



UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA

Mestrado em Engenharia Informática

Dissertação

**Framework de Pesquisas baseada numa aplicação Semantic
Web**

Nelson José Mendes Godinho

Orientadora

Prof. Doutora Irene Pimenta Rodrigues

2011

Mestrado em Engenharia Informática

Dissertação

**Framework de Pesquisas baseada numa aplicação Semantic
Web**

Nelson José Mendes Godinho

Orientadora

Prof. Doutora Irene Pimenta Rodrigues

Para a Carla, companheira de aventuras.

Prefácio

Este documento contém uma dissertação intitulada *Framework de pesquisas baseada numa aplicação Semantic Web*, um trabalho do aluno Nelson José Mendes Godinho¹, estudante de Mestrado em Engenharia Informática, na Universidade de Évora.

A orientadora deste trabalho é a Professora Doutora Irene Pimenta Rodrigues², do Departamento de Informática, da Universidade de Évora.

O autor é licenciado em Engenharia Informática, pela Universidade de Évora. A presente dissertação foi entregue em 20 de Setembro de 2011.

¹nelson.godinho@gmail.com

²ipr@di.uevora.pt

Agradecimentos

Após concluir com toda a dedicação e esforço mais uma etapa, quero deixar algumas palavras de apreço e gratidão a todas as pessoas que contribuíram para o sucesso deste trabalho.

Um especial agradecimento à minha família pelo apoio e pela formação que me deram e que ainda me dão. São dos principais responsáveis pela pessoa em que me tornei. Obrigado Pai, Mãe e Filipe.

Quero também agradecer à Professora Irene Rodrigues, minha orientadora, por todo o apoio e disponibilidade que ofereceu ao longo deste trabalho e que tornou o mesmo possível.

Um agradecimento especial para o Artur Romão e Bruno Carriço que sempre mostraram disponibilidade e me ajudaram na definição deste trabalho.

Agradeço também à Saphety pela disponibilização da informação sobre a plataforma estudada.

Não posso deixar de agradecer também a todos os amigos e colegas que me acompanharam e ouviram neste trabalho, Hugo, Nuno, Thiago, Mário, Bruno, Miguel e André.

Por fim, deixadas propositadamente para último, um grande mas grande agradecimento

às duas pessoas que sofreram com a minha ausência, as minhas preocupações e que sempre me deram o apoio necessário e motivação para avançar, Carla e António Duarte. No caso do António, apesar de ainda não ter nascido no momento em que estou a escrever este texto, acompanhou sempre o trabalho que desenvolvi na barriga da Mãe. Um grande beijo aos dois e muito Obrigado, neste momento são a minha razão de ser.

Muito Obrigado a Todos!

Sumário

Com o crescimento e aumento dos serviços *online* disponíveis lançados através do programa Simplex, surgiu em 2008 a primeira legislação para as Plataformas Electrónicas de Contratação Pública, cujo objectivo era desburocratizar o processo existente e, transmitir o máximo de transparência possível numa área que sempre levantou, e ainda levanta, dúvidas junto dos cidadãos Nacionais.

Por estarem apenas legisladas e, por não ter sido definido um modelo de dados único que fosse usado pelas diversas plataformas existentes, cada uma criou o seu modelo sendo que a integração entre sistemas é uma tarefa difícil e dispendiosa.

Com o presente trabalho criou-se uma ontologia que representa um procedimento de aquisição de bens e serviços, nomeadamente o Ajuste Directo – Regime Simplificado, e uma *framework* de pesquisas mais eficiente baseada na ontologia criada, recorrendo a *queries* utilizando linguagens de interrogação para Ontologias.

Palavras chave: ontologias, contratação pública, pesquisa semântica, sparql

Search Framework based on Semantic Web Application

Abstract

With the growth and increase of available online services launched through the Simplex program, the first legislation for Electronic Public Procurement Platforms appeared in 2008 , whose objective was to reduce bureaucracy in the existing process and convey as much transparency as possible in an area that has raised, and still raises, questions among nationals.

Despite this legislation, since no standard data representation model was defined, organizations have created their own data formats thereby making interoperability between these platforms a difficult and costly task.

The present work was created an ontology that represents a procedure of purchasing goods and services, namely Direct Adjustment - Simplified Regime, and a more efficient search framework based on this ontology using ontology queries language to retrieve results.

Keywords: ontologies, public procurement, semantic search, sparql

Conteúdo

Prefácio	i
Agradecimentos	iv
Sumário	v
Abstract	vii
Acrónimos	xviii
1 Introdução	1
1.1 Enquadramento e Motivação	1
1.2 Objectivos e Contribuições	3
1.3 Estrutura da Dissertação	4
2 Estado da Arte	5
2.1 Ontologia	5
2.1.1 Tipos de Ontologias	6
2.1.2 Componentes de uma Ontologia	7

2.1.3	Vantagens no uso de Ontologias	8
2.1.4	Ontologias para domínios específicos	9
2.1.5	Linguagens de descrição de Ontologias	10
2.1.6	Ferramentas de Construção e Manipulação	12
2.1.7	Metodologia para construção de uma Ontologia	14
2.2	SPARQL	18
3	Sistema proposto para pesquisas semânticas	21
3.1	Arquitectura	21
3.1.1	Plataforma	22
3.2	Ontologia	23
3.2.1	Listar termos relevantes	25
3.2.2	Definir taxonomia	26
3.2.3	Definir propriedades	29
3.2.4	Definir restrições	30
3.2.5	Definir Instâncias	30
3.3	Pesquisas	31
4	Usos do Sistema Proposto	33
4.1	Entidades Adjudicantes	33
4.2	Fornecedores	34
4.3	Interface gráfico	35
4.4	Testes efectuados	35
5	Conclusões	37
5.1	Trabalho futuro	38

<i>CONTEÚDO</i>	xi
Bibliografia	41
A Código da Ontologia	45

Lista de Figuras

2.1	Tipos de ontologias	7
2.2	Exemplo de uma Ontologia	8
2.3	Formalismos e Linguagens	10
2.4	Camadas das Linguagens para descrever Ontologias	11
2.5	Interface Gráfico do Protégé	14
3.1	Arquitetura do Sistema	22
3.2	Base de Dados de uma Plataforma de Contratação Pública	24
3.3	Taxonomia da Ontologia	28
4.1	Pesquisa por Procedimento	34
4.2	Pesquisa por Fornecedor	34
4.3	Pesquisa por Domínio	35
4.4	Classes e Restrições utilizadas na pesquisa	36

Lista de Tabelas

2.1	Vocabulário existente no OWL Lite e OWL DL/Full	12
3.1	Termos relacionados com entidades	25
3.2	Termos relacionados com pessoas	25
3.3	Termos relacionados com as equipas de procedimento	25
3.4	Termos relacionados com procedimentos	26
3.5	Classes Gerais	26
3.6	Classes Específicas das Entidades	27
3.7	Classes específicas dos colaboradores	27
3.8	Classes específicas do procedimento	27
3.9	Propriedades da classe procedure	29
3.10	Propriedades das restantes classes	29
3.11	Restrições da classe procedure	30
3.12	Restrições das restantes classes	30
3.13	Instâncias definidas	31
4.1	Resultados da Pesquisa por Évora	35

Acrónimos

ANC Agência Nacional de Compras

API *Application Programming Interface*

CEGER Centro de Gestão da Rede Informática do Governo

DAML *DARPA Agent Markup Language*

DAML+OIL *DARPA Agent Markup Language+Ontology Inference Layer*

DL Decreto-Lei

FOAF *Friend Of A Friend*

INCI Instituto da Construção e Mobiliário

OWL *Ontology Web Language*

RDF *Resource Description Framework*

RDFS *Resource Description Framework Schema*

SHOE *Simple HTML Ontology Extensions*

SPARQL *Sparql Protocol and RDF Query Language*

SQL *Structured Query Language*

URI *Uniform Resource Identifier*

W3C *World Wide Web Consortium*

XML *Extensible Markup Language*

Capítulo 1

Introdução

Neste primeiro capítulo é feita uma introdução sobre as áreas abrangidas nesta dissertação. Na secção 1.1 é apresentada a motivação e enquadramento do trabalho proposto. Na secção 1.2 os objectivos a que se propõe este trabalho, bem como as principais contribuições. E por fim, na secção 1.3 é apresentada a estrutura da dissertação.

1.1 Enquadramento e Motivação

Com a constante evolução tecnológica e, com o aumento de utilização por parte dos cidadãos Portugueses nos serviços *online*, por exemplo o *home banking* e as compras electrónicas, o Governo Português lançou em 2006 o SIMPLEX, um programa que consiste na Simplificação Administrativa e Legislativa que engloba um conjunto de iniciativas com o objectivo de simplificar os processos burocráticos com que os Cidadãos e Empresas se deparavam até aí.

A simplificação tem por objectivo melhorar a relação dos cidadãos com os serviços públicos, reduzir os custos de contexto das empresas no seu relacionamento com estes serviços, tornar a Administração pública mais eficiente e, assim, tornar Portugal mais competitivo. A estratégia de simplificação pode ser concretizada através de alguns objectivos genéricos:[8]

- Diminuir o número de atendimentos presenciais;

- Reduzir tempos de espera;
- Minimizar o número de interações relacionadas com o mesmo processo;
- Prestar serviços na hora;
- Dar mais e melhor acesso à informação.

Contudo e, apesar da iniciativa do Governo ter sido criada para melhorar o dia-a-dia dos seus cidadãos, existiam diversas áreas que ainda não eram abrangidas pelo programa e que continuavam a não ser transparentes. Uma das áreas era a da Contratação Pública de serviços ou bens. Para tentar resolver este problema surgiram em 2008 os primeiros Decretos-lei para que esta área ficasse o mais transparente possível. Surgiram assim as Plataformas de Contratação Pública. Estas plataformas permitem, a realização de procedimentos electrónicos públicos, bem como a aquisição electrónica de bens e serviços.

Estas soluções, enquadram-se no âmbito do DL n.o 18/2008 de 29 de Janeiro[2], do DL n.o 143-A/2008[1] de 25 de Julho e da Portaria 701-G/2008[3] de 29 de Julho. O DL n.o 18/2008 regula a formação e execução dos contratos públicos, definindo desta forma todos os procedimentos que decorrem desde o momento em que é tomada a decisão de contratar uma entidade até à adjudicação, assim como a execução do contrato. O DL n.o 143-A/2008 estabelece os princípios e regras gerais a que devem obedecer as comunicações, trocas e arquivo de dados e informações, previstos no Código dos Contratos Públicos, aprovado pelo Decreto -Lei n.o 18/2008, de 29 de Janeiro, em particular, a disponibilização das peças do procedimento, bem como o envio e recepção dos documentos que constituem as candidaturas, as propostas e as soluções. Por sua vez a Portaria 701-G/2008 de 29 de Julho define os requisitos e condições a que deve obedecer a utilização de plataformas electrónicas pelas entidades adjudicantes, na fase de formação de contratos públicos, e estabelece as regras de funcionamento daquelas plataformas.[25]

Devido a existirem inúmeras Plataformas Electrónicas de Contratação Pública, actualmente existem em Portugal 8 Plataformas certificadas pelo CEGER[7], foi necessário criar um Portal que reunisse toda a informação sobre os procedimentos criados em Portugal e, porque o Estado Português não quis “obrigar” os diversos organismos públicos a adoptar uma delas, surgiu assim o Portal de Contratos Públicos – BASE. No entanto, este portal apenas disponibiliza alguma das informações obrigatórias de publicação em Diário da República. Esta limitação surgiu porque cada uma das plataformas fez o seu entendimento da legislação e fez o seu próprio modelo de dados, não existindo no presente nenhum modelo único adoptado pelo estado Português.

Perante este quadro seria possível aprofundar o estudo na criação de um modelo que representasse um procedimento, vulgarmente conhecido como Concurso Público, e numa *framework* de pesquisas mais eficiente. Colocou-se o foco da dissertação nestas duas áreas.

Uma das formas de conseguir este objectivo, é recorrer ao uso de ontologias, que permitem estruturar e modelar conceitos e relações de um dado domínio, fornecendo uma base de conhecimento.

1.2 Objectivos e Contribuições

Na sequência do que foi descrito na secção anterior, o propósito da presente dissertação consiste na abordagem da criação de ontologias e pesquisas nas mesmas, em particular na área da Contratação Pública.

O objectivo principal deste trabalho é criar uma ontologia que permita representar um procedimento de aquisição de serviços e bens, neste caso particular o Ajuste Directo – Regime Simplificado, e uma *framework* de pesquisas mais eficiente.

A criação do modelo é conseguida através da análise da legislação Portuguesa e de um modelo de dados de uma das Plataformas de Contratação Pública existentes.

A *framework* de pesquisas é conseguida através da ontologia criada e, recorrendo a linguagens de interrogação, poderão ser feitas perguntas à ontologia como a que se segue:

- Devolver uma lista de domínios (procedimentos, entidades adjudicantes e fornecedores) onde exista a palavra pesquisada. Por exemplo, pesquisando por Évora, são devolvidos todos os procedimentos existentes nesse distrito, as entidades adjudicantes e fornecedores.

As principais contribuições apresentadas nesta dissertação são:

- Criação de uma ontologia que represente um procedimento de aquisição de serviços e bens do tipo Ajuste Directo – Regime Simplificado, e em particular:

Definição de classes, propriedades e restrições sobre as classes;

- Desenvolvimento de uma “camada” de pesquisas sobre a ontologia criada.

1.3 Estrutura da Dissertação

As estrutura da dissertação será desenvolvida da seguinte forma:

Capítulo 1

No presente capítulo é descrita a motivação e o enquadramento do problema a solucionar, assim como os seus objectivos. É apresentada a estrutura da dissertação e as suas principais contribuições.

Capítulo 2

Neste capítulo serão desenvolvidos os temas relacionados com o estado da arte em que se enquadra o trabalho apresentado nesta dissertação, as Ontologias.

Capítulo 3

Neste capítulo será apresentado o sistema proposto, dando atenção acrescida ao desenvolvimento de uma arquitectura baseada em dois componentes complementares: camada ontológica e camada de pesquisas.

Capítulo 4

Neste capítulo são efectuados testes à aplicação, para validar o cumprimento dos objectivos e é feita uma comparação com a solução de pesquisas actual.

Capítulo 5

Por fim, são apresentadas as principais conclusões atendendo aos objectivos que foram definidos no início do trabalho e, descrevem-se os problemas em aberto para uma investigação futura.

Capítulo 2

Estado da Arte

Este capítulo consiste numa introdução do estado da arte na área em que se enquadra o trabalho apresentado nesta dissertação.

A secção 2.1 apresenta o conceito de ontologias onde é descrita a sua definição, os vários tipos de ontologias existentes (2.1.1), as vantagens no seu uso (2.1.3), a sua utilização (2.1.4), incluindo o uso de ontologias na Contratação Pública (2.1.4), as linguagens que as descrevem (2.1.5), as ferramentas existentes (2.1.6) e as metodologias para construção de ontologias (2.1.7).

Na secção 2.2 é apresentada a linguagem de consulta para RDF, o Sparql. É feita uma descrição da linguagem, da estrutura das queries, da adição de restrições às mesmas e são apresentados alguns exemplos.

2.1 Ontologia

O termo “Ontologia” tem a sua origem nas palavras gregas *ontos* (ser) e *logos* (palavra). Apesar da antiguidade da sua origem, só na transição da Idade Média para a Idade Moderna, por volta dos séculos XVII e XVIII, se utilizou para designar um campo da Filosofia, o estudo da natureza da existência. Recentemente, à semelhança de outros termos, com os avanços na investigação na área das ciências da computação, recebeu um significado específico diferente do original [11] [4].

Na área das ciências da computação, a definição aceite pela maioria dos investigadores, foi enunciada por Gruber, sendo mais tarde reescrita por Studer:

“... is an explicit and formal specification of a conceptualization.”[4]

Por outras palavras, uma ontologia descreve formalmente um domínio através de um conjunto de conceitos representados num modelo de dados, ou seja, o tipo de objectos e/ou conceitos que existem e, as suas propriedades e relações [13].

O uso do termo ontologia tem sido referido como um meio para atingir a interoperabilidade de sistemas de informação, no que se refere à normalização terminológica. A possibilidade de partilhar o conhecimento foi a principal razão da utilização da ontologia no campo da Inteligência Artificial nos últimos anos. Ela é mais que um vocabulário padrão, pois assegura que os termos escolhidos sejam suficientes para especificar, definir conceitos e, permitir relações adequadas a partir da escolha terminológica realizada. É mais que uma taxonomia¹, pois inclui a expressão exacta do domínio específico do conhecimento.

2.1.1 Tipos de Ontologias

As ontologias não apresentam sempre a mesma estrutura, mas existem características e componentes básicos comuns presentes em grande parte delas. Mesmo apresentando propriedades distintas, é possível identificar tipos bem definidos. Segundo Guarino[14], as ontologias podem ser divididas em quatro tipos, de acordo com a generalização que pretendemos do domínio:

- **Ontologia de Alto Nível** - Descreve conceitos gerais como espaço, tempo, assunto, objecto, evento, acção, etc., os quais são independentes dum problema ou domínio específico;
- **Ontologia de Domínio/Tarefa** - Descreve o vocabulário relacionado com o domínio genérico (exemplo: medicina, automóvel), ou uma tarefa/actividade genérica (exemplo: diagnóstico, venda), especializando os termos introduzidos na ontologia de alto nível.

¹Teoria ou nomenclatura das classificações científicas.

- **Ontologia de Aplicação** - Descreve conceitos dependendo de um domínio e tarefa específicos, os quais são frequentemente especializações das ontologias relacionadas. Estes conceitos muitas vezes correspondem a papéis desempenhados por entidades do domínio enquanto executam uma determinada actividade.

Na figura 2.1, extraída de [14], pode-se constatar como os diferentes tipos de ontologias, de acordo com o nível de dependência de uma determinada tarefa ou ponto de vista, estão relacionados. As setas a negro representam as relações de especialização.

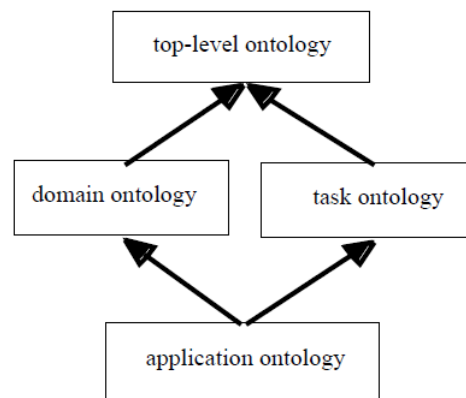


Figura 2.1: Tipos de ontologias

No trabalho apresentado nesta dissertação são utilizadas ontologias de domínio, pois o objectivo é partilhar o conhecimento sobre um domínio genérico, neste caso a Contratação Pública Portuguesa.

2.1.2 Componentes de uma Ontologia

Independente da linguagem em que são definidas, as ontologias partilham a mesma estrutura. Tipicamente, os componentes de uma ontologia são [26]:

- **Classe:** representa um conceito abstrato, podendo também descrever um conjunto, inclusivé de outras classes;
- **Instância:** é o componente básico da ontologia, o objecto de uma classe;

- **Atributo:** define uma característica da classe, e quando numa instância, possui um valor;
- **Relação:** exerce um importante papel para o enriquecimento semântico duma ontologia. As relações geralmente incluem uma hierarquia de classes.

Na figura 2.2, extraída de [22], é mostrado um exemplo de uma ontologia.

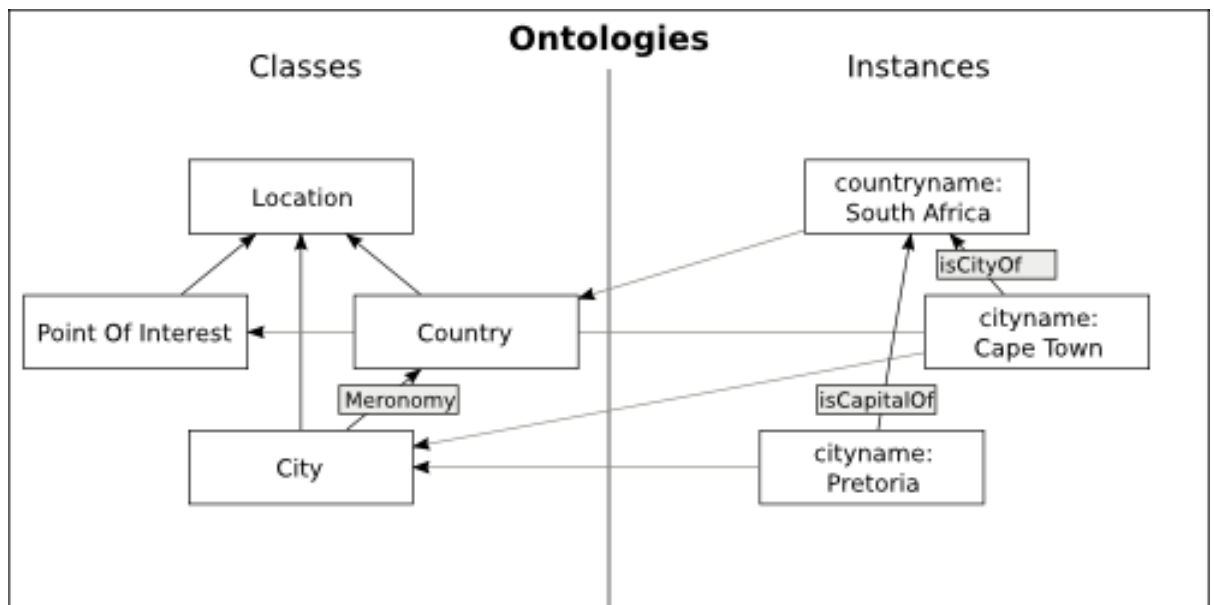


Figura 2.2: Exemplo de uma Ontologia

2.1.3 Vantagens no uso de Ontologias

A utilização de ontologias tem inúmeras vantagens na área das ciências da computação. De seguida, apresenta-se uma lista com as que mais se destacam [16]:

1. Fornecem um vocabulário para representação do conhecimento. Esse vocabulário tem uma especificação que o sustenta, evitando assim interpretações ambíguas do mesmo.

2. Permitem a partilha de conhecimento.
3. Fornecem uma descrição exacta do conhecimento. Contrariamente à linguagem natural, em que as palavras podem ter um significado totalmente diferente conforme o seu contexto, a ontologia, por ser escrita em linguagem formal, não deixa espaço para o *gap* semântico existente na linguagem natural.
4. É possível fazer o mapeamento da linguagem da ontologia sem que com isso seja alterada a sua especificação, ou seja, uma mesma especificação pode ser expressa em várias línguas.
5. Pode ser possível estender o uso de uma ontologia genérica de forma a que ela se adeque a um domínio específico.

2.1.4 Ontologias para domínios específicos

Sendo as Ontologias uma tecnologia relativamente recente, a sua utilização ainda não é massiva. Actualmente, começa-se a expandir a várias áreas, desde a pesquisa semântica, passando pela biomedicina, até às redes sociais. Os projectos Open-Source têm sido também um dos grandes impulsionadores na expansão das Ontologias, o projecto FOAF[9], um projecto que descreve pessoas e as relações entre elas, é um bom exemplo. Começaram também a surgir os primeiros motores de pesquisa, a maioria em versão Beta, sendo o Hakia [15], considerado um dos mais promissores. Existe também um número considerável de ontologias publicadas, em diversas áreas, que podem ser reaproveitadas pela comunidade [34]. A W3C, apresenta na sua webpage uma listagem de casos estudo da aplicação de ontologias em aplicações semânticas [32].

Ontologias na Contratação Pública

Na contratação pública, tanto a nível nacional como europeu, não se conhece o uso de ontologias. Em 2011 arrancou um projecto, o PEPPOL [29], que opera sobre o “ICT Policy Support Programme”[28] do Programa de Competitividade e Inovação da Comissão Europeia (DG-MARKT), tem por principal objectivo a criação de condições que facilitem a contratação pública transfronteiriça. Deste projecto, derivam essencialmente dois subprodutos: uma rede de intercâmbio de documentos e um conjunto de normas de

estruturação de documentos utilizados na contratação pública, devidamente suportado pela OASIS. Com a conclusão deste projecto poderão estar lançadas as bases para a definição de uma ontologia, a nível europeu, para a contratação pública.

2.1.5 Linguagens de descrição de Ontologias

Nos últimos anos têm aparecido diversas linguagens para representação de ontologias. As principais linguagens são: SHOE, DAML, DAML+OIL, OIL, Microformats, RDF, RDFS e OWL. Todas elas baseadas em formalismos. Na figura 2.3, extraída de [23] pode verificar-se a relação entre as linguagens e os formalismos.

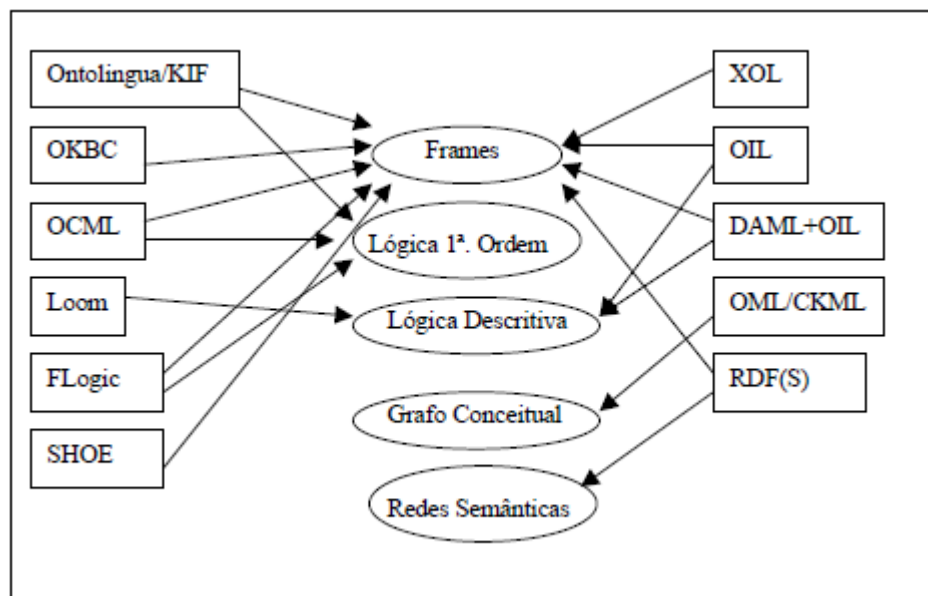


Figura 2.3: Formalismos e Linguagens

A partir da figura 2.4, extraída de [23], conclui-se que o XML é a base do desenvolvimento das linguagens SHOE, XOL e RDF, enquanto que para o OIL e DAML+OIL, é o RDF a base. Por fim, o OWL tem por base o DAML+OIL.

Nesta dissertação apenas se irá descrever o OWL por ser a linguagem proposta pelo W3C para linguagem padrão para construção de ontologias e, por ser a linguagem escolhida para o trabalho descrito nesta dissertação.

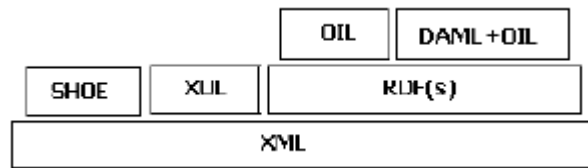


Figura 2.4: Camadas das Linguagens para descrever Ontologias

Owl

A OWL é uma das linguagens mais recentes. Esta linguagem é uma evolução das suas antecessoras RDF e DAML+OIL permitindo algumas especificações que não existiam nesta. A OWL foi desenvolvida com o intuito de permitir às aplicações processar a informação, em vez de apenas apresentá-la aos utilizadores.

A linguagem OWL permite uma maior interoperabilidade entre máquinas do que a suportada pelos seus antecessores XML, RDF e RDFS, pois fornece um vocabulário adicional para definição das classes e das suas propriedades e apresenta uma semântica formal. Esta linguagem fornece três sub-linguagens, cada uma oferece níveis de expressividade diferentes [19]:

- **OWL Lite** - Suporte aos utilizadores que necessitam de uma classificação hierárquica e restrições simples. Só permite valores de cardinalidade 0 ou 1 nas restrições.
- **OWL DL** - Suporte aos utilizadores que querem o máximo de expressividade, mantendo completude computacional (todas as deduções são computáveis). OWL DL inclui todo o vocabulário da linguagem OWL, mas com certas restrições (por exemplo, uma classe não pode ser uma instância de outra classe). OWL DL é assim chamado devido à sua correspondência com a lógica descritiva, um campo de pesquisa que estuda as lógicas que formam a base formal da OWL.
- **OWL Full** - Suporte aos utilizadores que querem usar a máxima expressividade sem quaisquer restrições na sintaxe do RDF sem necessitar de garantir a coerência e a completude das inferências.

OWL Full pode ser considerada uma extensão do RDF, enquanto OWL Lite e DL podem ser vistos como extensões de conjuntos restritos do RDF [19].

	OWL Lite	OWL DL/Full
Igualdades	X	X
Desigualdades	X	X
Propriedades	X	X
Restrições	X	X
Intersecções	X	X
Axiomas	-	X
Expressões Booleanas	-	X
Informação	-	X

Tabela 2.1: Vocabulário existente no OWL Lite e OWL DL/Full

Na tabela 2.1 pode-se verificar o vocabulário existente para o OWL Lite e para o OWL DL e OWL Full.

Os seus conceitos podem ser especificados por combinações lógicas como intersecção, união, ou complementos de outros conceitos, ou por enumerações de objectos especiais. Em OWL pode-se indicar que uma propriedade é transitiva, simétrica, funcional ou inversa, em relação a uma outra propriedade; quais indivíduos (instâncias) pertencem a quais conceitos; que conceitos e propriedades podem usar indicações de equivalência e disjunção; que indicações de igualdade e diferença podem ser utilizadas entre indivíduos, entre outras informações fundamentais para fornecer o suporte semântico necessário para os primeiros passos da Web Semântica.

Em 2009 surgiu a nova versão do OWL, o OWL2, que ainda está em fase de desenvolvimento, estando nesta altura como candidata a recomendação pela W3C.

2.1.6 Ferramentas de Construção e Manipulação

Por se tratar de uma tarefa morosa, qualquer apoio na construção/manipulação de ontologias, pode significar um aumento de performance importante. Actualmente existem inúmeras ferramentas disponíveis para esse propósito. Estas ferramentas permitem o desenvolvimento, inspecção, navegação e manutenção de ontologias, sendo que, na sua maioria, fornecem um interface gráfico de acordo com as normas existentes para o desenvolvimento de aplicações Web based [18].

Das ferramentas existentes podemos destacar as seguintes: Protégé, Jena, Bossam², Jess³, Racer⁴, Ontolingua⁵, WebONTO⁶, WebODE⁷, OntoEdit⁸, OilEd⁹, Apollo¹⁰, OntoSaurus¹¹, DOE¹², IsaViz¹³, SemTalk¹⁴, Icom¹⁵, OntoBuilder¹⁶, etc. Todas elas usam, normalmente, diferentes linguagens de representação, que foram descritas na secção 2.1.5. Nesta secção apresentamos detalhadamente o Protégé e o Jena por terem sido as escolhidas para o trabalho proposto nesta dissertação. A escolha recaiu nestas duas devido a serem as ferramentas mais utilizadas, aceites e referidas pela maioria dos utilizadores que criam e/ou manipulam ontologias.

Protégé [18]

Editor de ontologias, criado pelo departamento de informática médica da Universidade de Stanford, na Califórnia, Estados Unidos, está acessível e é gratuito para diversas plataformas. Desenvolvido em Java, possui um interface gráfico, ver figura 2.5, bastante rico que auxilia os utilizadores na construção de ontologias, permitindo a edição de classes, atributos, relações, entre outros. Por ser uma ferramenta *open-source*, permite a integração com diversas aplicações e com *plugins* de terceiros que adicionam funcionalidades extras que não estão acessíveis na versão base [20]. Pode-se mesmo concluir que a extensibilidade, é a sua grande vantagem quando em comparação com outras ferramentas. Com o Protégé é possível importar ou exportar ontologias de e para diversos formatos tais como RDF(s), XML, XML Schema e OWL.

Jena [17]

Framework desenvolvida em Java para construção de aplicações de Web Semântica criada pelos Hewlett-Packard (HP) Labs for Semantic Web Research. Esta API permite a manipulação de ontologias escritas em RDF, RDFS e OWL. Tem também um motor de

²<http://bossam.wordpress.com/>

³<http://www.jessrules.com/>

⁴<http://www.sts.tu-harburg.de/~r.f.moeller/racer/>

⁵<http://www.ksl.stanford.edu/software/ontolingua/>

⁶<http://projects.kmi.open.ac.uk/webonto/>

⁷<http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/downloads/60-webode>

⁸<http://www.ontoprise.de/en/products/ontostudio/>

⁹<http://oiled.semanticweb.org/building/>

¹⁰<http://apollo.open.ac.uk/>

¹¹http://ksi.cpsc.ucalgary.ca/KAW/KAW96/swartout/ontosaurus_demo.html

¹²<http://www.eurecom.fr/troncy/DOE/>

¹³<http://www.w3.org/2001/11/IsaViz/>

¹⁴<http://www.semtalk.com/>

¹⁵<http://www.inf.unibz.it/franconi/icom/>

¹⁶<http://ontobuilder.bitbucket.org/>

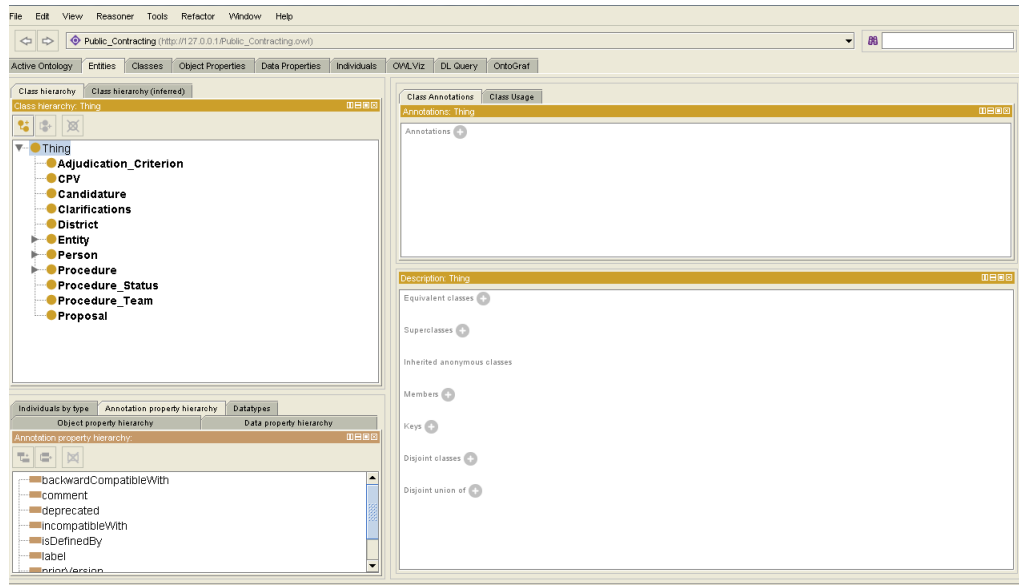


Figura 2.5: Interface Gráfico do Protégé

inferência para regras. Entre muitas funcionalidades contém um *parser* de RDF/XML e a API Jena 2 Ontology. Por ser uma API independente da linguagem, os nomes das classes Java não mencionam a linguagem subjacente. Por exemplo, a classe Java `OntClass` pode representar uma classe OWL, RDFS ou DAML. Para representar as diferenças entre as várias representações, cada uma das línguas da ontologia tem um perfil, que lista as construções autorizadas e os nomes das classes e propriedades. Assim, no perfil DAML, a URI para a propriedade `object` é `DAML:OBJECTPROPERTY`, no perfil OWL é `OWL:OBJECTPROPERTY` e no perfil do RDFS é nula porque em RDFS não se definem propriedades do `object` [17].

2.1.7 Metodologia para construção de uma Ontologia

O processo de desenvolvimento de ontologias ainda se encontra muito pouco desenvolvido, não existe consenso sobre a metodologia mais correcta. Existem variadas propostas disponíveis embora nenhuma delas esteja totalmente madura e, cada uma é aplicada pelo grupo que a criou. Das metodologias existentes poderemos destacar as seguintes [16]:

- Enterprise Ontology;
- Tove;

- Menthontology;
- On-To-Knowledge;
- Simple Knowledge-Engineering Methodology.

De seguida são enumerados os passos necessários para a construção de uma ontologia utilizando cada uma das metodologias citadas acima. A metodologia Simple Knowledge-Engineering Methodology será descrita detalhadamente visto que é a metodologia escolhida na elaboração do trabalho proposto nesta dissertação. A escolha recaiu sobre esta devido a ser uma das mais utilizadas e, por ser aquela que mais se aproxima do processo de desenvolvimento de software.

Enterprise Ontology - Uscholdand King's Method[30]

1. Identificar a proposta da ontologia
2. Construir a ontologia capturando, codificando e integrando conhecimento apropriado a partir de ontologias existentes
3. Avaliar a ontologia
4. Documentar a ontologia

TOVE - Toronto Virtual Enterprise[12]

1. Capturar cenários de motivação
2. Formular questões de competência informal
3. Terminologia formal
4. Formular questões em FOL

5. Especificar axiomas
6. Avaliar a ontologia

METHONTOLOGY[10]

1. Especificar o requisito
2. Conceptualizar o domínio do conhecimento
3. Formalizar o modelo conceptual numa linguagem formal
4. Implementar um modelo formal
5. Fazer a manutenção de ontologias implementadas

Actividades de suporte desempenhadas durante o processo de construção:

- Aquisição do conhecimento
- Integração
- Avaliação
- Documentação

On-To-Knowledge[27]

1. *Kick Off*
 - São capturados e especificados requisitos da ontologia
 - São identificadas questões de competência

- Ontologias potencialmente reutilizadas são estudadas e uma versão “draft” da ontologia é construída
2. Refinamento
 3. Avaliação
 4. Manutenção

Simple Knowledge-Engineering Methodology [21]

1. Âmbito - Trata-se do primeiro e mais importante passo. Uma ontologia pode ser considerada como um modelo dum domínio particular, construído para um propósito particular. Em consequência disto, não existe uma ontologia “correcta” para um domínio de conhecimento específico. Como para o desenvolvimento de uma ontologia é necessário a abstracção de um determinado domínio, é bastante comum existirem várias alternativas. As questões principais a serem pensadas neste passo são: qual o propósito e aplicação da ontologia; qual será o seu uso; para quais tipos de questões ela disponibilizará resposta; e quem irá usá-la e mantê-la.
2. Reutilização - É importante considerar a sua reutilização. Actualmente não é necessário começar o desenvolvimento de uma ontologia do “zero”, pode-se normalmente, aproveitar alguma ontologia já criada como ponto inicial para desenvolvimento de uma nova.
3. Listar termos relevantes - Após as definições iniciais, o próximo passo para o desenvolvimento de uma ontologia é listar todos os termos relevantes. Existem ferramentas que auxiliam o desenvolvimento desta etapa, a obter estes termos e uma estrutura inicial para eles.
4. Definir taxonomia - Quando os termos estiverem definidos, precisam ser organizados de forma a que se consiga mapear uma hierarquia. Existem divergências sobre qual forma é a mais eficiente, se é a *top-down*, *bottom-up* ou a mista. A abordagem *top-down* consiste em criar primeiro as classes mais gerais e depois ir descendo na hierarquia até às classes mais específicas. A abordagem *bottom-up* consiste no inverso, primeiro são criadas as classes mais específicas e só depois se sobe na hierarquia para a criação das classes mais gerais. Por fim, a abordagem mista, onde

são criadas as classes de vários níveis para no final serem organizadas numa hierarquia. A abordagem escolhida fica a cargo de quem estiver a desenvolver a ontologia.

5. Definir propriedades - Com as classes e as hierarquias já definidas é necessário definir as propriedades das classes. São as propriedades que irão contribuir para que as ontologias se aproximem dum modelo real, análogo ao pensamento humano. As propriedades possuirão valores ou restrições que definirão melhor a ontologia.
6. Definir restrições - Os tipos de valores que podem ser atribuídos às propriedades são definidos pela restrição de tipos. As restrições podem ser caracteres, booleanos, números, entre outros. A cardinalidade especifica quais os valores que uma propriedade pode ter.
7. Definir instâncias - Por fim, é altura de definir as instâncias da ontologia.

Todas estas metodologias de desenvolvimento estão relacionadas ao nível de linguagem natural, ou seja, uma linguagem que necessita ser compreendida pelo homem, onde serão feitas documentações, anotações, revisões e melhorias ao processo. É preciso no entanto que a ontologia seja também compreendida e decodificada pela máquina.

2.2 SPARQL

RDF é a base da Web Semântica. Espera-se que no futuro estejam disponíveis, cada vez mais, bibliotecas de dados RDF para consulta. Permitir fácil acesso a essas bibliotecas exige uma linguagem de consulta que seja capaz de lidar com esses dados [5].

No dia 15 de Janeiro de 2008, o *RDF Data Access Working Group* do W3C lançou uma linguagem de consulta para RDF chamada SPARQL [33], que é um acrónimo de **S**parql **P**rotocol and **R**DF **Q**uery **L**anguage.

Sparql pode ser usado para consultas em diversas fontes de dados, se os dados estiverem armazenados nativamente em RDF ou consultados como RDF via middleware[31]. Permite consultar grafos padrão obrigatórios e opcionais bem como conjunções e disjunções, existindo já implementação para diversas linguagens de programação [33].

Citando Sir Tim Berners-Lee: “*Sparql vai fazer uma enorme diferença*”¹⁷[24].

Estrutura das Queries

A maioria das *queries* Sparql contém um conjunto de padrões triplos chamados *basic graph pattern*. Cada parte do triplo actua como um assunto, um predicado, ou como um objecto que pode ser definido explicitamente como um recurso ou literal ou como variável. As variáveis Sparql têm o prefixo ? ou \$.

A linguagem Sparql disponibiliza 4 variações de queries para diferentes propósitos:

SELECT Query

Usada para extrair valores em bruto de um *endpoint* Sparql. Os valores são retornados numa tabela.

CONSTRUCT Query

Usada para extrair informações de um *endpoint* Sparql e transformar os resultados num RDF válido.

ASK Query

Usada para devolver um resultado simples de *True/False* para uma consulta num *endpoint* Sparql.

DESCRIBE Query

Usada para extrair um grafo RDF a partir de um *endpoint* Sparql.

Cada uma destas formas tem um bloco ***WHERE*** para restringir a pesquisa, embora no caso do *Describe* este seja opcional.

Exemplo de uma Query

A estrutura de uma *query* Sparql é muito semelhante à de uma query SQL. No exemplo seguinte pode-se ver como é representada uma query para devolver os nomes e emails de cada pessoa do mundo, desde que estejam presentes no domínio indicado:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?name ?email
```

¹⁷no original *SPARQL will make a huge difference*

```

WHERE {
  ?person a foaf:Person.
  ?person foaf:name ?name.
  ?person foaf:mbox ?email.
}

```

A cláusula **PREFIX** define o namespace do domínio a pesquisar. O **SELECT** especifica o que é devolvido, neste caso o nome e email, e por fim o **WHERE** define o grafo padrão para coincidir com os dados pretendidos. Os resultados devolvidos podem ser:

1. Um conjunto de grafos;
2. Um único grafo;
3. Nenhum grafo.

Adição de Restrições

Como vimos no exemplo acima, podem ser devolvidos vários resultados numa pesquisa. Para restringir o universo dos resultados o Sparql permite adicionar restrições. As restrições são feitas recorrendo à cláusula **FILTER**.

No exemplo seguinte pode-se verificar o uso da cláusula **FILTER** numa *query*.

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name ?email
WHERE {
  ?person a foaf:Person.
  ?person foaf:name ?name.
  ?person foaf:mbox ?email.
  FILTER regex(?name, \^Nelson").
}

```

Neste caso a *query* iria retornar todos as pessoas que contenham "Nelson" no nome e o respectivo email.

Existem diversas cláusulas no Sparql, tais como adicionar valores opcionais, o **Count** bem como o uso de *subqueries* dentro da cláusula **WHERE**. No entanto, nesta dissertação apenas descrevemos em pormenor aquelas que foram utilizadas para o trabalho desenvolvido.

Capítulo 3

Sistema proposto para pesquisas semânticas

Neste capítulo é apresentada um sistema que dá resposta aos objectivos propostos. Na secção 3.1 será apresentada a arquitectura, constituída por duas camadas principais: a camada ontológica e a camada de pesquisas (figura 3.1). A secção 3.2 apresenta a camada ontológica do sistema e, na secção 3.3 é apresentada a camada de pesquisas.

3.1 Arquitectura

A arquitectura do sistema proposto pode ser apresentada em duas camadas principais, uma ontológica e uma de pesquisas.

A camada ontológica pode ser vista como o core do sistema, é nela que está definida toda a semântica, através de uma ontologia, que representa um procedimento para aquisição de bens e serviços e as instâncias da mesma.

A camada de pesquisas fornece aos utilizadores a possibilidade de efectuarem pesquisas, neste caso, de acordo com os objectivos da dissertação, na camada ontológica. A figura 3.1 representa a arquitectura do sistema.

Este sistema foi desenhado para responder à legislação portuguesa DL n.º 18/2008 de 29 de Janeiro [2], do DL n.º 143-A/2008 [1] de 25 de Julho e da Portaria 701-G/2008 [3] de 29 de Julho, no que respeita à realização de procedimentos electrónicos públicos para aquisição de bens e serviços.

Nas próximas secções serão descritas individualmente e pormenorizadamente as camadas que fazem parte da arquitectura apresentada.

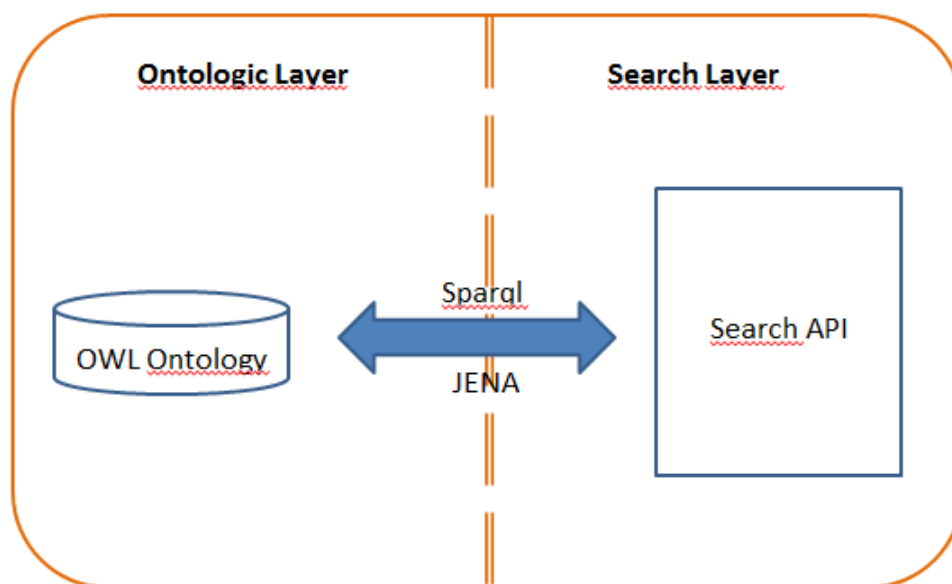


Figura 3.1: Arquitectura do Sistema

3.1.1 Plataforma

Todo o trabalho foi desenvolvido utilizando a plataforma Windows.

A criação da ontologia e consequentemente a produção do código OWL foi feita recorrendo à aplicação Protégé, por ser uma das ferramentas mais flexíveis, por causa dos plugins disponíveis e, porque segundo Cardoso [6] é um dos editores mais utilizados.

Para manipular a ontologia e executar as queries SPARQL foi utilizada a *framework* JENA. A ontologia foi publicada recorrendo ao *WebServer* Apache.

3.2 Ontologia

Como referido anteriormente, a base da camada ontológica é uma ontologia que representa um procedimento para aquisição de serviços e bens.

Para criar esta ontologia fez-se uma análise à legislação Portuguesa sobre procedimentos electrónicos de aquisição de bens e serviços e a uma plataforma electrónica de contratação pública, neste caso em particular o Bizgov. A figura 3.2 representa a base de dados simplificada da plataforma analisada. A metodologia utilizada para criar a ontologia é a *Simple Knowledge-Engineering Methodology* [21] (2.1.7) e utilizou-se o OWL como linguagem de representação, por ser mais eficiente e por ser a recomendação do W3C.

Seguindo a metodologia utilizada e assumindo como dado adquirido qual era o âmbito da ontologia e não existindo ontologias neste domínio (ver 2.1.4), avançamos directamente para o terceiro passo da metodologia, “**listar termos relevantes**”.

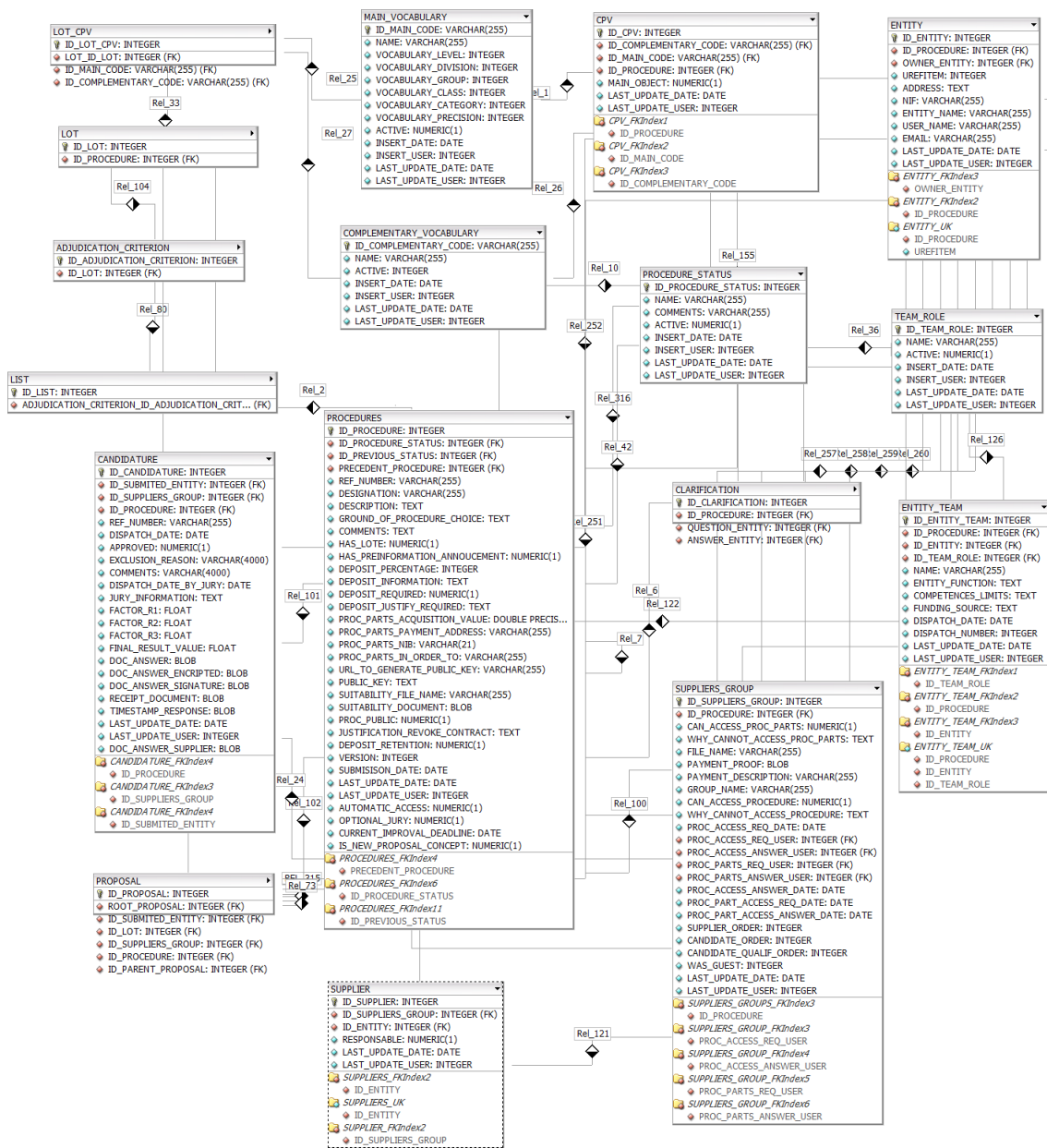


Figura 3.2: Base de Dados de uma Plataforma de Contratação Pública

3.2.1 Listar termos relevantes

Da análise efectuada anteriormente foram identificados os termos que se encontram representados nas seguintes tabelas.

A tabela 3.1 contém os termos relacionados com as entidades, na 3.2 estão representados os termos relacionados com as pessoas, a 3.3 contém os termos relacionados com as equipas e por fim a tabela 3.4 contém os termos relacionados com os procedimentos.

Termos de entidades
Nome
Morada
Número de Contribuinte

Tabela 3.1: Termos relacionados com entidades

Termos de pessoas
Nome
Email

Tabela 3.2: Termos relacionados com pessoas

Termos de equipas de procedimento
Presidente do Júri
Responsável pelos esclarecimentos
Board Member

Tabela 3.3: Termos relacionados com as equipas de procedimento

Termos de procedimentos
Tipo de Procedimento
Código CPV
Esclarecimentos
Propostas
Entidade Adjudicante
Fornecedor
Candidaturas
Critérios de Adjudicação

Tabela 3.4: Termos relacionados com procedimentos

3.2.2 Definir taxonomia

Depois de terem sido definidos todos os termos importantes, é necessário organizar os mesmos numa hierarquia. Para isso foram criadas as classes gerais representadas na tabela 3.5.

Classe	Descrição
Entity	Representa as entidades do procedimento
Procedure	Representa o procedimento
CPV	Representa o código CPV do procedimento
Adjudication Criterion	Representa os critérios de adjudicação
Candidature	Representa as candidaturas do procedimento
Clarifications	Representa os esclarecimentos do procedimento
District	Representa os Distritos das entidades
Person	Representa os colaboradores que fazem parte das equipas
Procedure Team	Representa a equipa do procedimento
Proposal	Representa as propostas do procedimento
Procedure Status	Representa o estado do procedimento

Tabela 3.5: Classes Gerais

Por fim foram definidas as classes específicas, representadas nas seguintes tabelas.

Classe	Descrição
Buyer	Representa a entidade adjudicante do procedimento
Supplier	Representa o fornecedor do procedimento

Tabela 3.6: Classes Específicas das Entidades

Classe	Descrição
Procedure Board Member	Representa os vogais do procedimento
Procedure Competent Jury for Clarification	Representa o responsável pelos esclarecimentos
Procedure President	Representa o Presidente do procedimento

Tabela 3.7: Classes específicas dos colaboradores

Classe	Descrição
Direct Adjustment	Representa o procedimento por ajuste directo
Normal Procedure	Representa o procedimento normal
Restricted By Prior Qualification	Representa o procedimento por qualificação prévia
Simplified Direct Adjustment	Representa o ajuste directo simples

Tabela 3.8: Classes específicas do procedimento

Na figura 3.3 apresenta-se a taxonomia da ontologia construída.



Figura 3.3: Taxonomia da Ontologia

3.2.3 Definir propriedades

Nesta etapa da metodologia definem-se as propriedades, focou-se essencialmente as relações entre as classes definidas no passo anterior.

Na tabela 3.9 estão definidas as propriedades para a classe Procedure.

Classe Domínio	Classe Intersecção	Relação
Procedure	Adjudication_Criterion	hasAdjudicationCriterion
Procedure	Buyer	hasBuyer
Procedure	CPV	hasCPV
Procedure	Clarifications	hasClarifications
Procedure	Procedure_Team	hasProcedureTeam
Procedure	Proposal	hasProposal
Procedure	Procedure_Status	hasStatus
Procedure	Supplier	hasSupplier

Tabela 3.9: Propriedades da classe procedure

Na tabela 3.10 encontram-se definidas as propriedades para as restantes classes:

Classe Domínio	Classe Intersecção	Relação
Supplier	Clarifications	askClarifications
Buyer	Clarifications	askClarifications
Procedure_Team	Procedure_Board_Member	hasBoardMember
Restricted By Prior Qualification	Candidature	hasCandidature
Procedure_Team	Procedure Competent Jury for Clarifications	hasProcedureCompetent Jury ForClarifications
Entity	District	hasDistrict
Procedure_Team	Procedure_President	hasPresident
Procedure	Procedure_Status	hasStatus
Procedure	Supplier	hasSupplier

Tabela 3.10: Propriedades das restantes classes

Para todas as propriedades foram definidas propriedades inversas em que a classe domínio passa a ser a classe intersecção e a classe intersecção passa a ser a domínio. Todas as

propriedades inversas foram definidas pelo termo "is....Of".

3.2.4 Definir restrições

O último passo da construção da ontologia passa pela definição das restrições entre classes, isto é, o tipo de valores que podem ser atribuídos às propriedades definidas.

Na tabela 3.11 estão definidas as restrições da classe procedure.

Classe Domínio	Restrição
Procedure	hasAdjudicationCriterion some Adjudication_Criterion
Procedure	hasBuyer exactly 1 Buyer
Procedure	hasCPV some CPV
Procedure	hasProcedureTeam exactly 1 Procedure_Team
Procedure	hasProposal some Proposal
Procedure	hasStatus exactly 1 Procedure_Status
Procedure	hasSupplier some Supplier

Tabela 3.11: Restrições da classe procedure

Na tabela 3.12 estão definidas as restantes restrições.

Classe Domínio	Classe Intersecção
Entity	hasDistrict some District
Restricted_By_Prior_Qualification	hasCandidature some Candidature
Procedure_Team	hasBoardMember max 3 Procedure_Board_Member
Procedure_Team	hasCompetentJuryForClarifications max 1 Procedure_Compentent_Jury_for_Clarifications
Procedure_Team	hasPresident exactly 1 Procedure_President

Tabela 3.12: Restrições das restantes classes

3.2.5 Definir Instâncias

O último passo da metodologia utilizada passa pela definição de instâncias da ontologia. Para efeitos de testes foram utilizadas as instâncias definidas na tabela 3.13.

Estas instâncias foram criadas através da base de dados de testes referida na secção 3.2 (pág. 23).

Procedimento	Ent. Adju.	Distrito da Ent. Adju.	Fornecedor
Procedure 1	Ent. Adju. Teste 1	Évora	Fornecedor Teste 1
Procedure 1	Ent. Adju. Teste 1	Évora	Fornecedor Teste 2
Procedure 2	Ent. Adju. Teste 2	Lisboa	Fornecedor Teste 1
Procedure 3	Ent. Adju. Teste 3	Porto	Fornecedor Teste 3
Procedure 3	Ent. Adju. Teste 3	Porto	Fornecedor Teste 4
Procedure 4	Ent. Adju. Teste 4	Évora	Fornecedor Teste 1
Procedure 4	Ent. Adju. Teste 4	Évora	Fornecedor Teste 4
Procedure 4	Ent. Adju. Teste 4	Évora	Fornecedor Teste 5
Procedure 5	Ent. Adju. Teste 5	Faro	Fornecedor Teste 6
Procedure 5	Ent. Adju. Teste 5	Faro	Fornecedor Teste 7

Tabela 3.13: Instâncias definidas

3.3 Pesquisas

Esta camada tem o propósito de fornecer aos utilizadores a possibilidade de efectuarem pesquisas na ontologia que obedeçam aos objectivos definidos para esta dissertação.

Para construir esta camada recorreu-se à linguagem SPARQL (pág. 18) e foi definida a seguinte querie, conforme os objectivos definidos.

Objectivo: Devolver a lista de domínios (procedimentos, entidades adjudicantes e fornecedores) onde exista a palavra pesquisada, definida pela seguinte sintaxe:

```
PREFIX mydomain: <http://localhost/Public_Contracting.owl>

SELECT DISTINCT ?proc
WHERE
{
  ?proc a mydomain:Procedure;
    mydomain:hasBuyer ?buyer;
    mydomain:hasSupplier ?supplier.
  ?supplier mydomain:hasDistrict ?supplierDistrict.
```

```

?buyer mydomain:hasDistrict ?buyerDistrict.
FILTER (?supplierDistrict = 'String Pesquisada' || ?buyerDistrict = 'String Pesquisada')
}

```

Com esta querie são retornados todos os procedimentos que contenham a palavra pesquisada. Depois basta percorrer esta lista e listar todas as entidades adjudicantes e fornecedores recorrendo às propriedades *has_buyer* e *has_supplier*.

É possível obter os mesmos resultados recorrendo às diversas funcionalidades que a *Framework* JENA disponibiliza. Esta abordagem apesar de ser mais complexa, porque tem de se ter um conhecimento elevado da ontologia e das suas propriedades e restrições, possibilita aos programadores mais liberdade e controlo na manipulação da informação pretendida. O código para devolver os objectivos propostos é o seguinte:

```

Model model;
Resource supplierClass = model.getResource(MY_NS + "Supplier");
Resource buyerClass = model.getResource(MY_NS + "Buyer");
Resource procClass = model.getResource(MY_NS + "Procedure");
Property districtProp = model.getProperty(MY_NS + "District");
Property hasBuyerProp = model.getProperty(MY_NS + "hasBuyer");
Property hasSupplierProp = model.getProperty(MY_NS + "hasSupplier");
StmtIterator iter =
    model.listStatements(new SimpleSelector(null, districtProp, "String Pesquisada"));
while (iter.hasNext()) {
    Resource subject = iter.next().getSubject();
    if (!subject.hasProperty(RDF.type))
        continue;
    Resource subjectClass = subject.getPropertyResourceValue(RDF.type);
    SimpleSelector sel;
    if (subjectClass.equals(supplierClass))
        sel = new SimpleSelector(null, hasSupplierProp, subject);
    else if (subjectClass.equals(buyerClass))
        sel = new SimpleSelector(null, hasBuyerProp, subject);
    else
        continue;
    StmtIterator innerIter = model.listStatements(sel);
    while (innerIter.hasNext()) {
        Resource proc = innerIter.next().getSubject();
        if (!proc.hasProperty(RDF.type) ||
            !proc.getPropertyResourceValue(RDF.type).equals(procClass))
            continue;
    }
}
}

```


Capítulo 4

Usos do Sistema Proposto

Neste capítulo é apresentada a utilização do sistema proposto no capítulo 3 para obter os resultados esperados nos objetivos definidos. É também feita a comparação, em três perspectivas diferentes (Entidade Adjudicante, Fornecedores e Interface Gráfico), entre o sistema apresentado e o sistema existente numa plataforma de contratação pública.

4.1 Entidades Adjudicantes

As entidades adjudicantes utilizam essencialmente as plataformas para criarem novos procedimentos de aquisição de bens e serviços e, para pesquisarem por procedimentos similares aos que pretendem para poderem à partida ter uma ideia dos custos envolvidos, fornecedores interessados, duração dos serviços, etc.

Na plataforma estudada, as entidades adjudicantes têm duas formas de pesquisa, por procedimento e por fornecedor (figuras 4.1 e 4.2). Pela forma como estas pesquisas estão implementadas, apenas a pesquisa por fornecedores consegue devolver uma lista-gem de fornecedores por distrito, não sendo possível retornar todos os procedimentos por determinado distrito conforme a solução apresentada nesta dissertação.

Pesquisa de procedimentos

Nº do procedimento:

Designação:

Ordenar por:

Tipo de procedimentos:

Estado:

Data de publicação do procedimento:

Data de publicação em DR:

Data da Abertura de propostas:

Figura 4.1: Pesquisa por Procedimento

Pesquisa de fornecedores

Entidade:	NIPC:	Localidade:	Email:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Resultados da Pesquisa

Entidade	NIPC/NIF	Contactos
Nenhuma entidade encontrada		

Figura 4.2: Pesquisa por Fornecedor

4.2 Fornecedores

A utilização da plataforma por parte dos fornecedores cinge-se apenas à consulta para concorrer a procedimentos. Um fornecedor para conseguir encontrar procedimentos que possam ser interessantes para si apenas dispõe de duas possibilidades: pesquisa e visualização da lista de novos procedimentos. A última hipótese, do ponto de vista da performance, é desaconselhável visto que em média são criados, diariamente, cerca de 50 novos procedimentos na plataforma estudada e, na primeira hipótese o fornecedor apenas tem disponível para pesquisa os campos representados na figura 4.2. Com a solução apresentada (ver figura 4.3), o fornecedor tem ao seu dispor uma importante mais-valia, a pesquisa por distrito, que lhe devolve a listagem de todos os procedimentos abertos para um determinado distrito.

Pesquisa por Domínio

Domínio:

Figura 4.3: Pesquisa por Domínio

Procedimento	Ent. Adju.	Distrito da Ent. Adju.	Fornecedor
Procedure 1	Ent. Adju. Teste 1	Évora	Fornecedor Teste 1
Procedure 1	Ent. Adju. Teste 1	Évora	Fornecedor Teste 2
Procedure 4	Ent. Adju. Teste 4	Évora	Fornecedor Teste 1
Procedure 4	Ent. Adju. Teste 4	Évora	Fornecedor Teste 4
Procedure 4	Ent. Adju. Teste 4	Évora	Fornecedor Teste 5

Tabela 4.1: Resultados da Pesquisa por Évora

4.3 Interface gráfico

No interface gráfico encontra-se uma das mais-valias da solução proposta. Actualmente, alterar o interface gráfico do formulário de criação de um procedimento é uma tarefa manual e dispendiosa, com a agravante deste formulário ser alterado sempre que a legislação que rege a contratação pública electrónica é reformulada. Com a solução proposta, a criação deste formulário poderá ser automática, percorrendo a ontologia criada, bastando para isso alterar a ontologia sempre que a legislação é alterada. Esta funcionalidade diminui o tempo de implementação, e consegue-se assegurar que sempre que existem alterações estas são reflectidas em toda a plataforma.

4.4 Testes efectuados

Depois de ter sido definida a ontologia, as respectivas instâncias e implementada a camada de pesquisas foram efectuados testes para comprovar a validade dos resultados obtidos.

Pesquisa por "Évora":

Utilizando a querie definida em 3.3, e pesquisando pela string "Évora" foram obtidos os resultados enunciados na tabela 4.1.

Estes resultados foram obtidos recorrendo à Ontologia, mais propriamente às classes *Pro-*

cedure, *Buyer*, *Supplier* e *District* e pelas relações *hasBuyer*, *hasSupplier* e *hasDistrict* definidas conforme ilustrado na figura 4.4.

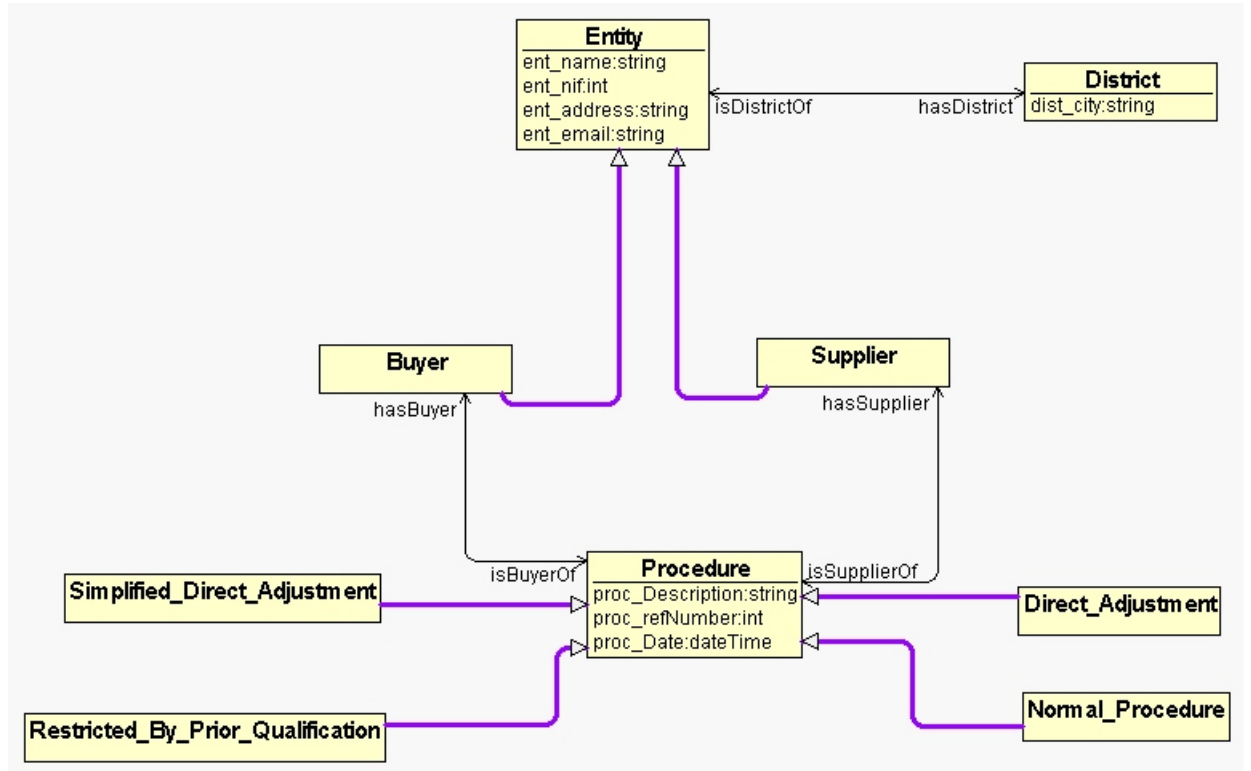


Figura 4.4: Classes e Restrições utilizadas na pesquisa

Nota: Foram obtidos os mesmos resultados recorrendo às funcionalidades da *Framework* Jena.

Capítulo 5

Conclusões

Neste capítulo será apresentada uma reflexão sobre o trabalho apresentado ao longo desta dissertação. Na secção 5.1 são enumerados aspectos que podem ser vistos como oportunidade de melhoria ou como pistas trabalho futuro.

Com a implementação do sistema proposto, apresentado no capítulo 3, os objectivos previstos no capítulo 1.2, foram de um modo global atingidos.

Este sistema faz uso de uma ontologia que representa um procedimento de aquisição de bens e serviços e de uma *framework* de pesquisas sobre a ontologia definida.

Tratando-se de um trabalho de investigação, por não existir nada implementado nesta área específica, foi necessário procurar várias hipóteses para se conseguir chegar à solução final. Nem sempre a solução estudada foi a melhor, sendo por isso necessário recuar e procurar novos caminhos, em parte devido a não existir uma única forma de definir ontologias.

A ontologia criada envolve um conjunto de classes, propriedades e restrições e está representada na linguagem OWL. Esta ontologia foi criada de raiz, pois como referido anteriormente, não existem trabalhos desenvolvidos neste âmbito.

A *framework* de pesquisas foi pensada para responder a questões que não estão disponíveis na plataforma de contratação pública actual. Estas questões são uma mais valia para os seus futuros utilizadores, visto que até aqui apenas era possível pesquisar por procedimentos ou por fornecedores. Com a solução proposta, os utilizadores conseguirão pesquisar por um domínio e obter resultados indexados a vários recursos; procedimentos, entidades adjudicantes e fornecedores. Esta *framework* foi implementada recorrendo ao Sparql e/ou às funcionalidades fornecidas pelo Jena.

Conforme referido anteriormente, este sistema foi implementado para cumprir a legislação portuguesa para procedimentos electrónicos de aquisição de bens e serviços.

5.1 Trabalho futuro

Existem diversos pontos que podem ser vistos como oportunidades de melhoria ou como trabalho futuro, designadamente:

1. Efectuar perguntas do tipo:
 - (a) Devolver a lista de concursos abertos com base naqueles onde já concorreu, recorrendo ao CPV;
 - (b) Devolver a lista de concursos abertos por entidades adjudicantes daqueles onde já concorreu.
2. Expandir a ontologia aos restantes tipos de procedimentos existentes.
3. Integrar esta solução no ambiente de produção e adaptar o *layout* gráfico ao mesmo.
4. Integrar uma candidatura ao QREN em conjunto com a INCI e com a ANC para se poder definir uma ontologia única que seja seguida por todas as plataformas electrónicas portuguesas.

Bibliografia

- [1] Decreto-lei n° 143-a/2008, 2008. Disponível em: <http://dre.pt/pdf1sdip/2008/07/14301/0000200006.PDF>. Último acesso: 14/08/2011.
- [2] Decreto-lei n° 18/2008, 2008. Disponível em: <http://dre.pt/pdf1s/2008/01/02000/0075300852.pdf>. Último acesso: 14/08/2011.
- [3] Portaria 701-g/2008, 2008. Disponível em: <http://www.base.gov.pt/codigo/Documents/Portaria701G2008.pdf>. Último acesso: 14/08/2011.
- [4] Gregoris Antoniou and Frank Harmelen. *A Semantic Web Primer*. The MIT Press, Cambridge, Massachusetts and London, England, 2003.
- [5] Hanief Bastian. Faceted exploration of multiple rdfdata sources using sparql. Master's thesis, University of Duisburg-Essen, Duisburg, May 2009.
- [6] Jorge Cardoso. The semantic web vision: Where are we? *IEEE Intelligent Systems*, 22(5):84–88, 2007.
- [7] Base Portal de Contratos Públicos Online. Plataformas de contratação pública certificadas, 2011. Disponível em: <http://www.base.gov.pt/plataformaselectronicas/Paginas/plataformascertificadas.aspx>. Último acesso: 14/08/2011.
- [8] Presidência do Concelho de ministros. Orientações para a simplificação, 2009. Disponível em: <http://www.simplex.pt/downloads/orientacoesideiasimplex.pdf>. Último acesso: 14/08/2011.
- [9] Friend of a friend project, 2010. <http://www.foaf-project.org/>.

- [10] A. Gómez-Pérez. Knowledge sharing and reuse. In J. Liebowitz, editor, *Handbook of Expert Systems*. CRC, 1998.
- [11] Paulo Gomes and Fernando Gama. Ontologias - arquitetura de informação digital, 2009. Disponível em: <http://www.slideshare.net/loncarovich/ontologia>. Último acesso: 12/01/2011.
- [12] M. Grüninger and M. S. Fox. Methodology for the design and evaluation of ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conduction with IJCAI-95*, Montreal, Canada, 1995.
- [13] Tom Gruber. Ontology, 2009. Disponível em: <http://tomgruber.org/writing/ontology-definition-2007.htm>. Último acesso: 12/01/2011.
- [14] Nicola Guarino. Formal ontology and information system. In *Proceedings of FOIS 98*, pages 3–15, Amsterdam, 1998. IOS Press.
- [15] Semantic search, 2010. <http://www.hakia.com>.
- [16] Verlani Hinz. Proposta de criação de uma ontologia de ontologias. Master's thesis, Universidade Católica de Pelotas, Pelotas, Brasil, Dezembro 2006.
- [17] Jena 2 ontology api, 2010. Disponível em <http://jena.sourceforge.net/ontology/>. Acedido a 29 de Julho de 2011.
- [18] Andreia Malucelli. *Ontology-based Services for Agents Interoperability*. PhD thesis, Departamento de Engenharia Electrotécnica e de Computadores - Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2006.
- [19] Deborah L. McGuinness and Frank van Harmelen. . web ontology language - overview. w3c recommendation, 2004. Disponível em: <http://www.w3.org/TR/owl-features>. Último acesso: 12/06/2011.
- [20] N. F. Noy, R. W. Ferguson, and M. A. Musen. The knowledge model of Protege-2000: Combining interoperability and flexibility. In *2nd International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, Juan-les-Pins, France, 2000.
- [21] Natalya F. Noy and Deborah L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. Online, 2001.
- [22] Current state of the semantic web, 2008. http://kore-nordmann.de/blog/current_state_of_semantic_web.html.

- [23] Simone Prado. *UM EXPERIMENTO NO USO DE ONTOLOGIAS PARA REFORÇO DA APRENDIZAGEM EM EDUCAÇÃO A DISTÂNCIA*. PhD thesis, Universidade de São Paulo, São Paulo, Brasil, 2004.
- [24] Reuters. Berners-lee looks for web's big leap. 2006.
- [25] Saphety. Bizgov - plataforma de contratação pública, 2008. Disponível em: <http://www.bizgov.pt/conteudos/default.asp?ID=23&IDP=8&P=8>. Último acesso: 14/08/2011.
- [26] Thiago Silva. Estudo de integração entre ontologia de domínio e processos de negócio - acessando protótipo através do aris. Master's thesis, Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, Brasil, Novembro 2009.
- [27] S. Staab, R. Studer, H.-P. Schnurr, and Y. Sure. Knowledge processes and ontologies. 16, 2001.
- [28] EU European Union. Ict policy support programme, 2007. Disponível em: http://ec.europa.eu/information_society/apps/projects/index.cfm?prog_id=IPSP. Último acesso: 16/08/2011.
- [29] EU European Union. Peppol - pan european public procurement online, 2008. Disponível em: <http://www.peppol.eu>. Último acesso: 16/08/2011.
- [30] Mike Uschold and Martin King. Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, Montreal, Canada, 1995.
- [31] W3C. Sparql query language for rdf, 2008. Disponível em: <http://www.w3.org/TR/rdf-sparql-query/>. Último acesso: 16/08/2011.
- [32] Semantic web case studies and use cases, 2010. <http://www.w3.org/2001/sw/sweo/public/UseCases/>.
- [33] Wikipédia. Sparql, 2008. Disponível em: <http://en.wikipedia.org/wiki/SPARQL>. Último acesso: 16/08/2011.
- [34] Published ontologies examples, 2010. [http://en.wikipedia.org/wiki/Ontology_\(information_science\)#Examples_of_published_ontologies](http://en.wikipedia.org/wiki/Ontology_(information_science)#Examples_of_published_ontologies).

Anexos

Anexo A

Código da Ontologia

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY Public_Contracting "http://127.0.0.1/Public_Contracting.owl#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://127.0.0.1/Public_Contracting.owl#"
  xml:base="http://127.0.0.1/Public_Contracting.owl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:Public_Contracting="http://127.0.0.1/Public_Contracting.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about="http://127.0.0.1/Public_Contracting.owl"/>

  <!--
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  //
  // Datatypes
  //
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  -->
```

```

<!--
//
// Object Properties
//
//
-->

<!-- http://127.0.0.1/Public_Contracting.owl#askClarifications -->

<owl:ObjectProperty rdf:about="&Public_Contracting;askClarifications">
  <rdfs:domain rdf:resource="&Public_Contracting;Buyer"/>
  <rdfs:range rdf:resource="&Public_Contracting;Clarifications"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Supplier"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasAdjudicationCriterion -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasAdjudicationCriterion">
  <rdfs:range rdf:resource="&Public_Contracting;Adjudication_Criterion"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasBoardMember -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasBoardMember">
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_Board_Member"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure_Team"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasBuyer -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasBuyer">
  <rdfs:range rdf:resource="&Public_Contracting;Buyer"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasCPV -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasCPV">
  <rdfs:range rdf:resource="&Public_Contracting;CPV"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;isCPVOf"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasCandidature -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasCandidature">
  <rdfs:range rdf:resource="&Public_Contracting;Candidature"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Restricted_By_Prior_Qualification"/>

```

```

    <owl:inverseOf rdf:resource="&Public_Contracting;isCandidatureOf"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasClarifications -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasClarifications">
  <rdfs:range rdf:resource="&Public_Contracting;Clarifications"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasCompetentJuryForClarifications -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasCompetentJuryForClarifications">
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_Competent_Jury_for_Clarifications"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure_Team"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;isCompetentJuryForClarificationsOf"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasDistrict -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasDistrict">
  <rdfs:range rdf:resource="&Public_Contracting;District"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Entity"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;isDistrictOf"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasPresident -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasPresident">
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_President"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure_Team"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasProcedureTeam -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasProcedureTeam">
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_Team"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasProposal -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasProposal">
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:range rdf:resource="&Public_Contracting;Proposal"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasStatus -->

```

```

<owl:ObjectProperty rdf:about="&Public_Contracting;hasStatus">
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_Status"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#hasSupplier -->

<owl:ObjectProperty rdf:about="&Public_Contracting;hasSupplier">
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:range rdf:resource="&Public_Contracting;Supplier"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;isSupplierOf"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isAdjudicationCriterionOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isAdjudicationCriterionOf">
  <rdfs:domain rdf:resource="&Public_Contracting;Adjudication_Criterion"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;hasAdjudicationCriterion"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isBoardMemberOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isBoardMemberOf">
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure_Board_Member"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_Team"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;hasBoardMember"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isBuyerOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isBuyerOf">
  <rdfs:domain rdf:resource="&Public_Contracting;Buyer"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;hasBuyer"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isCPVOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isCPVOf">
  <rdfs:domain rdf:resource="&Public_Contracting;CPV"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isCandidatureOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isCandidatureOf">
  <rdfs:domain rdf:resource="&Public_Contracting;Candidature"/>
  <rdfs:range rdf:resource="&Public_Contracting;Restricted_By_Prior_Qualification"/>
</owl:ObjectProperty>

```



```

<!-- http://127.0.0.1/Public_Contracting.owl#isClarificationOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isClarificationOf">
  <rdfs:domain rdf:resource="&Public_Contracting;Clarifications"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;hasClarifications"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isCompetentJuryForClarificationsOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isCompetentJuryForClarificationsOf">
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure_Competent_Jury_for_Clarifications"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_Team"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isDistrictOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isDistrictOf">
  <rdfs:domain rdf:resource="&Public_Contracting;District"/>
  <rdfs:range rdf:resource="&Public_Contracting;Entity"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isPresidentOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isPresidentOf">
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure_President"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_Team"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;hasPresident"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isProcedureTeamOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isProcedureTeamOf">
  <rdfs:range rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure_Team"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;hasProcedureTeam"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isProposalOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isProposalOf">
  <rdfs:range rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Proposal"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;hasProposal"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isStatusOf -->

```

```

<owl:ObjectProperty rdf:about="&Public_Contracting;isStatusOf">
  <rdfs:range rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure_Status"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;hasStatus"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#isSupplierOf -->

<owl:ObjectProperty rdf:about="&Public_Contracting;isSupplierOf">
  <rdfs:range rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Supplier"/>
</owl:ObjectProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#responseClarifications -->

<owl:ObjectProperty rdf:about="&Public_Contracting;responseClarifications">
  <rdfs:domain rdf:resource="&Public_Contracting;Clarifications"/>
  <rdfs:range rdf:resource="&Public_Contracting;Procedure_Competent_Jury_for_Clarifications"/>
  <rdfs:range rdf:resource="&Public_Contracting;Supplier"/>
  <owl:inverseOf rdf:resource="&Public_Contracting;askClarifications"/>
</owl:ObjectProperty>

<!--
////////////////////////////////////
//
// Data properties
//
////////////////////////////////////
-->

<!-- http://127.0.0.1/Public_Contracting.owl#adju_factor -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;adju_factor">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Adjudication_Criterion"/>
  <rdfs:range rdf:resource="&xsd;float"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#adju_name -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;adju_name">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Adjudication_Criterion"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#cand_approved -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;cand_approved">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>

```

```

    <rdfs:domain rdf:resource="&Public_Contracting;Candidature"/>
    <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#cand_dispatchDate -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;cand_dispatchDate">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Candidature"/>
  <rdfs:range rdf:resource="&xsd;dateTime"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#cand_finalResult -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;cand_finalResult">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Candidature"/>
  <rdfs:range rdf:resource="&xsd;float"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#cand_refnumber -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;cand_refnumber">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Candidature"/>
  <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#cpv_name -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;cpv_name">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;CPV"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#dist_city -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;dist_city">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;District"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#ent_address -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;ent_address">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Entity"/>
  <rdfs:range rdf:resource="&xsd:string"/>

```

```

</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#ent_email -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;ent_email">
  <rdfs:domain rdf:resource="&Public_Contracting;Entity"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#ent_name -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;ent_name">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Entity"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#ent_nif -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;ent_nif">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Entity"/>
  <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#pers_email -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;pers_email">
  <rdfs:domain rdf:resource="&Public_Contracting;Person"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#pers_name -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;pers_name">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Person"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#proc_Date -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;proc_Date">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:range rdf:resource="&xsd;dateTime"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#proc_Description -->

```

```

<owl:DatatypeProperty rdf:about="&Public_Contracting;proc_Description">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://127.0.0.1/Public_Contracting.owl#proc_refNumber -->

<owl:DatatypeProperty rdf:about="&Public_Contracting;proc_refNumber">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->

<!-- http://127.0.0.1/Public_Contracting.owl#Adjudication_Criterion -->

<owl:Class rdf:about="&Public_Contracting;Adjudication_Criterion">
  <owl:disjointWith rdf:resource="&Public_Contracting;CPV"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Candidature"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Clarifications"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Entity"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Person"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_Team"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Proposal"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Buyer -->

<owl:Class rdf:about="&Public_Contracting;Buyer">
  <rdfs:subClassOf rdf:resource="&Public_Contracting;Entity"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Supplier"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#CPV -->

<owl:Class rdf:about="&Public_Contracting;CPV">
  <owl:disjointWith rdf:resource="&Public_Contracting;Candidature"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Clarifications"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Entity"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Person"/>

```

```

    <owl:disjointWith rdf:resource="&Public_Contracting;Procedure"/>
    <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_Team"/>
    <owl:disjointWith rdf:resource="&Public_Contracting;Proposal"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Candidature -->

<owl:Class rdf:about="&Public_Contracting;Candidature">
  <owl:disjointWith rdf:resource="&Public_Contracting;Clarifications"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Entity"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Person"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_Team"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Proposal"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Clarifications -->

<owl:Class rdf:about="&Public_Contracting;Clarifications">
  <owl:disjointWith rdf:resource="&Public_Contracting;Entity"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Person"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_Team"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Proposal"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Direct_Adjustment -->

<owl:Class rdf:about="&Public_Contracting;Direct_Adjustment">
  <rdfs:subClassOf rdf:resource="&Public_Contracting;Procedure"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Normal_Procedure"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Restricted_By_Prior_Qualification"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Simplified_Direct_Adjustment"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#District -->

<owl:Class rdf:about="&Public_Contracting;District"/>

<!-- http://127.0.0.1/Public_Contracting.owl#Entity -->

<owl:Class rdf:about="&Public_Contracting;Entity">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasDistrict"/>
      <owl:someValuesFrom rdf:resource="&Public_Contracting;District"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="&Public_Contracting;Person"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_Team"/>

```

```

    <owl:disjointWith rdf:resource="&Public_Contracting;Proposal"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Normal_Procedure -->

<owl:Class rdf:about="&Public_Contracting;Normal_Procedure">
  <rdfs:subClassOf rdf:resource="&Public_Contracting;Procedure"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Restricted_By_Prior_Qualification"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Simplified_Direct_Adjustment"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Person -->

<owl:Class rdf:about="&Public_Contracting;Person">
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_Team"/>
  <owl:disjointWith rdf:resource="&Public_Contracting;Proposal"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Procedure -->

<owl:Class rdf:about="&Public_Contracting;Procedure">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasStatus"/>
      <owl:onClass rdf:resource="&Public_Contracting;Procedure_Status"/>
      <owl:qualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasProcedureTeam"/>
      <owl:onClass rdf:resource="&Public_Contracting;Procedure_Team"/>
      <owl:qualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasBuyer"/>
      <owl:onClass rdf:resource="&Public_Contracting;Buyer"/>
      <owl:qualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasCPV"/>
      <owl:someValuesFrom rdf:resource="&Public_Contracting;CPV"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasAdjudicationCriterion"/>
      <owl:someValuesFrom rdf:resource="&Public_Contracting;Adjudication_Criterion"/>
    </owl:Restriction>
  </rdfs:subClassOf>

```

```

        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&Public_Contracting;hasProposal"/>
            <owl:someValuesFrom rdf:resource="&Public_Contracting;Proposal"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&Public_Contracting;hasSupplier"/>
            <owl:someValuesFrom rdf:resource="&Public_Contracting;Supplier"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_Team"/>
    <owl:disjointWith rdf:resource="&Public_Contracting;Proposal"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Procedure_Board_Member -->

<owl:Class rdf:about="&Public_Contracting;Procedure_Board_Member">
    <rdfs:subClassOf rdf:resource="&Public_Contracting;Person"/>
    <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_Competent_Jury_for_Clarifications"/>
    <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_President"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Procedure_Competent_Jury_for_Clarifications -->

<owl:Class rdf:about="&Public_Contracting;Procedure_Competent_Jury_for_Clarifications">
    <rdfs:subClassOf rdf:resource="&Public_Contracting;Person"/>
    <owl:disjointWith rdf:resource="&Public_Contracting;Procedure_President"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Procedure_President -->

<owl:Class rdf:about="&Public_Contracting;Procedure_President">
    <rdfs:subClassOf rdf:resource="&Public_Contracting;Person"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Procedure_Status -->

<owl:Class rdf:about="&Public_Contracting;Procedure_Status"/>

<!-- http://127.0.0.1/Public_Contracting.owl#Procedure_Team -->

<owl:Class rdf:about="&Public_Contracting;Procedure_Team">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&Public_Contracting;hasCompetentJuryForClarifications"/>
            <owl:onClass rdf:resource="&Public_Contracting;Procedure_Competent_Jury_for_Clarifications"/>
            <owl:maxQualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxQualifiedCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

```



```

    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasPresident"/>
      <owl:onClass rdf:resource="&Public_Contracting;Procedure_President"/>
      <owl:qualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasBoardMember"/>
      <owl:onClass rdf:resource="&Public_Contracting;Procedure_Board_Member"/>
      <owl:maxQualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">3</owl:maxQualifiedCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="&Public_Contracting;Proposal"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Proposal -->

<owl:Class rdf:about="&Public_Contracting;Proposal"/>

<!-- http://127.0.0.1/Public_Contracting.owl#Restricted_By_Prior_Qualification -->

<owl:Class rdf:about="&Public_Contracting;Restricted_By_Prior_Qualification">
  <rdfs:subClassOf rdf:resource="&Public_Contracting;Procedure"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&Public_Contracting;hasCandidature"/>
      <owl:someValuesFrom rdf:resource="&Public_Contracting;Candidature"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="&Public_Contracting;Simplified_Direct_Adjustment"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Simplified_Direct_Adjustment -->

<owl:Class rdf:about="&Public_Contracting;Simplified_Direct_Adjustment">
  <rdfs:subClassOf rdf:resource="&Public_Contracting;Procedure"/>
</owl:Class>

<!-- http://127.0.0.1/Public_Contracting.owl#Supplier -->

<owl:Class rdf:about="&Public_Contracting;Supplier">
  <rdfs:subClassOf rdf:resource="&Public_Contracting;Entity"/>
</owl:Class>

```

