins Barata

Vogais: Doutor Paulo Urbano

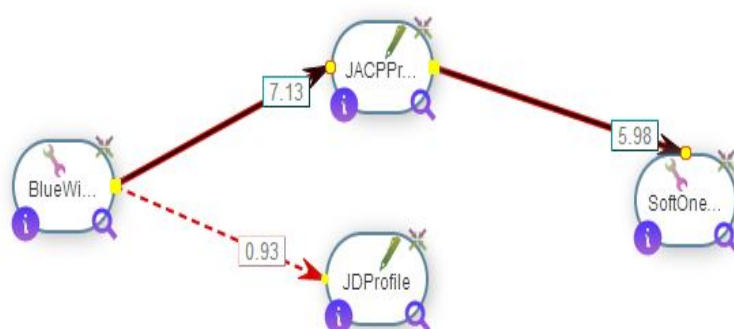
Doutor Paulo Manuel Trigo Cândido da Silva

Março, 2014



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Eletrónica e Telecomunicações e de Computadores



e-Recruitment – Descrição e Inferência sobre Oferta e Procura

BRUNO MIGUEL ALVES CARRIÇO

(Bacharelato)

Trabalho de Projeto para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientador : Prof. Doutor Paulo Trigo Silva

Júri:

Presidente: Doutor Manuel Martins Barata

Vogais: Doutor Paulo Urbano

Doutor Paulo Manuel Trigo Cândido da Silva

Março, 2014

A todos os que esperaram...

Agradecimentos

Ao meu orientador de projeto, Professor Doutor Paulo Trigo Silva, pela disponibilidade e orientação prestada.

À Virgínia, pelo apoio constante e motivação prestada ao longo do desenvolvimento do projeto.

À família e amigos pelo apoio e compreensão pela minha ausência, tantas vezes sentida, ao longo do projeto.

... e só se pode ver com o coração.

in O Príncipezinho

Antoine de Saint-Exupéry

Resumo

A procura pelo currículo que melhor responda às necessidades dos Recursos Humanos, bem como encontrar a proposta de trabalho que corresponda à expectativa de quem procura emprego, motivou a elaboração de um modelo que dê resposta eficiente a estas necessidades.

O modelo é baseado em ontologia, para representação de informação, bem como para consulta e troca desta, entre companhias e sistemas, numa forma clara e independente de *templates* e formatos próprios.

A opção por um modelo baseado em ontologia permite explorar a pesquisa na ontologia (TBox e ABox), através de equivalências semânticas entre conceitos (propostas de trabalho e candidatos).

É proposto um método para explorar a pesquisa na ontologia (TBox e ABox), através de equivalências semânticas entre conceitos (propostas de trabalho e candidatos) e, para classificar os resultados. A equivalência semântica é obtida de duas formas: a) via equivalência formal pela lógica de descrição (DL), e b) pela combinação entre a proximidade sintática e pela distância ao longo das relações da ontologia. A última abordagem permite encontrar pares de conceitos, que estão perto o suficiente, para serem considerados (pelo sistema) como sugestões (para o utilizador) de conceitos que poderiam ser encontrados através de equivalência formal (DL).

Foi desenvolvida uma interface cliente, onde a informação é apresentada num formato de Grafo Social, em que cada candidato ou proposta de trabalho é um nó. Os nós estão ligados entre si, com base na equivalência estabelecida entre eles. Desta forma, é possível ir expandindo os nós, e ter acesso aos seus pares, isto é a partir de uma proposta de trabalho expandem-se todos os nós de candidatos

x

que estão associados e, por sua vez, podem-se expandir os nós dos candidatos para se obterem-se as suas propostas de trabalho e assim por diante.

O modelo pode ser testado e validado, para tal, foi criado um ambiente fechado, com uma base de conhecimento restrita, onde se pode interagir com o sistema.

Palavras-chave: HR-XML Recruiting Specification, OWL/DL Description Logic, Recruitment Semantic Matching.

Abstract

The search for the best curriculum for the needs of the Human Resources, as well find job postings that matches the expectations of job seekers, motivated the development of a model that responds effectively to these needs.

This is an ontology-based model for information representation, querying and exchange, among companies and systems, in a clear way and independent of proprietary formats and templates.

The choice of an ontology-based model allows to explore searches in ontology (TBox and ABox) through semantic equivalences between concepts (job offers and job seekers).

Is also proposed a method to explore the ontology (TBox and ABox) searching for the semantic matching between concepts (job postings and job seekers) and compute the ranking of results. The semantic matching is achieved in two major ways: a) via description logics (DL) formal equivalence, and b) via a combination of syntactic proximity and the distance from a path traversal through the ontology relations. This last approach enables to find pairs of concepts that are close enough to be regarded (by the system) as suggestions (for the user) of concepts that can be lifted into a formal (DL) equivalence.

It was developed a client interface, where the information is presented in a form of Social Graph, in which each job seeker or job offer is a node. The nodes are connected together, based on the equivalence established between them. Thus, it is possible to expand the nodes, and have access to their peers, ie, from a job offer expand all nodes of job seekers who are associated and, in turn, can be expanded nodes of job seekers to obtain their job offers and so on.

The model can be tested and validated, to do this, we created a closed environment with a limited knowledge base, where you can interact with the system.

Keywords: HR-XML Recruiting Specification, OWL/DL Description Logic, Recruitment Semantic Matching.

Nomenclatura

ABox Assertional box

BOD Business Object Document

JAX-RS Java API for RESTful Services

JSON JavaScript Object Notation

OWL Web Ontology Language

RDF Resource Description Framework

REST Representational State Transfer

SOAP Simple Object Access Protocol

TBox Terminological box

XML Extensible Markup Language

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	2
1.3	Contributos	3
1.4	Organização do Documento	5
2	Trabalho Relacionado	7
2.1	Temas Abordados	7
2.1.1	Descrição (Representação da Informação)	7
2.1.2	Inferência (Interrogação e Recuperação de Informação)	10
2.1.3	Grafo Social e Interação via Browser	12
2.1.4	Pesquisa por Proximidade	12
2.2	Trabalhos Acadêmicos	13
3	Modelo Proposto	17
3.1	Abordagem	17
3.1.1	Arquitetura	18
3.1.2	Construção e Exploração da Base de Conhecimento	19
3.1.3	Apresentação Baseada em Grafo Social	23
3.2	Processo de Pesquisa	24
3.3	Correspondência entre Oferta e Procura	26

4	Implementação do Modelo	33
4.1	Representação de Conhecimento	33
4.1.1	Propriedades de Equivalência Semântica	34
4.1.2	Propriedades de Descrição	35
4.2	Equivalência Semântica entre Oferta e Procura	36
4.2.1	Exemplo de Equivalência Semântica entre Oferta e Procura	37
4.2.2	Sugestão de Equivalência	39
4.2.3	Ordenação do Resultado de uma Interrogação	41
4.3	Arquitetura	43
4.3.1	Interface Cliente	43
4.3.2	Servidor	46
5	Validação e Testes	49
5.1	Interrogação sobre Candidatos	50
5.2	Interrogação sobre Propostas	52
5.3	Avaliação dos testes	53
6	Conclusões e Trabalho Futuro	55
	Bibliografia	60
A	Representação de Conhecimento	i

Lista de Figuras

1.1	Propriedade <i>propertyDescription</i>	3
1.2	Exemplo de equivalência semântica e proximidade sintática	4
2.1	Exemplo de um processo de recrutamento	8
2.2	Arquitetura de um BOD	9
3.1	Representação simplificada do modelo	18
3.2	<i>Framework</i> Protégé	21
3.3	Representação em grafo social da relação entre propostas e candidatos	24
3.4	Processo de recrutamento	25
3.5	TBox e ABox relativo ao candidato	27
3.6	TBox e ABox relativo à proposta	27
3.7	Pesquisa <i>ad hoc</i> de proposta de trabalho	28
3.8	Pesquisa de candidatos por proposta de trabalho	29
4.1	Propriedades <i>owl:equivalentProperty</i> e <i>owl:sameAs</i>	35
4.2	Exemplo de equivalência semântica	37
4.3	Propriedades <i>owl:equivalentProperty</i> e <i>owl:sameAs</i>	38
4.4	Interface gráfica de pesquisa	39
4.5	Exemplo de sugestão de equivalência	41

4.6	Arquitetura proposta	43
4.7	Camada de apresentação - Grafo Social	44
4.8	Representação visual de Proposta e Candidato	45
5.1	Tipos inferidos no Protégé	54

Lista de Tabelas

2.1	Resumo das características por trabalhos relacionados	16
3.1	Elemento <i>PositionOpening</i> do <i>standard</i> HR-XML para OWL	20
3.2	Exemplo de utilização de um BOD para pesquisa de candidatos . . .	25
3.3	Base de conhecimento para ilustração do processo de correspon- dência	26
3.4	Conceitos da TBox relativos à candidatura	28
4.1	<i>Namespaces</i> : Diferentes domínios de informação	34
4.2	Simbologia da camada de apresentação	45

Listagens

3.1	XML schema - Elemento <i>PositionOpening</i> do standard HR-XML . . .	20
3.2	OWL - Classe <i>PositionOpening</i>	20
3.3	Código para indexação da ontologia	23
3.4	JSON - Exemplo de utilização do elemento <i>GetSearchDocument</i> . . .	25
3.5	Código para interrogação SPARQL DL	31
4.1	OWL - Exemplo de utilização da propriedade <i>propertyDescription</i> . .	36
4.2	OWL - Exemplo de utilização da propriedade <i>nameLiteral</i>	36
4.3	Código para obtenção de equivalência	40
4.4	Pontuação de equivalência do Lucene	42
4.5	Classe Java e assinatura do método do serviço <i>GetSearchDocument</i> .	46



Introdução

Neste capítulo é apresentada a motivação do trabalho, os objetivos propostos e os principais contributos alcançados.

1.1 Motivação

No atual espaço globalizado, o trabalho já não se restringe a áreas geográficas, sendo cada vez mais frequente o intercâmbio de recursos humanos entre instituições. A área do recrutamento tem vindo a estar cada vez mais assente na Internet [33], com portais e plataformas de emprego, públicos e privados. Estes portais permitem o registo de propostas de emprego, bem como de currículos e, disponibilizam também, consulta sobre esta informação.

No entanto, a informação trocada, bem como as interfaces de comunicação, são específicas de cada plataforma, o que dificulta o relacionamento entre a informação da oferta (propostas de trabalho) e da procura (candidatos/currículos).

Em geral as instituições propõem formatos específicos para submissão de currículos e propostas de trabalho, bem como para guardar toda a informação que estes contêm. A informação é tipicamente guardada em base de dados relacional, que permite apenas recuperação de dados estruturados. Para efetuar a sua recuperação é necessário conhecer a estrutura de dados, que campos existem e sobre estes efetuar a pesquisa. Uma das motivações do projeto é tornar a pesquisa

simples, sem ser necessário conhecer a estrutura de dados. Por exemplo, poder pesquisar currículos que contenham um determinado curso universitário, sem ter conhecimento de qual o nome do campo que tem a informação dos cursos.

1.2 Objetivos

Pretende-se com este trabalho desenvolver um sistema para que empresas de recrutamento possam facilmente obter informação de pedidos de emprego que estejam alinhados com as características dos seus recursos, ou, por outro lado, que quem procura recursos possa receber os currículos que estejam de acordo com determinado perfil.

O projeto assenta em dois pontos centrais, a equivalência semântica entre oferta e procura e a apresentação gráfica dessa mesma correspondência.

Propõe-se a utilização de uma ontologia OWL [41] para descrição formal dos conceitos e relações no domínio do recrutamento, onde são definidos os conceitos *proposta de trabalho* e *candidato*, que representam, respetivamente, a oferta e a procura.

O *standard* adotado no domínio do recrutamento, para troca de dados, define um método de pesquisa por par chave-valor. O par chave-valor constitui uma limitação do *standard*, pois é necessário ter conhecimento da estrutura para indicar corretamente qual a chave pela qual se pretende efetuar a pesquisa.

Para ultrapassar esta limitação é desenvolvido, neste trabalho, um modelo de representação e inferência baseado em "descrição-valor", onde a chave passa a ser uma descrição do campo que se pretende pesquisar. Por exemplo, para pesquisar currículos que contenham um determinado curso universitário, a chave pode conter a descrição "Curso Universitário", "Curso" ou outras. Assim, através da descrição, pode-se obter resultados que se enquadrem com o critério de pesquisa (chave).

Este projeto aplica tecnologias desenvolvidas no contexto da Web Semântica no contexto da área de Recursos Humanos, concretamente no domínio do recrutamento. São identificados modelos padrão (*standard*) nesta área, ao nível das ontologias e é proposta uma arquitetura para dar suporte à interação entre os intervenientes, em processos de recrutamento.

1.3 Contributos

Os contributos do trabalho assentam, nos seguintes temas:

Descrição com grafo de conceitos - Na ontologia, os conceitos estão relacionados por anotações semânticas, que permitem estabelecer relações de equivalência entre estes.

Através de uma anotação específica adicionada à ontologia, são estabelecidas descrições para as propriedades que relacionam os vários conceitos, como exemplificado na figura 1.1. Estas anotações permitem que as pesquisas efetuadas, com base no par chave-valor, possam ter como base o contexto do que se pretende pesquisar e não apenas o nome da propriedade onde incide a pesquisa.

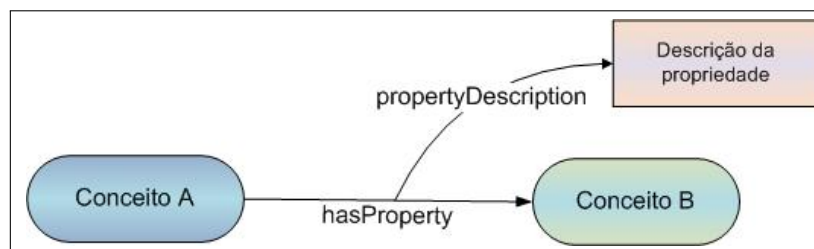


Figura 1.1: Propriedade *propertyDescription*

Inferência baseada em DL e pesquisa por proximidade - Na ontologia, os conceitos propostos de trabalho e candidatos podem estar semanticamente relacionados, entre si, por relações de equivalência formal, ou por grau de semelhança:

- A relação de equivalência formal, é criada através de anotações semânticas de equivalência entre os conceitos propostos de trabalho e candidato. Os conceitos ligados desta forma são, então, semanticamente equivalentes e podem ser inferidos em interrogações sobre os seus equivalentes.
- A relação por grau de semelhança, é obtida pela combinação entre a proximidade sintática e pela distância ao longo das relações da ontologia. Esta relação está diretamente dependente dos critérios de pesquisa, pois são semelhantes aqueles conceitos que estão próximos do conceito pesquisado.

Tendo em conta que um conceito tem outros conceitos que são equivalentes, ou por anotações semânticas, ou por grau de semelhança, permite que ao efetuar uma pesquisa, por par chave-valor, sejam obtidos os conceitos que estão diretamente relacionados com os critérios da pesquisa e os conceitos inferidos através de equivalência. A figura 1.2 representa a pesquisa efetuada por uma proposta P (instância do conceito proposta de trabalho), cujo resultado da interrogação à ontologia resulta na obtenção dos candidatos A , B e C (instâncias do conceito candidato). O candidato A é obtido via par descrição-valor, o candidato B é obtido via equivalência formal e o candidato C é obtido via grau de semelhança.

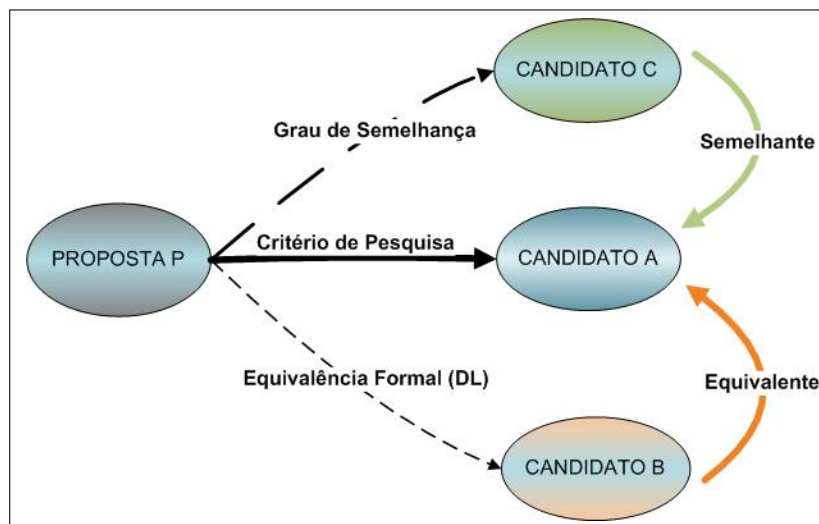


Figura 1.2: Exemplo de Equivalência Semântica e Proximidade Sintática

Os elementos obtidos por relações de equivalência formal são obtidos recorrendo a processos de inferência fundados na Lógica de Descrição (DL - do Inglês "Description Logic"). O processo de inferência permite obter (de modo automático) todos os elementos que estão relacionados via anotações semânticas, aquando da realização da interrogação sobre a ontologia.

Quanto aos elementos obtidos por grau de semelhança, são obtidos através da combinação de proximidade sintática com os elementos pesquisados e da distância ao longo das relações ontológicas destes.

Os conceitos e propriedades que constam na ontologia passam por um processo de indexação sendo construído um modelo vetorial, para suportar pesquisa por proximidade sintática.

A partir das relações apresentadas anteriormente, sobre os próprios elementos, onde candidatos têm relações entre si, bem como as propostas, passamos para as relações entre a propostas e a candidatos, que são obtidas por via formal, ou por grau de semelhança.

- A relação formal entre oferta e procura é obtida através do par chave-valor, pertencentes à proposta ou candidato, que tenham correspondência no outro.
- A relação por grau de semelhança é obtida através dos resultados do par descrição valor, encontrando sujeitos semelhantes aos obtidos.

Desta forma, são apresentados resultados formais, fruto dos critérios de pesquisa e da DL (equivalência formal), mas também resultados que possam ser relevantes no contexto da pesquisa efetuada, obtidos por grau de semelhança.

Apresentação de semelhança entre conceitos baseado em grafo social - É proposto um grafo para representação da relação entre procura e oferta, com interação humana. É possível interagir com o grafo, expandir o grafo ao longo das relações, alterar os critérios de relacionamento, criando, assim, um grafo dinâmico. É disponibilizada informação sobre as relações e sobre os elementos ligados por estas.

1.4 Organização do Documento

O presente capítulo faz o enquadramento geral do trabalho, apresentando a motivação, objetivos e contributos do trabalho desenvolvido.

O segundo capítulo apresenta os temas que serão abordados ao longo do trabalho. São apresentados, também, os trabalhos académicos relacionados com o presente trabalho.

O terceiro capítulo apresenta o modelo proposto para a concretização do trabalho.

O quarto capítulo apresenta todo o desenvolvimento efetuado, iniciando na construção da base de conhecimento, passando pela interrogação à base de conhecimento e terminando na apresentação dos resultados. É também descrita a arquitetura do trabalho.

O quinto capítulo apresenta as conclusões finais e alinha perspetivas de trabalho futuro.

2

Trabalho Relacionado

Este capítulo começa por apresentar as várias áreas abordadas no trabalho com indicação das tecnologias utilizadas em cada uma das áreas. Posteriormente são apresentadas as abordagens académicas relacionados com este trabalho. No final, o trabalho desenvolvido é enquadrado num quadro resumo que inclui as várias abordagens analisadas.

2.1 Temas Abordados

Segue-se a apresentação dos temas abordados no trabalho, com indicação das tecnologias utilizadas em cada um dos temas.

2.1.1 Descrição (Representação da Informação)

O consórcio HR-XML [6] é um organismo independente, dedicado ao desenvolvimento de especificações XML (*standards*), para permitir a troca de informação sobre recursos humanos. A especificação HR-XML Recruiting, vocacionada para a área do recrutamento humano, define *standards* sobre os modelos de dados (*schemas*) que compõem toda a informação relacionada com esta área.

Os *schemas*, que definem os elementos base da especificação, são:

- **Candidate** - descreve a estrutura de informação relacionada com um candidato, incluindo informação pessoal, académica e profissional;
- **PositionOpening** - descreve a estrutura de informação relacionada com uma proposta de trabalho, incluindo informação administrativa e requisitos do trabalho/função;
- **SearchDocument** - descreve a estrutura de informação que permite realizar pesquisas sobre candidaturas ou propostas de trabalho, com base em critérios específicos.

Esta especificação é usada neste projeto como base para a modelação da informação, quer na forma como é guardada, quer na forma como é disponibilizada.

O consórcio define também o formato de comunicação entre os diversos sistemas informáticos para troca de informação. A comunicação *standard* é realizada com troca de mensagens bem definidas, via SOAP.

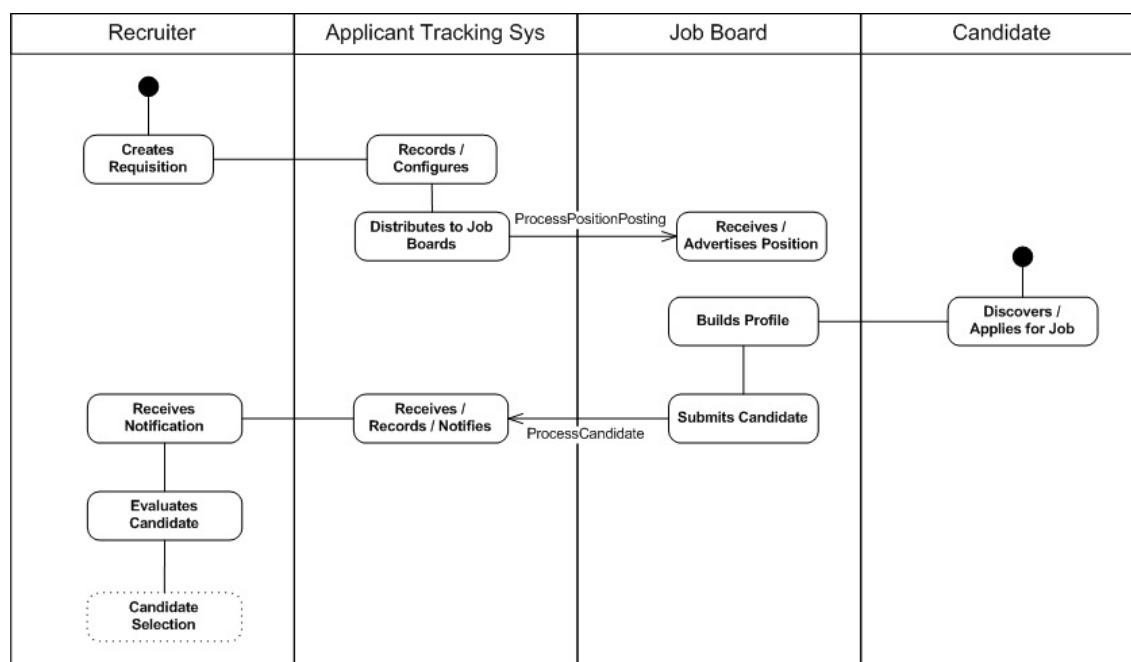


Figura 2.1: Exemplo de processo de recrutamento definido pelo HR-XML

As especificações e *standards* definidos pelo HR-XML, permitem suportar processos e fluxos (*workflows*) entre sistemas para diversos cenários de recrutamento. Por exemplo, suporta todo o processo desde a abertura de uma proposta de trabalho, até ao preenchimento da vaga por um candidato. O fluxo representado na figura 2.1, mostra este processo, descrito na documentação de especificação do HR-XML Recruiting.

Especificação dos Serviços

O HR-XML define *standards* para os serviços, para troca de mensagens entre sistemas. Os serviços permitem a comunicação de mensagens para criar, remover, atualizar ou apenas obter informação de candidatos ou propostas de trabalho.

Os principais serviços são:

Candidate - contém as mensagens associadas ao candidato;

PositionOpening - contém as mensagens associadas à proposta de trabalho;

SearchDocument - contém as mensagens associadas à pesquisa de elementos.

As mensagens são construídas de acordo com as metodologias do grupo Open Applications Group (OAGIS) [29], denominadas de Business Object Document (BOD), onde é mostrada na figura 2.2 a sua arquitetura base.

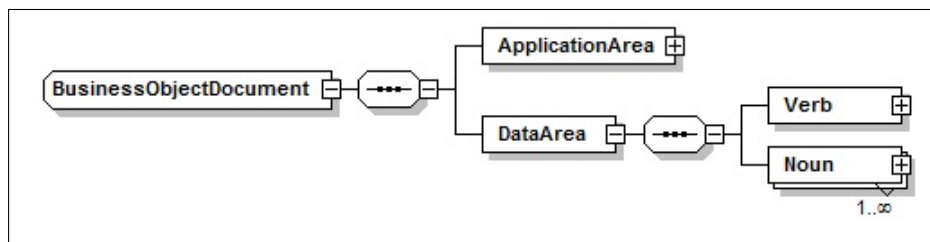


Figura 2.2: Arquitetura base de um BOD [6]

O BOD contém dois elementos:

ApplicationArea - pode conter informação específica da mensagem, como por exemplo o identificador ou versão da mensagem, ou outra informação relevante para a infraestrutura ou aplicação que recebe a mensagem.

DataArea - contém a informação de negócio, ou seja, os dados que são comunicados e qual a ação a realizar sobre os dados, como atualizar, remover ou outra.

Por sua vez a *DataArea* é composta pelos elementos:

Verb - identifica a ação a realizar sobre os dados enviados (*Noun*).

Noun - contém os dados de negócio enviados na mensagem, como um candidato ou proposta.

Os BOD's encapsulam o comportamento (*Verb*) e os dados (*Noun*) da mensagem, desta forma, permitem indicar, por exemplo, se se pretende criar, remover ou alterar (*Verb*) um candidato ou proposta (*Noun*).

Esta especificação é a referência para definir o protocolo de comunicação disponibilizado pelo nosso sistema. No entanto os serviços são disponibilizados através de *Web Services RESTfull*, assentes numa arquitetura REST (*Representational State Transfer* [34] [30]).

Seguir a especificação HR-XML Recruiting permite a interoperabilidade entre plataformas, onde os sistemas podem comunicar entre si, tanto no acesso à informação como à interpretação desta.

2.1.2 Inferência (Interrogação e Recuperação de Informação)

Este tema assenta sobre os seguintes tópicos, complementares entre si:

- Ontologia;
- Web Ontology Language (OWL);
- Lógica de Descrição (DL);
- Interrogação SPARQL [44].

Ontologia é o ramo da Filosofia que trata da natureza e existência do ser, que estuda o ser enquanto ser e o que é comum a todos os seres [26]. Na década de 90, o termo ontologia ganhou maior expressão no seio da comunidade de Engenharia do Conhecimento, pelo que Guarino (1998) propôs a distinção entre "Ontologia" ('O' maiúsculo) e "ontologia" ('o' minúsculo) de modo a definir e diferenciar o seu significado entre a Filosofia e a Engenharia do Conhecimento, respetivamente. Uma das primeiras definições de ontologia foi dada por Neches (1991), em que uma ontologia define os termos básicos e relações que compõem o vocabulário de uma determinada área, bem como as regras que combinam termos e relações para definir extensões ao vocabulário [7].

Uma das definições mais divulgadas é a de Tom Gruber (1993), em que uma ontologia é uma especificação explícita de uma conceptualização. Quando o conhecimento de um domínio é representado num formalismo declarativo, o conjunto de objetos que são representados é denominado de universo do discurso. Os nomes das entidades desse universo (por exemplo, classes, relações, funções ou outros

objetos) são definidas por texto legível, descrevendo o que os nomes significam, e por axiomas formais que restringem a interpretação e permitem o uso correto desses termos [17].

A ontologia é dividida em duas partes, TBox ("terminological box") e ABox ("assertional box"). A TBox contém os conceitos de um determinado domínio e as suas relações hierárquicas, enquanto que a ABox contém os indivíduos, as instâncias dos conceitos da TBox.

Web Ontology Language (OWL) é uma linguagem de Web Semântica, assente sobre o RDFS [43]. É baseada em lógica computacional, desenhada para representação de conhecimento, que permite definir e instanciar ontologias. Tem um conjunto de vocabulário que permite expressar semântica, através de relações e anotações sobre os termos ou conceitos. Por exemplo, na relação "é irmão" é possível definir que essa relação é simétrica; a atribuição dessa propriedade (de simetria) à relação permite, por exemplo, inferir que "B é irmão de A" ao sabermos que "A é irmão de B". Acrescenta, entre outras, a possibilidade de definir equivalência entre conceitos.

A Lógica de Descrição (DL) têm-se desenvolvido como uma família de formalismos para representação de conhecimento, que tem a capacidade de efetuar inferências [21].

OWL-DL é a junção de ambos, permitindo expressar semântica e efetuar inferência.

SPARQL [46] é uma linguagem de interrogação para RDF [42], que permite interrogar e obter resultados sobre uma estrutura de dados RDF.

SPARQL DL [45] é uma linguagem para interrogação sobre OWL. É um subconjunto do SPARQL que está adaptado para efetuar interrogações sobre ontologias, onde tira partido das anotações do OWL e, com o DL, tem capacidade de obter resultados por inferência.

Partindo da estrutura de dados definida no *standard* HR-XML, foi criada uma ontologia para representação de conhecimento sobre a área de Recursos Humanos. Sobre a ontologia são realizadas interrogações SPARQL, que permite obter informação direta à interrogação. No entanto, este tipo de informação, por si só, não acrescenta valor às pesquisas realizadas. O valor acrescentado está associado à Lógica de Descrição (DL) e Recuperação de Informação (RI). A DL permite inferir informação adicional, à recolhida pelo SPARQL, através da equivalência semântica entre a informação. Utilizando o OWL, através das relações e anotações semânticas, é possível processar a ontologia e realizar inferências sobre esta.

Para construção e edição da ontologia é usado o *framework* Protégé [38], que permite modelar ontologia e efetuar interrogações sobre esta.

2.1.3 Grafo Social e Interação via Browser

Grafo Social, cuja definição pode ser vista como uma representação de relações, teve o seu maior crescimento com o facebook, onde o seu fundador utilizou o termo para descrever relações entre pessoas [24].

O jsPlumb [19], é uma biblioteca que permite ligar e relacionar elementos em páginas web, permitindo, assim, criar esquemas e diagramas. Disponibiliza uma API que permite facilmente relacionar elementos, através de *endpoints* localizados nestes, atribuir diferentes estilos aos elementos, aos *endpoints* e às ligações entre estes. Cada elemento pode ter vários *endpoints* e cada *endpoint* pode ter várias ligações, permitindo assim que um elemento se relacione com outros através do mesmo *endpoint* ou, por outro lado, que um elemento possa estar relacionado com outros elementos por *endpoints* diferentes.

No trabalho apresentado, propostas e candidatos apresentam relações entre si, uma vez que uma determinada proposta tem candidatos que correspondem às suas necessidades e vice-versa. Temos, então, um grafo social entre estes, descrito pelas necessidades (critérios) que caracterizam cada um.

Aliando o conceito Grafo Social com a biblioteca jsPlumb, podemos criar uma representação para o grafo social numa página web.

2.1.4 Pesquisa por Proximidade

Para a pesquisa é utilizado o LARQ [13], que é um módulo do *framework* Apache Jena [12]. Este módulo combina o módulo ARQ [11], do mesmo *framework*, com o Lucene [14]. Permite associar pesquisa livre, sobre a ontologia, ao modelo vetorial, obtendo os resultados próximos dos critérios de pesquisa, com uma pontuação sobre os resultados obtidos.

O ARQ, módulo do *framework* Apache Jena, é um motor de pesquisa compatível com a especificação SPARQL 1.1 [44].

O Lucene, ou projeto Apache Lucene, suporta indexação, construção de modelo vetorial e pesquisa textual. Disponibiliza analisadores sintáticos para várias línguas, inclusivamente para Português.

2.2 Trabalhos Acadêmicos

M. Mochol, R. Oldakowski e R. Heese (2004) [27], propuseram a aplicação da Web Semântica no domínio do recrutamento. Tiveram como ponto de partida a plataforma "*virtual employment market*" da agência federal alemã para o emprego, BA (German Federal Employment Office), que tem como objetivo aumentar a transparência no mercado de trabalho.

Um dos principais problemas encontrados é que todos os participantes da plataforma têm de usar o formato proprietário para troca de dados definido pelo BA.

Para a concretização do cenário de recursos humanos, foi criada uma ontologia de recursos humanos (*HR-ontology*), tendo por base a versão alemã do *standard HR-XML*, *HR-BA-XML*.

Relativamente à correspondência entre propostas de trabalho e candidatos, estes são descritos através de vocabulário controlado da ontologia *HR-ontology*, o que permite calcular o grau de correspondência semântica entre ambos.

A informação sobre propostas de trabalho e candidatos está agrupada por grupos temáticos, "*thematic clusters*", por exemplo, informação sobre competências ou sobre o sector industrial. Através deste vocabulário controlado, agrupado por temas, são comparados os temas de uma proposta de trabalho com os temas de um candidato. A semelhança final entre uma proposta de trabalho e um candidato, é calculada como sendo a soma de todas as semelhanças entre os grupos temáticos de cada um. Uma vez que cada grupo temático tem um conjunto de vocabulário controlado, a semelhança entre os grupos temáticos é calculada com base na semelhança entre os seus conceitos.

No método usado, é determinada a semelhança entre dois conceitos através das suas posições hierárquicas e também pela semelhança das suas propriedades (relações com outros conceitos). Como exemplo, uma propriedade de um conceito do grupo de temas de competências (por exemplo, programação em java), é o nível de competência, que tem conceitos como por exemplo "*beginner*".

Com base nestas relações de semelhança, é definido o grau de semelhança e o peso, para os conceitos obtidos.

M. Fazel-Zarandi e M. S. Fox (2009) [10], combinam várias estratégias de correspondência para melhorar o processo de recrutamento. Propõe um modelo dedutivo para determinar o tipo de correspondência entre oferta e procura e um

modelo de classificação, baseado na semelhança (aproximação), entre estes.

Para este modelo foi criada uma ontologia baseada num modelo de competências (*SkillOnt*), onde são formalmente definidos os dois conceitos principais, candidato a emprego e proposta de emprego. Um candidato a emprego tem um conjunto de competências, níveis de competências e uma prova de competência para cada uma delas. A prova de competências pode ser um grau e/ou experiência de trabalho.

Um candidato expressa os seus requisitos em termos de descrições relacionadas com os requisitos de trabalho que deseja. Assim, para relacionar candidatos com propostas de trabalho apenas se tem em conta os requisitos. A correspondência entre proposta de trabalho e candidato é dividida entre qualificada e sub-qualificada (*Under-Qualified*). É considerada qualificada se todos os requisitos forem satisfeitos. A correspondência sub-qualificada dividi-se em:

- *Under-Qualified-Type-1* - um candidato é considerado sub-qualificado para a proposta de trabalho *D* se e só se: 1) o nível de competência for inferior ao nível de competência das competências requeridas em *D*; 2) no caso de ser pedido experiência de trabalho, o número de anos não for satisfeito. Ao determinar uma correspondência assume-se que as competências especificadas num requisito, existem no candidato.
- *Under-Qualified-Type-2* - se considerarmos todos os requisitos de uma proposta de trabalho, em que não se encontra correspondências em candidatos, quer correspondência qualificada, quer sub-qualificada, é possível iterar todos os requisitos que não foram satisfeitos, substituir uma competência de cada vez pelo seu "pai" (que é uma competência mais genérica) e realizar a pesquisa novamente, até que uma correspondência seja encontrada.

Para classificar os candidatos que correspondem a propostas de trabalho, são considerados três cenários, que correspondem aos três tipos de correspondência referidos anteriormente.

No primeiro cenário, as competências ou graus académicos desejados são usados para avaliar o grau de correspondência. É tido em conta o grau de interesse associado a cada requisito. O grau de correspondência global é a soma dos níveis de interesse das competências ou grau académicos que foram satisfeitos.

No segundo cenário são classificados os candidatos considerados sub-qualificados. Para estes casos, foram criadas duas medidas de dissimilaridade, uma baseada

nos níveis de competências e outra na experiência de trabalho. Esta medida é realizada através da diferença entre a distância do requisito e da competência/experiência de trabalho do candidato. Quanto maior for a diferença, maior é o nível de dissimilaridade.

O terceiro cenário envolve os casos em que um ou mais requisitos de competências não são satisfeitos. Para estes casos, é considerada uma medida de similaridade semântica, baseada em dois princípios:

- 1 - distância entre conceitos irmãos é maior do que a distância entre pai e filho;
- 2 - a distância semântica entre conceitos irmãos em níveis superiores é maior do que a distância semântica entre irmãos em níveis inferiores.

Usando esta medida de similaridade é calculado o grau de correspondência baseado na similaridade entre as competências que são pretendidas e as competências que realmente o candidato tem.

O trabalho apresentado difere dos trabalhos relacionados, descritos anteriormente, essencialmente, na forma como se define a equivalência entre propostas de trabalho e candidatos. Nos trabalhos relacionados é utilizado um conjunto de vocabulário controlado para descrição dos conceitos, onde a equivalência é estabelecida se os conceitos de um determinado requisito fizerem parte da proposta/candidato que se está a interrogar.

Neste trabalho a abordagem é diferente, pois é introduzida a pesquisa chave-valor, onde a equivalência é estabelecida se ambos forem satisfeitos, isto é, se houver um conceito *A* que tenha uma relação com *B*, cuja propriedade é satisfeita pela chave e *B* é satisfeito pelo valor do par chave-valor. Ainda, este método permite que quem interroga não tenha um conhecimento efetivo da ontologia nem das suas descrições.

A tabela 2.1 contém um resumo das características dos trabalhos apresentados.

Características	Trabalhos Relacionados		
	Ontology based recruitment process	Semantic matching for job recruitment	e-Recruitment
Descrição (HR-XML)	S	N	S
Inferência	S	S	S
Grafo Social	N	N	S
Pesquisa por Proximidade	N	S	S

Tabela 2.1: Resumo das características por trabalhos relacionados

3

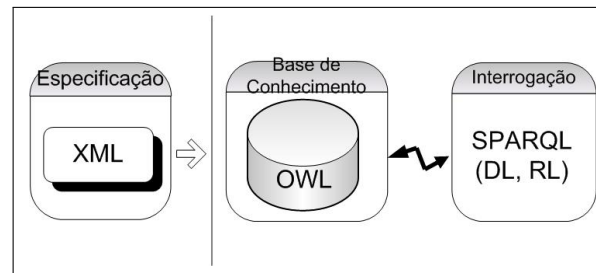
Modelo Proposto

Neste capítulo é apresentado o modelo proposto, começando pela abordagem adotada para a elaboração do trabalho. De seguida é apresentado o processo de pesquisa e por fim a correspondência entre oferta e procura.

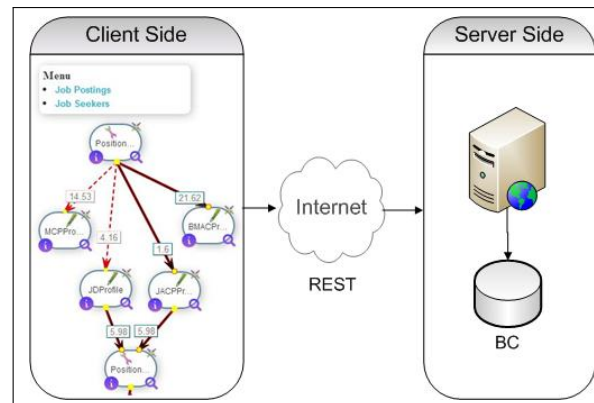
Para testar e validar o modelo, foi criado um ambiente controlado, com uma base de conhecimento restrita, onde se pode interagir com o sistema.

3.1 Abordagem

A abordagem adotada está apresenta na figura 3.1, onde se parte da especificação *standard* para a construção da base de conhecimento e se apresenta esta informação numa interface cliente. Cada uma destas fases estão descritas de seguida.



(a) Construção e exploração da base de conhecimento



(b) Arquitetura simplificada

Figura 3.1: Representação simplificada do modelo

3.1.1 Arquitetura

Nesta arquitetura, o servidor contém a base de conhecimento e disponibiliza *Web-services* para comunicação com o cliente. O cliente disponibiliza uma interface Web para interação com o utilizador, que permite que este efetue pesquisas sobre oferta ou procura de duas formas distintas:

- Pesquisa *ad hoc*, com base em critérios de pesquisa definidos pelo utilizador. A interface de cliente permite escolher se a pesquisa incide sobre propostas ou candidatos.
- Pesquisa através dos critérios de pesquisa próprios de cada elemento (proposta ou candidato). Cada elemento tem definido quais os critérios de pesquisa que lhe são relevantes, através dos quais se pretende obter equivalências.

O *Webservice*, disponibilizado pelo servidor, mantém a assinatura das mensagens – Business Object Model (BOD) – definidas pelo *standard* HR-XML de modo a

manter a compatibilidade com os sistemas baseados neste *standard*, disponibilizando métodos que permitem pesquisar e obter elementos de oferta e procura.

Os elementos retornados são, também, conforme a estrutura de dados do *standard*, permitindo, para além da compatibilidade entre sistemas, que mantenha total independência do módulo cliente.

O acesso ao servidor é realizado através do *Webservice* por este disponibilizado, de modo a efetuar as pesquisas invocadas pelo utilizador. A apresentação dos elementos obtidos do servidor é da sua responsabilidade, que os recebe na estrutura *standard* e os apresenta ao utilizador.

3.1.2 Construção e Exploração da Base de Conhecimento

Partindo da estrutura de dados definida no *standard* HR-XML, é definida e criada uma ontologia OWL, de modo a representar os conceitos do domínio de conhecimento, da área de recursos humanos, e os relacionamentos semânticos entre estes.

O HR-XML fornece a base para construção da *TBox* da ontologia, onde são criados os conceitos de proposta de trabalho e candidato. A ontologia é criada manualmente, onde são recriados os termos e conceitos presentes no HR-XML, bem como as suas relações e hierarquia. A estrutura de dados do *standard* contém informação muito vasta, por este motivo, é recriada apenas parte da sua estrutura na ontologia, o suficiente para ser possível criar um cenário capaz de dar suporte ao trabalho implementado.

Para exemplificar a passagem do XML para o OWL, vamos ter em conta o elemento *PositionOpening* do *schema* XML *standard*, apresentado na listagem 3.1 e a classe com o mesmo nome, *PositionOpening*, no OWL, apresentado na listagem 3.2.

Ao passar do XML para o OWL, são mantidos, maioritariamente, os nomes dos elementos do *schema*. O elemento *PositionOpening* do tipo *PositionOpeningType*, do XML, resulta numa classe *PositionOpening* no OWL. Este elemento contém o elemento *PositionProfile*, dando origem à propriedade *positionProfile* no OWL.

```

<!-- Elemento PositionOpening -->
<xsd:element name="PositionOpening" type="PositionOpeningType"/>
<xsd:complexType name="PositionOpeningType">
  <xsd:sequence>
    <xsd:element ref="PositionProfile" minOccurs="0" maxOccurs="unbounded"/>
    ...
  </xsd:sequence>
</xsd:complexType>

<!-- Elemento PositionProfile -->
<xsd:element name="PositionProfile" type="PositionProfileType"/>
<xsd:complexType name="PositionProfileType">
  <xsd:complexContent>
    <xsd:extension base="BasePositionProfileType">
      <xsd:sequence>
        <xsd:element ref="PositionPeriod" minOccurs="0"/>
        ...
      </xsd:sequence>
      <xsd:attribute name="languageCode" type="LanguageCodeContentType"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

Listagem 3.1: XML schema - Elemento *PositionOpening* do standard HR-XML

```

<owl:ObjectProperty rdf:about="pos#positionProfile">
  <rdfs:domain rdf:resource="pos#PositionOpening"/>
  <rdfs:range rdf:resource="pos#Position"/>
</owl:ObjectProperty>

```

Listagem 3.2: OWL - Classe *PositionOpening*

A tabela 3.1 ajuda a clarificar esta transformação.

Nome no XML	Nome no OWL	Propriedade LD	Conceito LD
PositionOpening	PositionOpening	positionProfile	Position

Tabela 3.1: Elemento *PositionOpening* do *standard* HR-XML para OWL

O elemento *PositionProfile* é do tipo *PositionProfileType*, no entanto, este elemento não resulta diretamente numa classe com o mesmo nome no OWL. Este elemento é do tipo *PositionProfileType* que por sua vez estende do tipo *BasePositionProfileType*, se fosse sempre criado um elemento com o mesmo nome no OWL, a ontologia ficaria demasiado complexa. Por este motivo, foi criada a classe *Position* que representa o elemento *PositionProfile*. Estão definidos os elementos necessários no OWL, a propriedade *positionProfile* que tem como domínio a classe *PositionOpening* e como contra-domínio a classe *Position*.

A ontologia tem informação sobre a experiência académica e profissional dos candidatos, bem como as suas preferências relativamente ao mercado de trabalho. A

ontologia contém sub-ontologias para classificação das atividades económicas no setor industrial, bem como para classificação dos níveis de ensino e graus académicos.

Sobre a ontologia são criados os indivíduos da *ABox*, ou seja, as instâncias dos conceitos definidos na *TBox*. São criados os candidatos, estabelecimentos de ensino destes, bem como outra informação necessária para descrever o âmbito da oferta e procura.

Para construção e edição das *TBox* e *ABox* da ontologia é utilizado o *framework* Protégé [38] (figura 3.2).

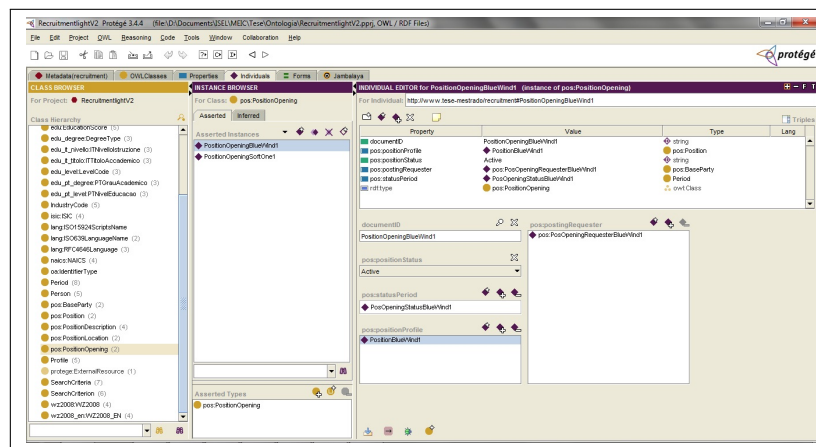


Figura 3.2: *Framework* Protégé

A ontologia, sobre a qual se realizam as interrogações SPARQL, tem por base o RDF, pelo que os seus recursos são relacionados por tripletos sujeito-predicado-objeto.

Para efetuar a pesquisa, interrogando a ontologia, é necessário transpor o par chave-valor do *standard* para ser utilizado sobre o tripleto. Os elementos chave e valor, do par chave-valor, são transpostos para os elementos predicado e objeto, do tripleto, respetivamente. Assim, são obtidos os sujeitos que correspondem ao par chave-valor, ou seja, os sujeitos cujo predicado/chave tem correspondência no objeto/valor. Na sua forma mais simples, os sujeitos são propostas/candidatos.

As interrogações sobre a ontologia, para obter candidatos que correspondam a determinada proposta de trabalho, e vice-versa, retornam vários elementos que correspondem à interrogação efetuada. No entanto, alguns elementos podem estar mais próximos da interrogação do que outros, pelo que é calculado o *ranking* destes elementos, de modo a classificá-los com grau de importância, permitindo

dar ao utilizador não só os resultados da pesquisa efetuada, mas também indicar quais os resultados mais importantes.

O ranking é calculado com base na proximidade sintática entre os termos que compõe os critérios de pesquisa e os termos dos elementos da ontologia. Cada critério de pesquisa é composto por um par chave-valor, em que os termos que compõe a chave são utilizados para efetuar interrogações sobre relações entre instâncias e, os termos do valor, são utilizados para efetuar interrogações sobre instâncias. Cada uma destas interrogações tem uma pontuação associada, pelo que o ranking é composto por ambas as pontuações. Ainda, se uma pesquisa tiver mais do que um critério de pesquisa, a atribuição do *ranking* tem em consideração todos eles.

Para a recuperação de informação, foi criado um índice sobre a ontologia, de modo a efetuar consultas sobre os elementos indexados, com base nos elementos provenientes da interrogação. A RI permite, assim, encontrar elementos que correspondam parcialmente à interrogação, e associá-los à interrogação SPARQL.

A indexação é criada através do Lucene, tendo como vantagem a melhoria em termos de eficiência da pesquisa e obtenção da informação. Adicionalmente, o índice devolve a pontuação sobre os termos pesquisados, que é usado no processo de classificação (*ranking*) dos resultados (conceitos) obtidos na pesquisa.

A listagem 3.3 contém um excerto do código onde se indexa a ontologia. Inicialmente é acedida a ontologia armazenada no sistema Jena TDB [15]. Para acesso e interrogação à ontologia, é necessário obter um modelo de acesso à ontologia. O modelo pode ser construído de acordo com as especificações RDFS, OWL ou OWL DL. Utilizamos a especificação OWL DL de modo a serem usadas as regras de inferência desta especificação.

Uma vez obtido o modelo, este é indexado pelo LARQ, ficando disponível em todas as interrogações SPARQL DL efetuadas à ontologia.

```
@Startup
@Singleton
public class DataAccess {

    static final String ASSEMBLER_FILE = "my_tdb-assembler.ttl";
    Dataset dataset;
    static OntModel model;
    static IndexLARQ index;

    @PostConstruct
    void init() {
        dataset = TDBFactory.assembleDataset(ASSEMBLER_FILE);
        Model dsModel = dataset.getDefaultModel();
        if (model == null) {
            model = ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM_RULE_INF, dsModel
            );
        }
        if (index == null) {
            index = buildIndex(model);
            LARQ.setDefaultIndex(index);
        }
    }
}
```

Listagem 3.3: Código para indexação da ontologia

3.1.3 Apresentação Baseada em Grafo Social

Foi desenvolvida uma interface Web, onde o utilizador pode efetuar as suas pesquisas e onde a informação é renderizada num formato tipo "Grafo Social". O grafo tem dois elementos base:

- nós;
- arestas.

Os nós representam candidatos e propostas de trabalho e têm componentes visuais que os permitem distinguir entre si.

As arestas ligam dois nós de tipos diferentes, isto é, relacionam candidatos com correspondência a determinada proposta e vice-versa. São direcionadas e apresentam o valor correspondente ao *ranking* dessa ligação. As arestas representam dois tipo de relação:

relação forte - obtida por equivalência formal;

relação fraca - obtida por grau de semelhança.

As arestas são representadas de forma diferente, de modo a evidenciá-las ente si. Este tipo de apresentação permite visualizar de forma mais intuitiva as relações entre:

- a) candidato e propostas de trabalho;
- b) candidatos que correspondam a algumas propostas de trabalho;
- c) propostas de trabalho que correspondam a candidatos e assim sucessivamente.

A figura 3.3 mostra a representação desta relações no grafo social.

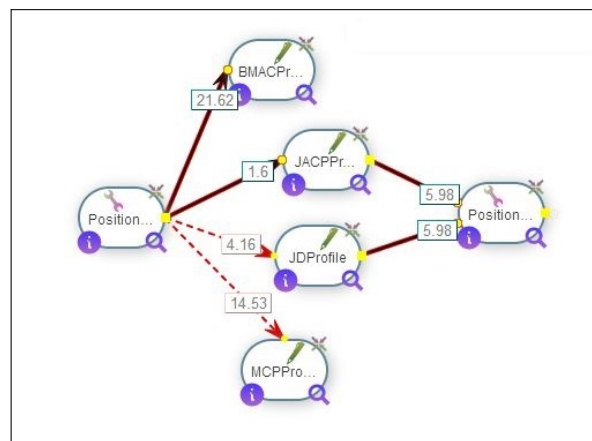


Figura 3.3: Representação em grafo social da relação entre propostas e candidatos

3.2 Processo de Pesquisa

De acordo com a abordagem e as especificações que serviram de suporte ao modelo, foi definido um processo para pesquisa de elementos (propostas e candidatos), com interação entre utilizadores e o sistema. O processo está representado na figura 3.4 e é detalhado de seguida.

Gestão

O sub-sistema Gestão recebe o pedido e verifica se se trata de pesquisas sobre propostas ou candidatos, reencaminhando o pedido para o sub-sistema Pesquisa. Ao receber o resultado, este processa o ranking e recolhe a informação necessária para construir a resposta ao pedido.

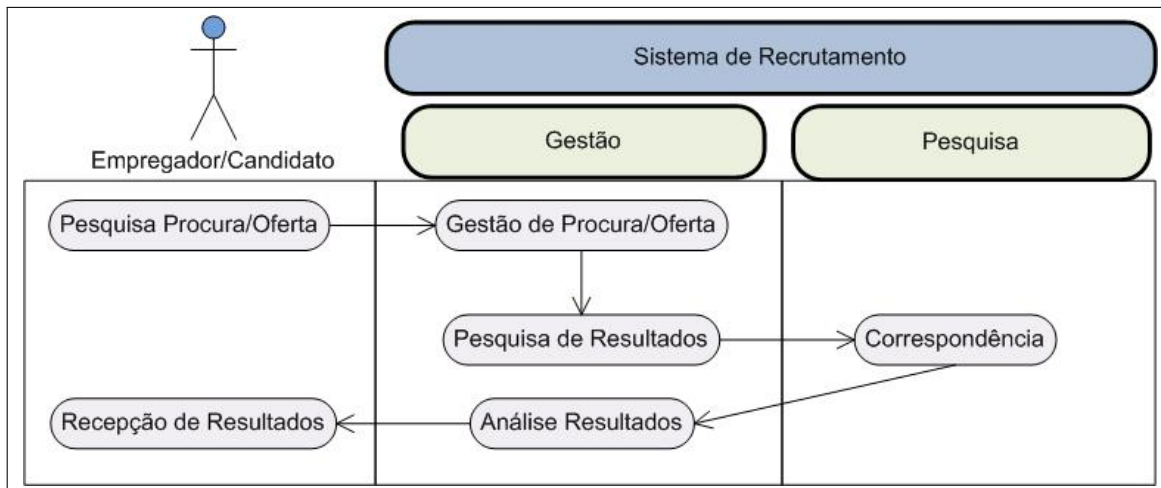


Figura 3.4: Processo de recrutamento

Pesquisa

Este sub-sistema efetua as pesquisas sobre a base de conhecimento.

Como exemplo, será acedido um serviço, implementado segundo a arquitetura REST, com envio de dados no formato JSON. Será enviada uma mensagem ao servidor para pesquisar candidatos, com um determinado critério de pesquisa. O serviço utilizado é o *SearchDocument*, com a mensagem *GetSearchDocument*. A estrutura BOD da mensagem enviada está representada na tabela 3.2.

BusinessObjectDocument	Verb	Noun
GetSearchDocument	Get	SearchDocument

Tabela 3.2: Exemplo de utilização de um BOD para pesquisa de candidatos

O elemento *GetSearchDocument*, em formato JSON, está representado na listagem 3.4, com os dados relativos ao critério de pesquisa.

```

{
  "DataArea": {
    "SearchDocument" : {
      "SearchCriteria" : {
        "SearchCriterion" :
          [
            {
              "CriterionName" : "Academic degree",
              "CriterionValue" : "Master"
            },
            {
              "CriterionName" : "Industrial Classification Code",
              "CriterionValue" : "Soybean Farming"
            }
          ]
      }
    }
  }
}
  
```

Listagem 3.4: JSON - Exemplo de utilização do elemento *GetSearchDocument*

Este elemento contém os critérios de pesquisa, em pares chave-valor. Segundo o *standard* podem existir vários pares chave-valor, dentro de um elemento *Search-Criterion*, cuja representação é *CriterionName:CriterionValue*.

Neste caso são enviados dois critérios, "Academic degree": "Master" e "Industrial Classification Code": "Soybean Farming".

3.3 Correspondência entre Oferta e Procura

Nesta secção é explicado, através de um exemplo ilustrativo, o processo de correspondência entre oferta e procura.

Os dados apresentados no exemplo são dados da ontologia e as correspondências entre oferta e procura são obtidas através do sistema implementado. Os elementos da base de conhecimento relevantes para o exemplo estão descritos na tabela 3.3. São apresentados num formato tabular de forma a facilitar a leitura dos mesmos.

Termos	Descrição
BlueWind	Empresa (proposta de trabalho)
SoftOne	Empresa (proposta de trabalho)
JACPProfile	Perfil do Candidato - representa o candidato JACP
JDProfile	Perfil do Candidato - representa o candidato JD
JACP	Candidato
JD	Candidato
01.12	Código da classificação WZ2008 ¹
Anbau von Reis	Decrição do código 01.12 da classificação WZ2008
0112	Código da classificação ISIC [28]
Growing of Rice	Decrição do código 0112 da classificação ISIC

Tabela 3.3: Base de conhecimento para ilustração do processo de correspondência

Os elementos da base de conhecimento indicados na tabela estão descritos no formato OWL, onde consta a *TBox* e *ABox*. As *TBox* e *ABox* relativas à candidatura estão representadas na figura 3.5 e relativas à proposta na figura 3.6. Os conceitos indicados em ambos os OWL são apenas a parte que permite acompanhar o exemplo.

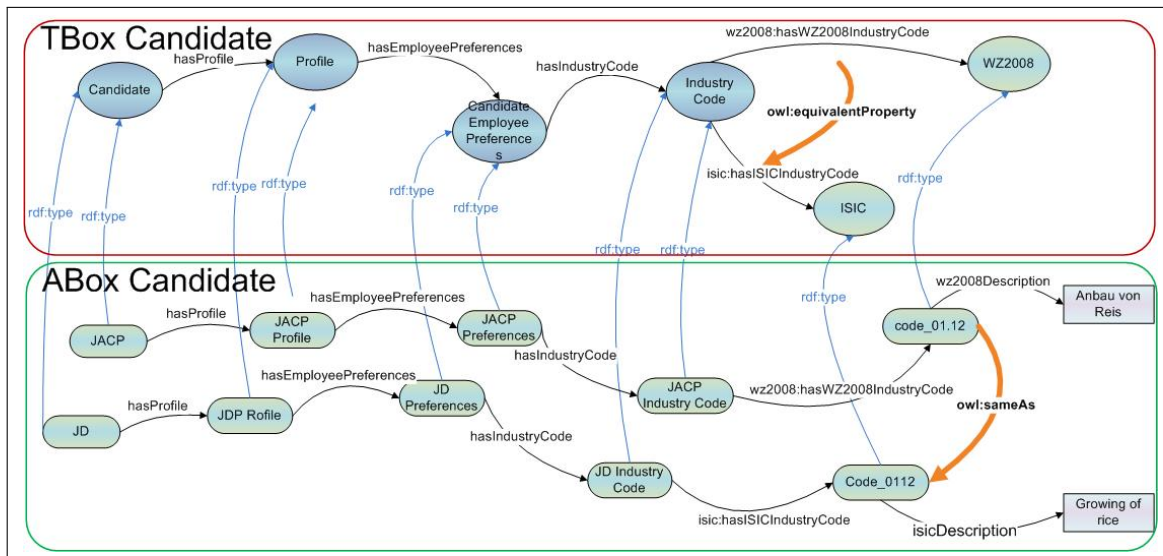


Figura 3.5: TBox e ABox relativo ao candidato

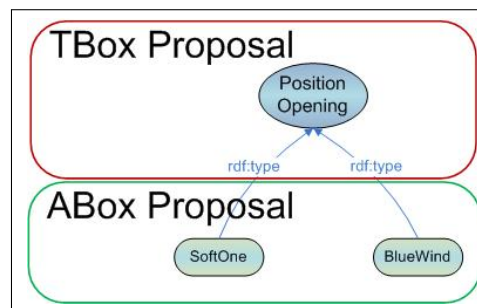


Figura 3.6: TBox e ABox relativo à proposta

Os conceitos da *TBox* da candidatura estão descritos na tabela 3.4.

A pesquisa, segundo o *standard*, é efetuada por pares chave-valor. Cada par corresponde a um critério de pesquisa e, numa pesquisa, podem ser utilizados tantos pares quantos se pretender. Como resultado da pesquisa, obtemos os elementos de procura (propostas) ou de oferta (candidatos) que correspondem aos critérios indicados.

No entanto, a interrogação sobre a ontologia é um pouco mais complexa, uma vez que a ontologia é um grafo e, como tal, existem relações hierárquicas entre os seus recursos. Posto isto, um critério de pesquisa pode incidir sobre qualquer nível hierárquico, pelo que o sujeito obtido pode não ser um elemento de proposta/candidato. Nestas situações é necessário percorrer a hierarquia de modo a alcançar o elemento de topo (um elemento de proposta/candidato).

A figura 3.7 mostra um caso de uso de pesquisa *ad hoc*, onde se pretende pesquisar

Conceito	Descrição
Candidate	Candidato
Profile	Perfil de um candidato
CandidateEmployeePreferences	Preferências do candidato sobre o trabalho/funções a desempenhar
IndustryCode	Atividades económicas
WZ2008	Classificação alemã das atividades económicas
ISIC	Classificação das Nações Unidas das atividades económicas

Tabela 3.4: Conceitos da TBox relativos à candidatura

ofertas de trabalho com localização em Lisboa. A pesquisa é realizada *ad hoc*, sobre as propostas, com base em critérios de pesquisa definidos pelo utilizador.

Figura 3.7: Pesquisa *ad hoc* de proposta de trabalho

Os critérios de pesquisa são encapsulados na estrutura *standard* em pares chave-valor, na interface cliente, e enviados para o servidor. Como resultado da interrogação à ontologia, é obtida a instância de uma proposta da empresa BlueWind.

A proposta de trabalho devolvida tem critérios sobre os quais se pretende encontrar correspondência nos candidatos.

A figura 3.8 exemplifica o caso de utilização de pesquisa através dos critérios de pesquisa próprios de cada elemento (proposta ou candidato). Neste exemplo são pesquisados os candidatos que se enquadrem com os critérios de pesquisa da proposta.

São mostrados os critérios de pesquisa da proposta, e os resultados (candidatos) obtidos pela pesquisa, que satisfazem os seus critérios.

O critério de pesquisa é representado através do seguinte par chave-valor:

chave: German economical activities;

valor: rice farming.

Da pesquisa efetuada são obtidos dois candidatos, o JACP e o JD, via equivalência formal e equivalência por proximidade sintática, respetivamente. Ambas as equivalências são apresentadas de seguida.



Figura 3.8: Pesquisa de candidatos por proposta de trabalho

Equivalência Formal

O candidato JACP, referido anteriormente (figura 3.8), não tem na sua informação nenhuma referência direta ao critério de pesquisa que lhe deu origem, como se pode constatar pela *ABox*. No entanto, este candidato tem associado ao seu perfil uma das atividades económicas da classificação WZ2008, cujo código é o 01.12 ("Anbau von Reis"). A descrição da classificação está descrita em alemão, pois é a classificação alemã WZ2008.

O critério de pesquisa utilizado, "rice farming", está descrito em inglês e tem como correspondência a atividade da classificação ISIC, 0112 ("Growing of rice"). Não existe relação entre o candidato JACP e a classificação ISIC, no entanto, é possível obter este candidato pois os códigos 01.12 da classificação WZ2008 e 0112 da classificação ISIC estão relacionados semanticamente, isto é, são semanticamente iguais.

A equivalência é representada na figura 3.8 por uma linha cheia.

Equivalência Sintática

Ao efetuar a pesquisa por grau de semelhança, é interrogada a ontologia para obter os sujeitos que tenham como objeto a chave "rice farming". O código industrial NAICS 111160 corresponde à interrogação, uma vez que a sua descrição

é "Rice Farming".

Para obter os candidatos equivalentes, por grau de semelhança, é percorrida a ontologia de modo a verificar que instâncias do conceito de candidato têm na sua hierarquia relação com este código industrial. O candidato encontrado é o JD, sendo considerado equivalente por grau de semelhança. Para o resultado obtido é verificada a distância entre os conceitos obtidos pelo valor do par chave-valor, relevante para o cálculo do *ranking*.

Cada aresta do grafo é etiquetada com o valor de relevância da relação que estabelece, isto é, o ranking do novo nó. Por exemplo, o nó BlueWind tem duas arestas que o ligam a dois nós, *JACPPProfile* e *JDProfile* e, cada uma delas indica o ranking do nó relativamente aos critérios de pesquisa que lhe deram origem.

O ranking é calculado com base no *score* dado pelo SPARQL/Lucene ao efetuar a interrogação à ontologia. Quanto mais próximo o resultado está dos termos do par chave-valor, melhor é a pontuação. Para os casos de proximidade sintática, apenas é tido em conta o *score* do valor do par. É introduzido um fator de compensação, variável consoante a distância ao longo das relações da ontologia.

A equivalência é representada na figura 3.8 por uma linha a tracejado.

O código da listagem 3.5 contém um excerto de código onde é realizada a interrogação SPARQL DL sobre a ontologia. Ao ser executada a interrogação à ontologia, é tida em conta a indexação previamente realizada, não sendo necessário indicar que se pretende usar o índice. Desta forma, é possível obter o *score* do resultado da interrogação tanto para a chave, como para o valor.

```

public List<?> find(Object object) {
    SearchCriterionType searchCriterionType = (SearchCriterionType) object;
    List<GenericSearchResult> searchResultList = new ArrayList<GenericSearchResult>();
    String property = searchCriterionType.getCriterionName();
    String value = searchCriterionType.getCriterionValue();

    String q = StrUtils
        .strjoin(
            "\n",
            "PREFIX pf:      <http://jena.hpl.hp.com/ARQ/property#>",
            "PREFIX rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#>",
            "PREFIX owl:     <http://www.w3.org/2002/07/owl#>",
            "SELECT ?s ?p ?o ?p2 ?o2 ?scoreP ?scoreO2{",
            "  ?p <http://www.tese-mestrado/recruitment#propertyDescription> ?pd.",
            "    (?pd ?scoreP) pf:textMatch '" + property + "'.",
            "  {",
            "    ?s ?p ?o.",
            "    ?o ?p2 ?o2.", //property nameLiteral
            "    (?o2 ?scoreO2) pf:textMatch '" + value + "'.",
            "  }",
            "}"");

    Query query = QueryFactory.create(q);
    QueryExecution qexec = QueryExecutionFactory.create(q,
        dataAccess.getModel());
    try {
        ResultSet results = qexec.execSelect();
        for (; results.hasNext();) {
            /**
             * Process results
             */
        }
    } finally {
        qexec.close();
    }
    return searchResultList;
}

```

Listagem 3.5: Código para interrogação SPARQL DL

4

Implementação do Modelo

Neste capítulo é apresentada a implementação do modelo proposto.

Na implementação do modelo identificam-se duas fases principais:

- a) construção da base de conhecimento;
- b) exploração da base de conhecimento.

Nas secções seguintes estão descritos, com maior pormenor, os aspetos aqui abordados.

4.1 Representação de Conhecimento

Para representação de conhecimento, nesta área específica do recrutamento humano, foi criada uma ontologia OWL com base na especificação HR-XML Recruiting, de modo a representar os conceitos sobre a informação de procura e oferta.

Alguns conceitos e instâncias estão apresentados na seguinte *TBox* e *ABox*. A listagem completa da *TBox* e *ABox* encontra-se em anexo.

TBox = {Candidate, Profile, PositionOpening, Position, hasProfile, positionProfile }

ABox = {Candidate(JACPCandidate), Profile(JACPProfile), PositionOpening(PositionOpeningSoftOne1), Position(PositionSoftOne1),

```
hasProfile(JACPCandidate, JACPProfile),
positionProfile(PositionOpeningSoftOne1, PositionSoftOne1) }
```

A ontologia segue alguns *standards* e classificações, tais como a classificação para os níveis de educação, *International Standard Classification of Education* (ISCED) [36] e, as classificações sobre a indústria, *North American Industry Classification System* (NAICS) [1] e *German Classification of Industry Sector* (WZ2008) [5]. A ontologia contém sub-ontologias para estas classificações, de modo a melhorar e enriquecer a pesquisa contendo informação de mercados internacionais.

Os conceitos estão organizados em espaços de nomes ("*namespaces*"), de modo a separar e dar realce aos vários domínios de informação, o que permite, também, dividir a ontologia em sub-ontologias representativas dos diferentes *namespaces*. Na tabela 4.1 estão representados alguns dos *namespaces* da ontologia.

Prefix	Namespace	Domínio de Informação
	http://www.tese-mestrado/recruitment#	Base da ontologia
edu	http://www.tese-mestrado/education#	Informação sobre a educação
isic	http://www.tese-mestrado/economical-classification/international-standard/ISIC-rev4#	Informação sobre a classificação ISIC

Tabela 4.1: *Namespaces*: Diferentes domínios de informação

4.1.1 Propriedades de Equivalência Semântica

São criadas relações de equivalência entre propriedades e relações de semelhança entre instâncias, como por exemplo, no caso das classificações sobre a indústria ISIC e WZ2008.

Estas relações de equivalência estão representadas na figura 4.1 e são explicadas de seguida.

A propriedade que relaciona sujeitos com os recursos de ISIC é marcada como equivalente da propriedade que relaciona sujeitos com os recursos de WZ2008. Esta equivalência é definida com a propriedade OWL *equivalentProperty*. Desta forma, quando se pesquisa por uma classificação, ambas são tidas em conta no momento da pesquisa, sem ser necessário criar critérios de pesquisa específicos

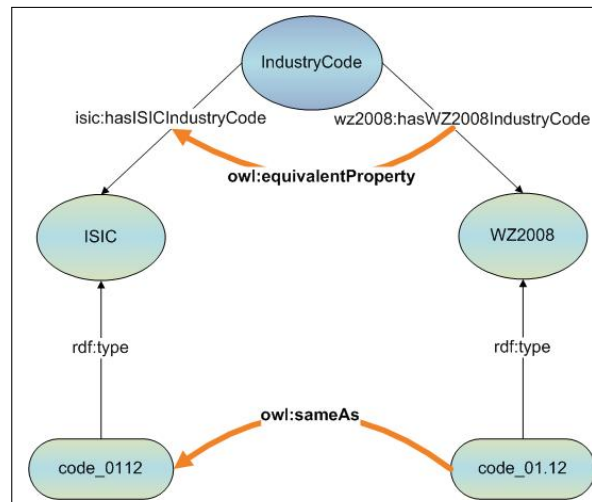


Figura 4.1: Representação das propriedades *owl:equivalentProperty* e *owl:sameAs*

para cada uma das classificações. Por exemplo, se interrogar a ontologia sobre relações *isic:hasISICIndustryCode*, serão também tidas em conta as relações *wz2008:hasWZ2008IndustryCode*.

As instâncias semelhantes são relacionadas com a propriedade OWL *sameAs*. Esta relação permite que as interrogações que incidam sobre uma determinada instância, tenham também em conta as instâncias semelhantes a esta. Por exemplo, numa interrogação à ontologia, ao ser obtido a instância *code_0112* de ISIC, é obtida também a instância *code_01.12* de WZ2008.

O mesmo exemplo pode ser dado para classificações ao nível do ensino, quer relativamente às habilitações literárias ou ao nível do ensino praticado por uma instituição.

4.1.2 Propriedades de Descrição

Foram introduzidas na ontologia duas novas propriedades, *propertyDescription* e *nameLiteral*, que têm um papel fundamental no contexto da pesquisa chave-valor e consequente interrogação à ontologia.

A propriedade *propertyDescription* é utilizada para atribuir descrição às propriedades, ou predicados. Desta forma, todas as propriedades podem ser pesquisadas pelo seu nome ou pelo que elas representam, isto é, a sua descrição. Na listagem 4.1 está indicada a propriedade e um exemplo de utilização desta, onde a propriedade *degreeName* tem a descrição "Education Degree" através desta propriedade.

```

<owl:DatatypeProperty rdf:ID="propertyDescription">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="http://www.tese-mestrado/education#degreeName">
  <rdfs:domain rdf:resource="http://www.tese-mestrado/education#EducationDegree"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <propertyDescription rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Education Degree
  </propertyDescription>
</owl:DatatypeProperty>

```

Listagem 4.1: OWL - Exemplo de utilização da propriedade *propertyDescription*.

A propriedade *nameLiteral* tem como objetivo representar o nome de um recurso em *string*. O Lucene indexa literais (*strings*), mas na ontologia nem todos os recursos são literais, logo nem todos os recursos podem ser indexados. No entanto é utilizada a propriedade *nameLiteral*, do tipo *string*, sobre os recursos, para representar o seu nome em literal. Deste modo todos os recursos passam a estar indexados através desta propriedade.

Na listagem 4.2 está indicada a propriedade e um exemplo de utilização desta, onde o recurso *MASTER* tem através da propriedade *nameLiteral* o seu nome em literal.

```

<owl:DatatypeProperty rdf:ID="nameLiteral">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

<edu_degree:DegreeType rdf:about="http://www.tese-mestrado/education#MASTER">
  <edu:nameLiteral rdf:datatype="http://www.w3.org/2001/XMLSchema#string"> MASTER
</edu_degree:DegreeTypeValue>
</edu_degree:DegreeType>

```

Listagem 4.2: OWL - Exemplo de utilização da propriedade *nameLiteral*.

4.2 Equivalência Semântica entre Oferta e Procura

O modelo desenvolvido para a pesquisa/equivalência assenta em associações atributo-valor. Cada um dos elementos da associação é um conjunto de termos, a partir do qual são encontradas correspondências, cujos termos (totais ou parciais) pertençam ao conjunto inicial.

4.2.1 Exemplo de Equivalência Semântica entre Oferta e Procura

Para o exemplo de equivalência é usado o mesmo exemplo ilustrado no subcapítulo 3.1.3, mas focado no modelo de equivalência semântica entre oferta e procura. As classes e propriedades usadas no exemplo são um subconjunto das classes e propriedades da ontologia, suficiente para a sua demonstração. As classes e propriedades estão representadas, de seguida, na forma de TBox e as instâncias na forma de ABox.

```
TBox = {NAICS, WZ2008, WZ2008_EN, ISIC, IndustryCode,
hasWZ2008ENIndustryCode, hasWZ2008IndustryCode, hasISICIndustryCode,
owl:equivalentProperty, owl:sameAs, owl:equivalentProperty(
hasWZ2008ENIndustryCode, hasWZ2008IndustryCode ),
owl:equivalentProperty( hasWZ2008IndustryCode, hasISICIndustryCode )}
```

```
ABox = {NAICS(code_111160), WZ2008(code_01.12), ISIC(code_1820),
ISIC(code_0112), IndustryCode( BMACPreferencesIndustry), IndustryCode(
JACPPreferencesIndustryCode ), IndustryCode( JDPreferencesIndustryCode),
sameAs(code_01.12, code_0112)}
```

As instâncias estão igualmente representadas num grafo OWL, na figura 4.2.

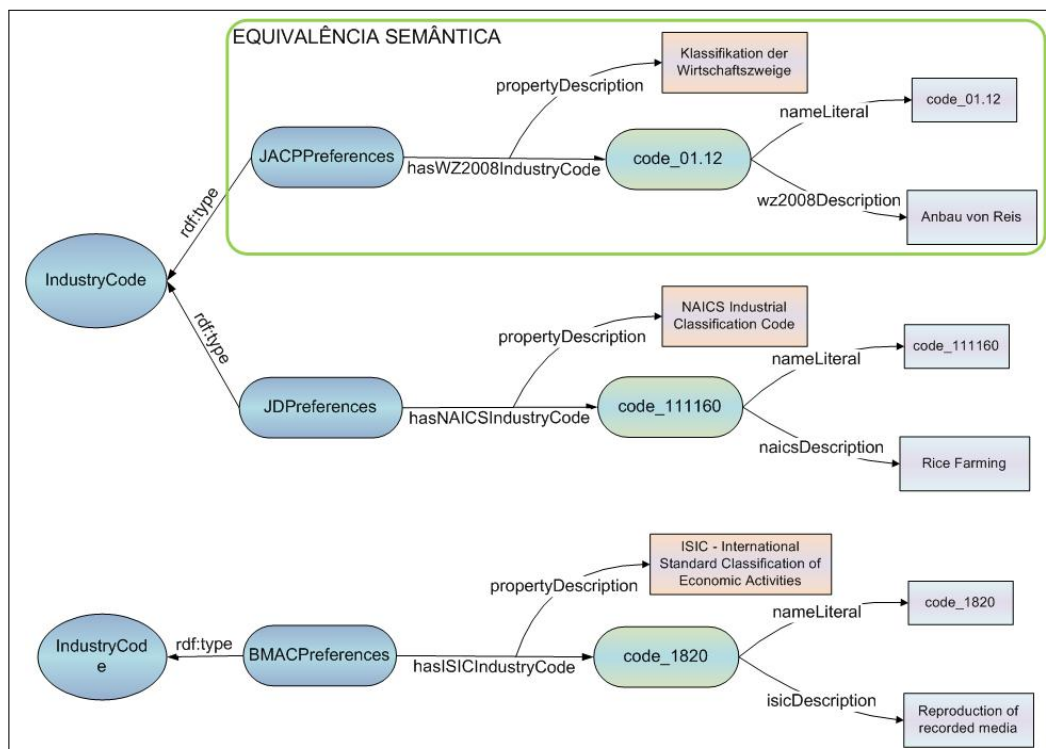


Figura 4.2: Exemplo de equivalência semântica

É importante recordar que *owl:equivalentProperty* tem o objetivo de relacionar semanticamente propriedades e *owl:sameAs* atribuir relações de igualdade entre instâncias.

Estas propriedades, bem como as classes que por elas estão relacionadas, são apresentadas na figura 4.3 num grafo OWL.

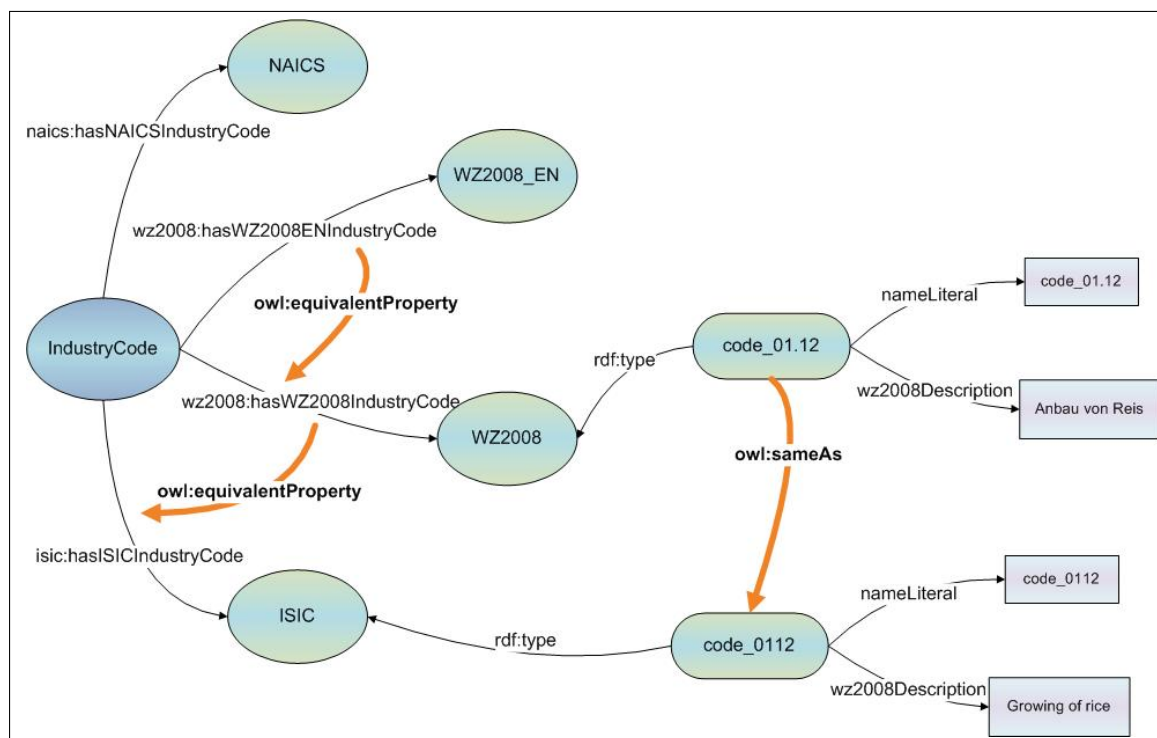


Figura 4.3: Representação das propriedades *owl:equivalentProperty* e *owl:sameAs* sobre o Código Industrial

Como critério de pesquisa é utilizado o seguinte par chave-valor:

chave:valor = [german economic activities]:[rice farming]

A figura 4.4 mostra a interface gráfica para esta pesquisa e o grafo com os resultados obtidos.

Os sujeitos devolvidos pela pesquisa são os que têm predicados cujo nome, ou descrição, têm pelo menos um termo do atributo (T_atributo) e, cujo objeto (domínio do predicado) tenha pelo menos um termo do valor (T_valor). No entanto, na ontologia da figura 4.2, não existe nenhuma propriedade cujos termos pertençam ao T_atributo e que tenham como domínios recursos cujos termos pertençam a T_valor. A propriedade *hasISICIndustryCode* tem, nos termos da sua descrição, correspondência aos termos da pesquisa, pelo que é uma propriedade a ter em conta. Com a igualdade semântica entre as propriedades, são tidas em conta as



Figura 4.4: Interface gráfica de pesquisa

hasWZ2008ENIndustryCode e *hasWZ2008IndustryCode*. Falta agora encontrar correspondências nos domínios destas propriedades. Encontra-se correspondência na instância *code_0112* da classe *ISIC*, que tem como termos "growing of rice". Como esta instância é igual (*owl:sameAs*) à instância *code_01.12* da classe *WZ2008*, então temos correspondência para o par atributo-valor. O elemento retornado é o *JACPPreferences*, obtido através de equivalência formal e DL.

Recorrendo à Logica de Descrição, combinada com as anotações semânticas, são devolvidos resultados semanticamente iguais.

4.2.2 Sugestão de Equivalência

Com o crescimento semântico da ontologia são adicionadas propriedades que podem ter significado semelhante a outras propriedades e, por isso, podem ter relevância para a pesquisa. Por exemplo, um utilizador tem um critério de pesquisa que incide sobre a classificação sobre a indústria e, ao ser adicionada uma nova classificação sobre a indústria, é interessante que esta possa ser alvo da pesquisa do utilizador, sem que este tenha de adicionar um novo critério de pesquisa para esta nova classificação.

A sugestão de equivalência é obtida em duas fases:

- a) através de uma combinação de proximidade sintática (entre literais);
- b) através da distância ao longo das relações da ontologia.

Na primeira fase são obtidos os recursos (objetos) que façam correspondência apenas com o *valor* do par atributo-valor. Estes recursos são sintaticamente próximos ao *valor*, obtidos com recurso ao LARQ.

Na segunda fase são obtidos os sujeitos (instâncias) que serão sugeridos como equivalentes. Estes sujeitos são todos aqueles que, partindo do recurso, façam parte das relações, encontradas recursivamente, até culminar em instâncias da mesma classe dos sujeitos iniciais, ou instâncias com *ancestor* comum. Desta forma, o predicado que está na base desta relação é o predicado equivalente. A listagem 4.3 contém um excerto do código para obtenção de equivalência. Com a *query* SPARQL (LARQ) obtêm-se os recursos equivalentes por proximidade sintática e, para cada um destes, é verificada a distância ao sujeito original.

```
public List<?> findEquivalent (Object object, Set<OntProperty> properties, Set<Resource>
    classTypeOfResourceMatched) {
    String value = ((SearchCriterionType) object).getCriterionValue();
    String q = StrUtils.strjoin(
        "\n",
        "PREFIX pf: <http://jena.hpl.hp.com/ARQ/property#>",
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>",
        "PREFIX owl: <http://www.w3.org/2002/07/owl#>",
        "SELECT ?s ?p ?o ?score02 ?s2 ?p2 ?o2{",
        "    ?s2 ?p2 ?o2.",
        "    (?o2 ?score02) pf:textMatch '" + value + "'.",
        "    ?s ?p ?s2.", //property nameLiteral
        "}" );

    Query query = QueryFactory.create(q);
    QueryExecution qexec = QueryExecutionFactory.create(q, dataAccess.getModel());

    try {
        ResultSet results = qexec.execSelect();
        for (; results.hasNext(); ) {
            QuerySolution soln = results.nextSolution();
            Resource resourceBase = soln.getResource("s");
            //Find distance beetwen concepts
            Integer distance = findDistanceBetweenResources(resourceBase,
                classTypeOfResourceMatched);
            ...
        }
    } finally {
        qexec.close();
    }
    return searchResultList;
}
```

Listagem 4.3: Código para obtenção de equivalência

Como exemplo de sugestão de equivalência vamos ter em conta a equivalência encontrada em 4.2.1, com a mesma ontologia e critério de pesquisa. De acordo com os critérios indicados para a sugestão de equivalência, os sujeitos sugeridos como equivalentes são aqueles que:

- têm relações com recursos que façam correspondência apenas com o valor do critério de pesquisa, e

- a instância pertence à mesma classe de qualquer sujeito do conjunto de resultados iniciais.

O único sujeito que satisfaz aquelas condições é o *JDPreferences*, evidenciado na ontologia da figura 4.5. Este sujeito é obtido através de proximidade semântica, pois a instância *code_111160* da classe *NAICS* tem os termos da pesquisa ("rice farming") e está relacionado com uma instância de uma classe obtida na pesquisa. Esta instância é a *JDPreferences*, associada através da propriedade *hasNAICSIndustryCode*.

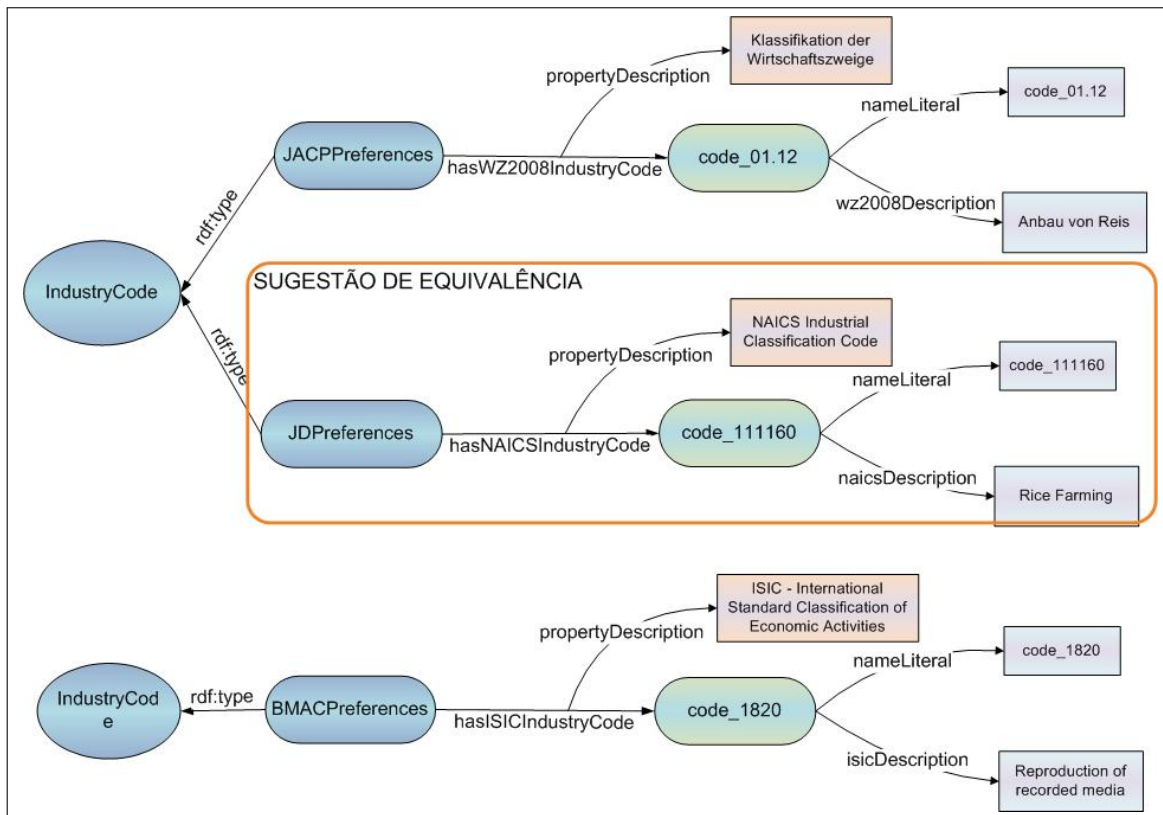


Figura 4.5: Exemplo de sugestão de equivalência

4.2.3 Ordenação do Resultado de uma Interrogação

Para cada sujeito (proposta/candidato) obtido é calculado o seu *ranking*, permitindo classificar os resultados quanto à interrogação. Desta forma, é possível indicar quais os resultados que estão mais próximos da interrogação efetuada. Para o cálculo do ranking, são utilizadas as pontuações do predicado e do sujeito, aquando da interrogação LARQ.

É realizado o produto entre ambas as pontuações, do predicado (p) e do sujeito (s), para cada critério da pesquisa. Esta pontuação é adicionada às obtidas anteriormente, de forma a obter o *ranking*. Este cálculo é realizado para cada candidato/oferta encontrado (baseado nos critérios de pesquisa).

Desta forma, o cálculo é realizado por:

$$Rank_j = \sum pp_i \times ps_i \quad (4.1)$$

Onde j representa um resultado da interrogação, pp_i a pontuação do predicado para cada critério e ps_i a pontuação do sujeito para cada critério.

A pontuação do predicado e do sujeito é atribuída pelo Lucene, através do modelo vetorial. Esta pontuação é acedida aquando da interrogação SPARQL, através de um parâmetro associado ao sujeito do predicado "*pf:textMatch*". A listagem 4.4 mostra um exemplo para obtenção da pontuação.

```
PREFIX pf: <http://jena.hpl.hp.com/ARQ/property#>
SELECT ?s ?p ?o ?score
{
  (?o ?score) pf:textMatch 'value'.
  ?s ?p ?o
}
```

Listagem 4.4: Pontuação de equivalência do Lucene

O cálculo do *ranking* para os conceitos obtidos por equivalência formal difere do cálculo para a equivalência por proximidade, uma vez que para esta tem-se em conta a distância entre os conceitos obtidos por grau de semelhança e os conceitos originais, obtidos formalmente. A pontuação do predicado também não é usada no cálculo, uma vez que só o valor do par chave-valor é utilizado na interrogação à ontologia. Para efetuar o cálculo, é retirada a pontuação do predicado e é introduzido um fator de degradação, pelo que, quanto maior for a distância entre conceitos, maior é o fator de degradação.

Desta forma, o cálculo é realizado por:

$$Rank_j = \sum ps_i \times fd(c_{ji}) \quad (4.2)$$

onde ps_i é a pontuação do sujeito para cada critério e $fd(c_{ji})$ é o fator de degradação, representado por:

$$fd(c_{ji}) = 1/(2^{d(c_{ji})}) \quad (4.3)$$

onde j representa um resultado da interrogação, i um critério da pesquisa e $d(c_{ji})$ é a distância entre ambos.

A distância é obtida recursivamente, partindo do conceito obtido e percorrendo as anotações ontológicas até culminar no conceito original.

4.3 Arquitetura

A figura 4.6 representa a arquitetura do trabalho, composta pela interface cliente, pelo servidor e repositório de informação. Foi criada uma interface de cliente, para realização de consultas e apresentação dos resultados obtidos. Foram seguidos os protocolos do *standard*, quer na comunicação da aplicação cliente com o servidor, quer na estrutura de dados utilizada na comunicação.

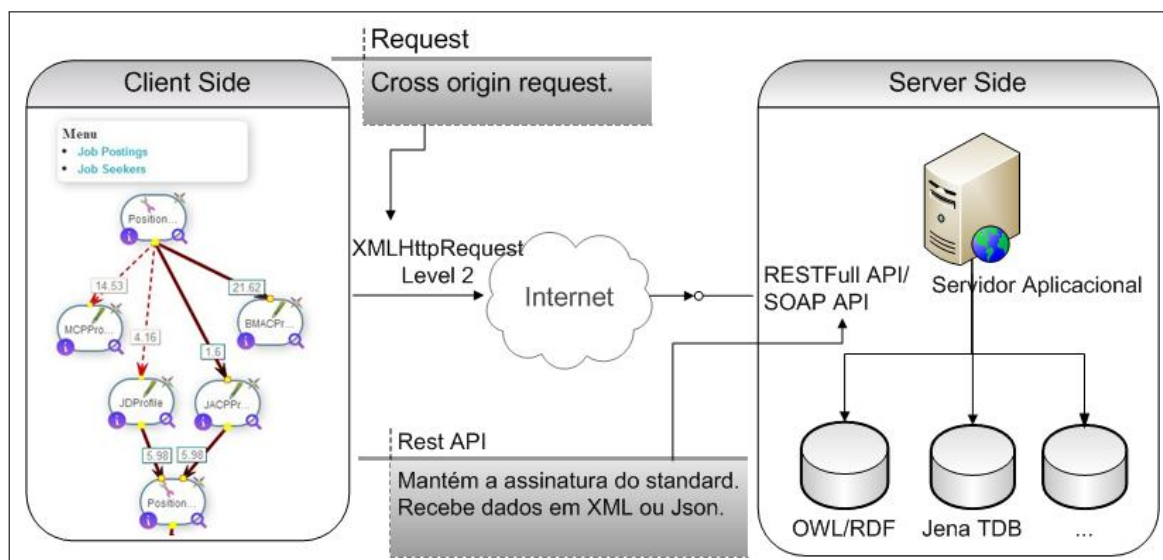


Figura 4.6: Arquitetura proposta

4.3.1 Interface Cliente

A aplicação cliente é realizada em HTML 5 [25], a comunicação com o servidor é realizada através da API *XMLHttpRequest Level 2* [40], para envio e recepção de dados. Os dados são enviados em formato JSON, construídos no cliente via *JavaScript* e enviados via POST. Os dados recebidos são também no formato JSON. A informação é renderizada num formato tipo "Grafo Social". Este tipo de apresentação permite visualizar de forma mais intuitiva as relações entre:

- candidato e propostas de emprego;
- candidatos que correspondam a algumas propostas de emprego;
- propostas de emprego que correspondam a candidatos a assim sucessivamente.

A figura 4.7 mostra um caso de utilização da camada de apresentação.

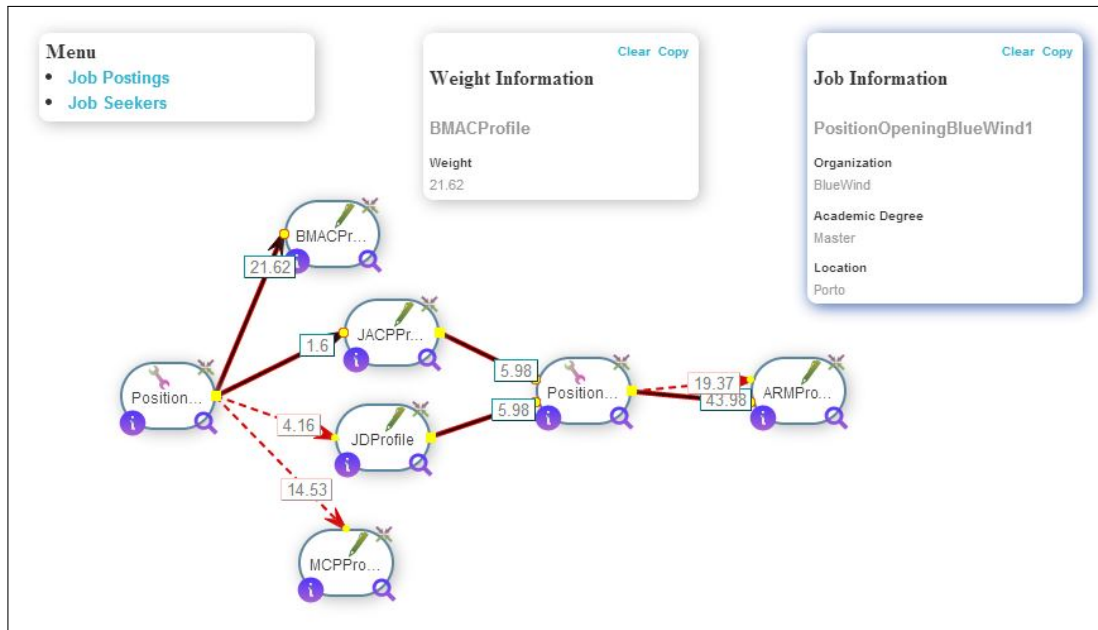


Figura 4.7: Camada de apresentação - Grafo Social

Cada proposta/candidato é representado por um elemento, composto por ícones que permitem distinguir candidatos de propostas, aceder à sua informação e pesquisar elementos que se relacionem com este.

Os elementos distinguem-se por um ícone central que caracteriza e diferencia propostas de candidatos. Disponibilizam acesso à informação sobre o candidato ou proposta. As propostas têm informação sobre a empresa e os candidatos têm informação pessoal e académica. Estes elementos estão representados na figura 4.8.

Cada elemento pode pesquisar outros elementos que se relacionem com este. Assim, é possível avançar ao longo do grafo, através das relações estabelecidas entre os elementos. O grafo não é estático, é dinâmico, é possível construir o grafo e obter novas relações entre os elementos, uma vez que, para cada pesquisa, é possível alterar os critérios de pesquisa que cada elemento tem por omissão. Desta forma, o grafo é construído de forma a responder às necessidades do utilizador.

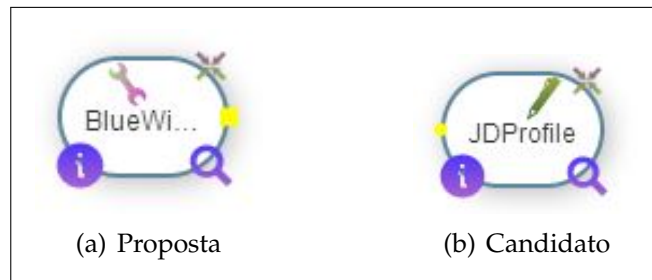


Figura 4.8: Representação visual de Proposta e Candidato

Os elementos do grafo são relacionados por dois tipos de linhas:

- **cheias** - representam as relações obtidas por equivalência semântica;
- **tracejadas** - representam as relações obtidas por equivalência por grau de semelhança.

Cada relação (linha) indica visualmente qual a classificação do elemento obtido, sendo possível, de imediato, perceber quais os elementos que melhor correspondem ao elemento original. Na figura 4.7 é possível distinguir os vários níveis do grafo bem como as relações entre eles - linhas cheias e tracejadas.

Os ícones da camada de apresentação são apresentados na tabela 4.2.








Símbolo	Descrição
	Proposta.
	Candidato.
	Aceder aos critérios de pesquisa do elemento.
	Aceder à informação do elemento: <ul style="list-style-type: none"> • Candidato - Informação pessoal sobre o candidato • Proposta - Informação sobre a empresa
	Remover as ligações aos restantes nós.
	Mostrar as ligações aos restantes nós.
	Pesquisar apenas os elementos sugeridos como equivalentes.

Tabela 4.2: Simbologia da camada de apresentação

4.3.2 Servidor

O servidor disponibiliza *Web Services* RESTfull, assentes numa arquitetura de software *Representational State Transfer* (REST). Apesar de seguir o *standard* do HR-XML, optou-se por esta arquitetura, em vez do protocolo SOAP, com envio e recepção de dados em JSON. O código é desenvolvido na linguagem de programação *Java* e é utilizado o *open source Jersey* [8], implementação de referência JAX-RS (Java API for RESTful Services) [18], para construção de *Web Services* RESTful. Através desta implementação de referência, os dados podem ser recebidos e enviados em formato JSON ou XML, cabendo essa decisão ao cliente que invoca os serviços. Do lado do servidor não existe esforço adicional na conversão de/para os formatos JSON e XML, uma vez que esta é realizada automaticamente. Para tal, basta indicar, através de anotações (*Consumes* e *Produces*) que explicitam quais os formatos que o método consome e produz. No código indicado na listagem seguinte, está representada a classe *Java* e a assinatura do método que dão origem ao serviço *GetSearchDocument*, e as respetivas anotações.

```
@Path("/searchDocument")
public class DocumentService {

    @POST
    @Path("GetSearchDocument")
    @Consumes({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML})
    public ShowSearchDocumentType getSearchDocument (JAXBElement<GetSearchDocumentType>
        searchDocumentParam) { ... }
}
```

Listagem 4.5: Classe Java e assinatura do método do serviço *GetSearchDocument*

A interrogação à ontologia é realizada em SPARQL, sendo usado um componente do framework Apache Jena, o ARQ, que permite fazer interrogações através de uma API *Java*.

Para a pesquisa e inferência é utilizado o *framework* Apache Jena, um *framework open source*, construído em Java, que disponibiliza várias API's para interação com grafos RDF, OWL, através de SPARQL, persitência de informação entre outras funcionalidades. Do Apache Jena, são usados os módulos ARQ, um motor de pesquisa compatível com a especificação SPARQL 1.1 e, o módulo LARQ que combina o ARQ com o Apache Lucene.

Desta forma o LARQ combina inferência, baseada na lógica de descrição, com proximidade sintática baseada no modelo vetorial.

O projeto Apache Lucene é um software *open source*, desenvolvido em Java, para

indexação e pesquisa textual, com grandes capacidades de análise textual (*analysis* e *tokenization*). Disponibiliza vários analisadores para várias línguas, mas tendo em conta a diversidade de línguas que podem ser usadas na ontologia, é utilizado o analisador sintático *standard*, *StandardAnalyser*. Este analisador cria *tokens* de acordo com as regras de quebra de palavras do algoritmo *Unicode Text Segmentation* especificado no *Unicode Standard Annex #29* [37]. Os tokens são convertidos para letras minúsculas e são eliminadas as *stop words* [16].

Os dados são armazenados e acedidos via ficheiro OWL e base de dados Jena TDB. Na fase inicial do projeto, foi definida na arquitetura o suporte do sistema Sesame (gestão de repositórios OWL) para armazenamento da ontologia. No entanto, ao utilizar alguns módulos do Apache Jena, verificou-se a existência do módulo TDB, vocacionado para o armazenamento e pesquisa de RDF. O TDB pode ser gerido inteiramente via API Java, o que facilita o processo de criação e gestão da base de conhecimento. Por estes motivos, o Jena TDB foi escolhido como base de conhecimento.

5

Validação e Testes

Neste capítulo são apresentados testes realizados à ontologia desenvolvida. Nos testes são obtidos propostas e candidatos, através de interrogações à ontologia. São apresentados resultados via equivalência formal, SPARQL-DL, e por grau de semelhança, proximidade sintática e distância ao longo das anotações ontológicas.

Nos testes que se seguem, são tidos em conta os seguintes pontos:

- Na pesquisa via equivalência formal, são obtidas, através de interrogação à ontologia, instâncias de conceitos que são sujeito do triplo (sujeito-predicado-objeto), cujo predicado corresponde à chave e o objeto corresponde ao valor do par chave-valor.
- Nos resultados obtidos via grau de semelhança, são considerados equivalentes as propostas/candidatos que tiverem na sua hierarquia elementos com proximidade sintática aos obtidos na equivalência formal. Para a interrogação é usada apenas o valor do par chave-valor. A instância obtida por equivalência formal é usada para calcular a distância entre esta e as que forem obtidas na interrogação.

A interrogação à ontologia retorna instâncias de conceitos que são sujeito do triplo (sujeito-predicado-objeto), cujo objeto corresponde ao valor do par.

5.1 Interrogação sobre Candidatos

Neste caso de teste pretende-se obter candidatos que correspondam ao seguinte par chave-valor:

chave:valor = [Education Degree]:[bachelor]

Os resultados obtidos formalmente e por grau de semelhança são descritos nas secções seguintes.

Resultados via Equivalência Formal

O triplete que descreve o resultado obtido pela interrogação está descrito de seguida.

Sujeito - `BMACDegreeUniversity`

instância de: `edu:EducationDegree`

Predicado - `edu:hasDegreeType`

descrição: `Education Degree Type`

pontuação (Lucene): 3,0689416

Objeto - literal

valor: Bachelor

pontuação (Lucene): 4,8986

A instância (`BMACDegreeUniversity`) obtida via equivalência formal não é uma instância do conceito que representa o candidato, mas sim uma instância que pertence à hierarquia de um candidato. São obtidas as pontuações do predicado e do objeto, atribuídas pelo Lucene aquando da interrogação. As pontuações permitem efetuar o cálculo do *ranking* do candidato obtido (ver cálculo em 4.1). O candidato e o *ranking*, calculado com as pontuações do predicado e do objeto, estão descritos de seguida.

Candidato - `BMACCandidate`

instância de: Candidate

ranking: $3,0689416 \times 4,8986 = 15,03$ (4.1)

Resultados via Grau de Semelhança

O triplete que descreve o resultado obtido pela interrogação está descrito de seguida.

Sujeito - MCPDegreeUniversity

instância de: edu:EducationDegree

Predicado - edu_it:IhasTTitoloAccademico

descrição: Titolo Accademico

Objeto - literal

valor: Laurea triennale

owl:sameAs: Bachelor

pontuação: 4.8986

distância: 0

O valor do objeto retornado não corresponde ao valor do par chave-valor ("bachelor"), no entanto é obtido pela interrogação. Isto é possível pois a conceito que representa "Bachelor" tem uma relação de igualdade (*owl:sameAs*) com o conceito que representa "Laurea triennale", assim, ao interrogar a ontologia o SPARQL-DL retorna este elemento por inferência. É obtida a pontuação atribuída pelo Lucene para o objeto e é obtida a distância ao longo das anotações da ontologia. Com a pontuação e a distância é efetuado o cálculo do *ranking* (ver cálculo em 4.2).

A distância da instância obtida por grau de semelhança à instância obtida formalmente é de 0, pois são instâncias da mesma classe. A instância de candidato a que pertence a instância obtida, bem como o *ranking* obtido estão descritos de seguida.

candidato - MCPCandidate

instância de: Candidate

ranking: $4,8986 \times 1/2^1 = 4,8986$ (4.2)

5.2 Interrogação sobre Propostas

Neste caso de teste pretende-se obter propostas que correspondem ao seguinte par chave-valor:

chave:valor = [Location]:[Lisbon]

Os resultados obtidos formalmente e por grau de semelhança são descritos nas secções seguintes.

Resultados via Equivalência Formal

O triplete que descreve o resultado obtido pela interrogação está descrito de seguida.

Sujeito - PositionSoftOne1

instância de: pos:Position

Predicado - pos:hasPositionLocation

descrição: Location

pontuação obtida: 5,304065

Objeto - literal

valor: Lisbon

pontuação: 5,304065

A instância obtida via equivalência formal não é uma instância do conceito que representa a proposta, mas sim uma instância que pertence à hierarquia de uma proposta específica. São obtidas as pontuações do predicado e do objeto, atribuídas pelo Lucene, aquando da interrogação. Através das pontuações é efetuado o cálculo do *ranking* (ver cálculo em 4.1). A proposta e *ranking*, calculado com as pontuações do predicado e do objeto estão descritos de seguida.

proposta - PositionOpeningSoftOne1

instância de: pos:PositionOpening

ranking: $5,304065 \times 5,304065 = 28,133$ (4.1)

Resultados via Grau de Semelhança

O triplete que descreve o resultado obtido pela interrogação está descrito de seguida.

Sujeito - PositionLocationTechPlus

instância de: pos:PositionLocation

Predicado - hasAddress

Objeto - literal

valor: Lisbon

pontuação: 5,304065

distância: 1

O objeto encontrado corresponde ao valor pesquisado ("Lisbon"), no entanto o sujeito não é instância da mesma classe da instância obtida via equivalência formal. Por este motivo, é verificada a distância entre ambos, verificando-se que a distância é 1. É atribuída, pelo Lucene, pontuação ao objeto e, juntamente com a distância, é calculado o *ranking* (ver cálculo em 4.2).

A instância de proposta a que pertence a instância obtida, bem como o *ranking* obtido estão descritos de seguida.

proposta - PositionOpeningTechPlus

instância de: pos:PositionOpening

ranking: $5,304065 \times 1/2^2 = 1,326$ (4.2)

5.3 Avaliação dos testes

Os testes realizados à ontologia, têm como intuito validar a interrogação à ontologia, os resultados obtidos e o cálculo do ranking destes.

Os conceitos e instâncias utilizados nos casos de teste, estão representados nas *TBox* e *ABox* no anexo A.

Em ambos os casos de teste foram obtidos resultados por equivalência formal e por grau de semelhança.

No primeiro caso de teste, onde foi efetuada uma interrogação à ontologia para obtenção de candidaturas, destaca-se a utilização da DL, na interrogação por grau de semelhança, para obtenção do objeto que tem correspondência com a chave. A chave (“bachelor”) tem correspondência no conceito de grau acadêmico *edu:BACHELOR*, no entanto este conceito não está diretamente relacionado com o objeto obtido. O conceito *edu_it_titulo:LAUREA_TRIENNALE* (versão italiana sobre graus acadêmicos) tem uma relação semântica com o conceito *edu:BACHELOR*, indicando que ambos têm o mesmo significado. Esta relação está indicada na figura 5.1, através do Protégé, onde constam os restantes tipos inferidos.

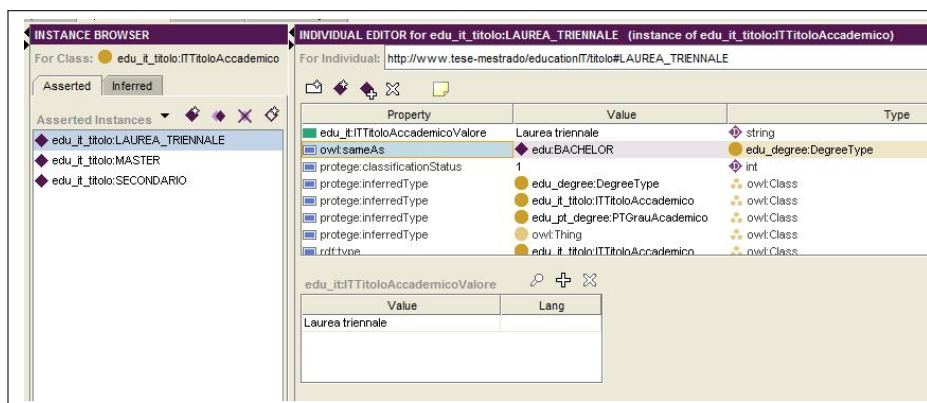


Figura 5.1: Tipos inferidos no Protégé

No segundo caso de teste, interrogação sobre propostas, destaca-se a validação da distância do conceito obtido via grau de semelhança ao conceito obtido via equivalência formal. Percorrendo as anotações da ontologia, verifica-se que fica a uma distância de 1, sendo 0 é a distância mínima, fica a uma iteração do conceito formal.

O cálculo do *ranking*, com a distância obtida, verifica-se que decai bastante, tendo em conta que a distância é pequena e que a pontuação obtida pelo Lucene é igual para ambos os objetos, quer por equivalência formal quer por grau de semelhança. Num trabalho futuro, a fórmula de cálculo (4.2) poderia ser revista, associando ao fator de degradação uma função logarítmica, de modo a que a degradação seja menos abrupta à medida que a distância aumenta.

Com a ontologia e o modelo proposto, foi possível criar casos de teste que respondem aos objetivos propostos para as pesquisas à ontologia.

6

Conclusões e Trabalho Futuro

Neste projeto foi implementado um sistema para descrição e inferência sobre oferta e procura na área de recursos humanos. O sistema tem por base uma ontologia para representação de conhecimento, onde foram descritas anotações semânticas, de modo a obter equivalências formais com recurso a interrogações SPARQL e à Lógica de Descrição (DL). A construção de um modelo vetorial associado à interrogação SPARQL permite obter resultados próximos dos critérios de pesquisa e, permite também, obter uma pontuação relativa à proximidade do literal encontrado com o critério de pesquisa. Esta pontuação é um dos critérios para estabelecer o ranking dos elementos obtidos por pesquisa.

Com a base de conhecimento que foi criada para validar o modelo, a utilização de SPARQL, DL e RI verificou-se adequada às interrogações efetuadas, tendo sido obtidos elementos provenientes de ambas as equivalências, formais (DL) e por proximidade semântica.

O modelo de apresentação dos resultados, num formato tipo grafo social, permite obter e relacionar facilmente os candidatos que se relacionam com propostas de trabalho e vice-versa. Os candidatos e propostas de trabalho são nós do grafo que podem ser expandidos sucessivamente.

Como trabalho futuro, o processo demonstrado para pesquisa de propostas de trabalho e de candidatos, pode ser estendido a outros processos, tais como a abertura de novas propostas de trabalho e disponibilização de novos candidatos, ou notificar automaticamente propostas quando surgem candidatos que correspondam

às suas características.

Relativamente à equivalência por grau de semelhança, para além de se obter as instâncias dos conceitos equivalentes, poderá ser mostrado ao utilizador as propriedades sugeridas como equivalentes, tendo este a possibilidade de as tornar efetivamente equivalentes, de modo a que futuras pesquisas tenham em conta os elementos que contêm estas propriedades. Desta forma passariam a ser obtidas via equivalência formal.

Bibliografia

- [1] U.S. Economic Classification Policy Committee (ECPC), Statistics Canada and Mexico's Instituto Nacional de Estadística y Geografía. NAICS: North American Industry Classification System @ONLINE, March 2013. URL <http://www.census.gov/eos/www/naics>.
- [2] Christian Bizer and Andreas Schultz. The berlin sparql benchmark. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(2):1–24, 2009.
- [3] J. Bruin. Newtest: command to compute new test @ONLINE, February 2011. URL <http://www.ats.ucla.edu/stat/stata/ado/analysis/>.
- [4] Chicago. *The Chicago Manual of Style*, pages 400–401. University of Chicago Press, thirteenth edition, 1982.
- [5] Classification of Economic Activities. WZ 2008: Classification of Economic Activities, Edition 2008 @ONLINE, March 2013. URL <https://www.destatis.de/DE/Methoden/Klassifikationen/GueterWirtschaftsklassifikationen/Content75/KlassifikationWZ08.html>.
- [6] HR-XML Consortium. HR-XML @ONLINE, November 2012. URL <http://www.hr-xml.org/>.
- [7] Oscar Corcho. *A layered declarative approach to ontology translation with knowledge preservation*, volume 116. Ios Press, 2005.
- [8] Oracle Corporation. Jersey - RESTful Web Services in Java @ONLINE, December 2012. URL <https://jersey.java.net/>.

- [9] Charles Louis Xavier Joseph de la Vallée Poussin, 1879. A strong form of the prime number theorem, 19th century.
- [10] Maryam Fazel-Zarandi and Mark S Fox. Semantic matchmaking for job recruitment: an ontology-based hybrid approach. In *Proceedings of the 8th International Semantic Web Conference*, 2009.
- [11] The Apache Software Foundation. ARQ @ONLINE, November 2012. URL <https://jena.apache.org/documentation/query/index.html>.
- [12] The Apache Software Foundation. Apache Jena @ONLINE, December 2012. URL <http://jena.apache.org/>.
- [13] The Apache Software Foundation. LARQ @ONLINE, November 2012. URL <https://jena.apache.org/documentation/larq>.
- [14] The Apache Software Foundation. Apache Lucene @ONLINE, December 2012. URL <http://lucene.apache.org/core>.
- [15] The Apache Software Foundation. TDB @ONLINE, November 2012. URL <http://jena.apache.org/documentation/tdb>.
- [16] The Apache Software Foundation. Lucene 4.4.0 demo API @ONLINE, January 2013. URL http://lucene.apache.org/core/4_4_0/demo/overview-summary.html.
- [17] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.
- [18] JAX-RS. Java API for RESTful Services (JAX-RS) @ONLINE, December 2012. URL <https://jax-rs-spec.java.net>.
- [19] jsPlumb. jsPlumb @ONLINE, March 2013. URL <http://jsplumbtoolkit.com>.
- [20] Donald E. Knuth. *The TeXbook*. Addison-Wesley, 1984.
- [21] Markus Krtzsch. *Description Logic Rules*. IOS Press, 2010.
- [22] K.S.Narendra and K.Parthsarathy. Identification and control of dynamical system using neural networks. *IEENN*, 1(1):4–27, 1990.
- [23] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 1986.

- [24] John Lovett. *Social media metrics secrets*, volume 159. Wiley. com, 2011.
- [25] Peter Lubbers, Brian Albers, and Frank Salim. *Pro HTML 5 Programming*. Apress0, 2010.
- [26] Alexander Maedche. *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers, 2002.
- [27] Malgorzata Mochol, Radoslaw Oldakowski, Ralf Heese, and Datenbanken und Informationssysteme. Ontology based recruitment process. In *GI Jahrestagung (2)*, pages 198–202, 2004.
- [28] United Nations. ISIC: International Standard Industrial Classification of All Economic Activities, Rev.4 @ONLINE, March 2013. URL <http://unstats.un.org/unsd/cr/registry/isic-4.asp>.
- [29] OAGi. OAGi @ONLINE, January 2013. URL <http://www.oagi.org/dnn2/>.
- [30] O'Reilly. *RESTful Web Services Cookbook*. Subbu Allamaraju, 2010.
- [31] Oren Patashnik. Using BibTeX. Documentation for general BibTeX users, January 1988.
- [32] Oren Patashnik. Designing bibtex styles. The part of BibTeX's documentation that's not meant for general users, January 1988.
- [33] M. Pfeffer. *E-recruitment: Analysis of Current Trends Based on an Empirical Survey Focussing on Online Career Networks*. VDM Publishing, 2008. ISBN 9783639032307. URL <http://books.google.pt/books?id=70G1PQAACAAJ>.
- [34] Jose Sandoval. *RESTful Java Web Services*. Packt Publishing, 2009.
- [35] William Strunk, Jr. and E. B. White. *The Elements of Style*. Macmillan, third edition, 1979.
- [36] UNESCO. ISCED: International Standard Classification of Education @ONLINE, March 2013. URL <http://www.uis.unesco.org/Education/Pages/international-standard-classification-of-education.aspx>.

- [37] Unicode, Inc. Unicode Standard Annex 29 @ONLINE, January 2013. URL <http://unicode.org/reports/tr29>.
- [38] Stanford University. Protege @ONLINE, February 2013. URL <http://protege.stanford.edu>.
- [39] Mary-Claire van Leunen. *A Handbook for Scholars*. Knopf, 1979.
- [40] W3C. XMLHttpRequest Level 2 @ONLINE, January 2013. URL <http://www.w3.org/TR/XMLHttpRequest2>.
- [41] W3C. OWL @ONLINE, February 2013. URL <http://www.w3.org/2004/OWL/>.
- [42] W3C. Resource Description Framework (RDF) @ONLINE, February 2013. URL <http://www.w3.org/2001/sw/wiki/RDF>.
- [43] W3C. RDF Vocabulary Description Language 1.0: RDF Schema (RDFS) @ONLINE, February 2013. URL <http://www.w3.org/2001/sw/wiki/RDFS>.
- [44] W3C. SPARQL 1.1 Query Language @ONLINE, January 2013. URL <http://www.w3.org/TR/sparql11-query>.
- [45] W3C. SPARQL-DL @ONLINE, January 2013. URL <http://www.w3.org/2001/sw/wiki/SPARQL-DL>.
- [46] W3C. SPARQL Query Language for RDF @ONLINE, January 2013. URL <http://www.w3.org/2001/sw/wiki/SPARQL>.
- [47] R. W. Zurek and L. J. Martin. Interannual variability of planet-encircling dust activity on Mars. 98(E2):3247–3259, 1993.



Representação de Conhecimento

Os conceitos e instâncias da ontologia estão apresentados na seguinte TBox e ABox.

TBox = {Candidate, Person, Profile, CandidateEmployeePreferences, EducationAttendance, EducationScore, EducationDegree, Period, EducationDegree, DegreeType, PTGrauAcademico, ITTituloAccademico, EducationOrganization, LevelCode, PTNivelEducacao, PositionOpening, Position, PositionDescription, PositionLocation, IndustryCode, ISIC, NAICS, WZ2008, WZ2008_EN, SearchCriteria, SearchCriterion, person, hasProfile, attendance, attendanceIn, hasDegree, hasScore, hasPeriod, hasDegreeType, hasPTGrauAcademico, hasLevelCode, hasPTNivelEducacao, hasEmployeePreferences, hasIndustryCode, hasISICIndustryCode, hasWZ2008ENIndustryCode, hasWZ2008IndustryCode, hasNAICIndustryCode, hasCriteria, hasCriterion, positionRequester, positionProfile, hasPositionLocation, positionFormattedDescription, hasAddress}

ABox = {Candidate(JACPCandidate), Candidate(BMACCandidate), Candidate(MCPCandidate), Person(JACP), Person(BMAC), Person(MCP), Profile(JACPPProfile), Profile(BMACProfile), Profile(MCPPProfile), CandidateEmployeePreferences(JACPCandidateEmployeePreferences), EducationAttendance(JACPAAttendance), EducationAttendance(BMACAttendance), EducationAttendance(MCPAttendance), EducationDegree(JACPDegreeUniversity),

EducationDegree(BMACDegreeUniversity),
 EducationDegree(MCPDegreeUniversity),
 EducationOrganization(EducationOrganizationISEL),
 EducationScore(JACPEducationScore), Period(JACPPeriodUniversity),
 EducationDegree(JACPDegreeUniversity),
 EducationDegree(BMACDegreeUniversity),
 EducationDegree(MCPDegreeUniversity), DegreeType(BACHELOR),
 PTGrauAcademico(MESTRE), ITTitoloAccademico(LAUREA_TRIENNALE),
 LevelCode(UNIVERSITY), PTNivelEducacao(UNIVERSIDADE),
 PositionOpening(PositionOpeningSoftOne1), Position(PositionSoftOne1),
 PositionDescription(PosDescriptionOpt_1),
 PositionDescription(PosDescriptionOptm_2),
 PositionLocation(PositionLocationSoftOne1),
 IndustryCode(JACPPreferencesIndustryCode), ISIC(code_1820),
 NAICS(code_111160), WZ2008(code_01.12), WZ2008_EN(code_01.12),
 SearchCriteria(SearchCandidateJACP),
 SearchCriteria(SearchPositionBlueWind1),
 SearchCriterion(CriterionEducationMaster),
 SearchCriterion(CriterionWZ2008ENRiceFarming),
 person(JACPCandidate, JACP), hasProfile(JACPCandidate, JACPProfile),
 attendance(JACPProfile, JACPAAttendance), attendanceIn(JACPAAttendance,
 EducationOrganizationISEL), hasDegree(JACPAAttendance,
 JACPDegreeUniversity), hasScore(JACPAAttendance, JACPEducationScore),
 hasPeriod(JACPAAttendance, JACPPeriodUniversity),
 hasPTGrauAcademico(JACPPeriodUniversity, MESTRE),
 hasLevelCode(EducationOrganizationISEL, UNIVERSITY),
 hasPTNivelEducacao(EducationOrganizationISEL, UNIVERSIDADE),
 hasEmployeePreferences(JACPProfile, JACPCandidateEmployeePreferences),
 hasIndustryCode(JACPCandidateEmployeePreferences,
 JACPPreferencesIndustryCode),
 hasWZ2008IndustryCode(JACPPreferencesIndustryCode, code_01.12),
 hasCriteria(JACPProfile, SearchCandidateJACP),
 hasCriterion(SearchCandidateJACP, CriterionEducationMaster),
 person(BMACCandidate, BMAC), hasProfile(BMACCandidate, BMACProfile),
 attendance(BMACProfile, BMACAttendance), attendanceIn(BMACAttendance,
 EducationOrganizationISEL), hasDegree(BMACAttendance,
 BMACDegreeUniversity), hasDegreeType(BMACDegreeUniversity),

BACHELOR),

person(MCPCandidate, MCP), hasProfile(MCPCandidate, MCPPProfile),
attendance(MCPPProfile, MCPAttendance), attendanceIn(MCPAttendance,
EducationOrganizationOXFORD), hasDegree(MCPAttendance,
MCPDegreeUniversity), hasITTitoloAccademico(MCPDegreeUniversity,
LAUREA_TRIENNALE),

positionRequester(PositionOpeningSoftOne1, PosOpeningRequesterSoftOne1),
positionProfile(PositionOpeningSoftOne1, PositionSoftOne1),
hasPositionLocation(PositionSoftOne1, PositionLocationSoftOne1),
positionFormattedDescription(PositionSoftOne1, PosDescriptionOptm_1),
positionFormattedDescription(PositionSoftOne1, PosDescriptionOptm_2),
hasAddress(PositionLocationSoftOne1, AddressSoftOne1) }

