



Universidade do Minho
Escola de Engenharia

João Pedro Pinto Vasques Ribeiro

**Resolução de Conflitos em Linha:
OntoLab uma aplicação ao Direito Laboral**

Tese de Mestrado
Mestrado em Informática

Trabalho efectuado sob a orientação do
Professor Doutor Paulo Novais

Outubro 2011

DECLARAÇÃO

Nome: João Pedro Pinto Vasques Ribeiro
Nº Cartão Cidadão /BI: 12793884
Correio eletrónico: ordep_scout@hotmail.com
Curso Mestrado em Informática

Tel./Telem.: 916234357

Ano de conclusão da dissertação: 2011

Escola de Engenharia, Departamento/Centro: Departamento de Informática

TÍTULO DISSERTAÇÃO:

Título em PT : Resolução de Conflitos em Linha: OntoLab uma aplicação ao Direito Laboral

Título em EN : Online Dispute Resolution: OntoLab an application in Labor Law

Orientadores : Professor Doutor Paulo Jorge Freitas de Oliveira Novais

Declaro sob compromisso de honra que a dissertação agora entregue corresponde à que foi aprovada pelo júri constituído pela Universidade do Minho.

Declaro que concedo à Universidade do Minho e aos seus agentes uma licença não-exclusiva para arquivar e tornar acessível, nomeadamente através do seu repositório institucional, nas condições abaixo indicadas, a minha dissertação, em suporte digital.

Concordo que a minha dissertação seja colocada no repositório da Universidade do Minho com o seguinte estatuto (assinale um):

1. Disponibilização imediata do trabalho para acesso universal;
2. Disponibilização do trabalho para acesso exclusivo na Universidade do Minho durante o período de
 1 ano, 2 anos ou 3 anos, sendo que após o tempo assinalado autorizo o acesso universal.
3. Disponibilização do trabalho de acordo com o **Despacho RT-98/2010 c)** (embargo___ anos)

Braga, 28 /10 /2011

Assinatura: _____

Agradecimentos

Ao longo do tempo que decorreu esta dissertação, desde a investigação das várias tecnologias até à escrita deste documento, tive a oportunidade de contar com o apoio de diversas pessoas que directa ou indirectamente, contribuíram para a realização da mesma.

Em primeiro lugar. Agradeço ao Professor Paulo Novais, meu orientador, pela sua disponibilidade, pela sua ajuda e espírito crítico que teve ao longo deste tempo, dando-me sempre todo o apoio para realização desta dissertação.

Ao Davide Carneiro, foi quem incentivou para que esta tese fosse terminada, mesmo nos momentos mais difíceis, onde pensei desistir por falta de motivação e com a sua ajuda consegui obter a motivação necessária para terminar este projecto.

Ao Professor Francisco Andrade, pela ajuda na escolha dos artigos mais importantes para o avanço do protótipo da ontologia.

À Sandra, por todo apoio e motivação dada nos momentos em que desistir era o que mais queria e pela sua paciência pela falta de tempo que tinha para ela.

À Versão Integral, mais precisamente ao Nuno e ao Hugo pela disponibilidade que sempre tiveram em me facultar os dias que necessitei de me ausentar.

À minha família que sempre esteve presente para me apoiar e ajudar a ultrapassar as adversidades que ocorreram.

A todos, os meus sinceros agradecimentos.

Esta dissertação surge integrada no projecto de investigação TIARAC – Telemática e Inteligência Artificial na Resolução Alternativa de Conflitos, financiado pela FCT-Fundação Ciência e Tecnologia, Ref.PTDC/JUR/71354/2006.

Resumo

A Resolução Alternativa de Conflitos, visa, por um lado, promover o acesso à justiça, sendo um meio alternativo à resolução de conflitos judiciais, neste caso recorre-se aos tribunais arbitrais e julgados de paz. Por outro lado, visa, também, apoiar a criação e o funcionamento de meios extrajudiciais para resolução alternativa de conflitos, incluindo a mediação, negociação e arbitragem.

No de Direito, especificamente na área do direito do trabalho, a utilização de métodos de resolução alternativa de conflitos é muito vantajosa. É conhecido o estado actual da justiça em Portugal, grande parte dos casos em litígio arrastam-se ao longo de anos nos tribunais, sem existir previsão de resolução. O recurso à resolução de conflitos em linha, visa retirar dos tribunais vários destes casos, tornando assim a resolução do litígio mais célere, bem como menos dispendiosa para ambas as partes.

Estes métodos de resolução *online* de conflitos são uma abordagem bastante recente, que se serve da Internet e de ferramentas de suporte à decisão. Neste tipo de ambientes as partes interagem e expõe os seus pontos de vista, em qualquer momento e em qualquer local, uma vez que estes sistemas estão disponíveis em linha.

Nesta dissertação procurou-se definir qual a melhor metodologia para o desenvolvimento de uma ontologia base, nesta área do conhecimento. Nesse sentido desenvolveu-se uma ontologia e um motor de inferência de conhecimento que actua sobre ela de forma a disponibilizar o conhecimento obtido ao sistema.

Com a utilização das ontologias aliadas aos sistemas de resolução de conflitos em linha existiram enormes ganhos para a justiça, na medida em que estes processos tendem a ser mais transparentes, mais rápidos e mais justos.

Abstract

Alternative Dispute Resolution aims to promote access to justice, as an alternative mean to litigation in court. On the other hand, Alternative Dispute Resolution also aims at supporting the establishment and operation of non-judicial means for alternative dispute resolution, including mediation, negotiation and arbitration.

In the field of the law, specifically in labor law, Alternative Dispute Resolution is very advantageous. In fact, the current state of the legal field in Portugal is well known, in which in most of the cases a dispute drags on for years in court, without prediction on when to end. The use of alternative methods, aims to remove many of the cases from the courts, thus making the resolution of the dispute faster and less expensive for both parties.

These methods of online conflict resolution are a fairly recent approach, which uses Web services and decision support tools. In such environments the parts interact and expose their points of view, at any time and any place, since these systems are available online.

In this work we tried to define the best methodology for the development of an ontology base in this area of knowledge. In this sense developed an ontology and an inference engine of knowledge that acts on it in order to provide the knowledge gained to the system.

With the combined use of ontologies to online systems of conflict resolution there are huge gains to justice, to the extent that these processes tend to be more transparent, faster and fairer.

Glossário

| | |
|------|---|
| ACL | Agent Communication Language |
| AI | Artificial Intelligence |
| ADR | Alternative Dispute Resolution |
| API | Application Program Interface |
| DAI | Distributed Artificial Intelligence |
| DAWG | Data Access Working Group |
| ESCP | European Small Claims Procedure |
| FIPA | Foundation for Intelligent Physical Agents |
| GUI | Graphical User Interface |
| JVM | Java Virtual Machine |
| KIF | Knowledge Interchange Format |
| LDS | Legal Decision-making System |
| OWL | Web Ontology Language |
| ODR | Online Dispute Resolution |
| OKBC | Open Knowledge Base Connectivity |
| PDF | Portable Document Format |
| RDF | Resource Description Framework |
| RDFs | Resource Description Framework Schema Specification |
| UDRP | Uniform Domain Names Dispute Resolution Policy |
| URIs | Uniform Resource Identifier |
| W3C | World Wide Web Consortium |
| XML | eXtensible Markup Language |

Índice

| | |
|--|-----------|
| 1 Introdução | 1 |
| 1.1 Motivação..... | 1 |
| 1.2 Visão da Dissertação..... | 2 |
| 1.3 Projecto UMCourt | 3 |
| 1.4 Projectos Relacionados | 4 |
| 1.5 Objectivos..... | 9 |
| 1.5.1 Objectivo Global..... | 9 |
| 1.5.2 Objectivos Especificos..... | 10 |
| 1.6 Metodologia da Investigação | 10 |
| 1.7 Estrutura do Documento | 10 |
| 2 Domínio do Problema..... | 13 |
| 2.1 Resolução Alternativa de Conflitos..... | 13 |
| 2.1.1 Negociação..... | 14 |
| 2.1.2 Mediação..... | 15 |
| 2.1.3 Arbitragem..... | 16 |
| 2.2 Resolução Online de Conflitos | 17 |
| 2.3 Modelação de Serviços ODR..... | 19 |
| 2.3.1 Partes Implicadas | 20 |
| 2.3.2 Tipos de Comunicação..... | 21 |
| 2.4 Inteligência Artificial no ODR | 22 |
| 3 Web Semântica..... | 25 |
| 3.1 O que é a Web Semântica? | 25 |
| 3.2 Arquitectura Ideal para a Web Semântica | 26 |
| 3.3 Arquitectura Actual para a Web Semântica | 27 |
| 4 Ontologias..... | 29 |

| | |
|---|-----------|
| 4.1 Métodos de Desenvolvimento Ontológico | 31 |
| 4.1.1 Ontology Development 101 | 31 |
| 4.1.2 On-to-Knowledge | 33 |
| 4.2.3 Methontology | 34 |
| 4.2 Ferramentas de Desenvolvimento Ontológico..... | 35 |
| 4.2.1 Protégé..... | 35 |
| 4.2.2 eXtensible Markup Language (XML)..... | 36 |
| 4.2.3 Resource Description Framework (RDF) | 37 |
| 4.2.4 RDF Schema | 40 |
| 4.2.5 Web Ontology Language (OWL) | 40 |
| 4.2.6 JENA | 41 |
| 4.2.7 SPARQL..... | 46 |
| 5 OntoLab | 49 |
| 5.1 Planeamento da Ontologia | 50 |
| 5.2 Desenvolvimento da Ontologia..... | 51 |
| 5.2.1 Especificação da Ontologia..... | 51 |
| 5.2.2 Gramática da Ontologia..... | 51 |
| 5.2.3 Classes..... | 54 |
| 5.2.4 Indivíduos | 58 |
| 5.2.5 Propriedades | 60 |
| 5.3 Motor de Inferência..... | 63 |
| 5.3.1 Opções de Implementação..... | 63 |
| 5.3.2 Implementação da Aplicação | 63 |
| 6 Conclusões e Trabalho Futuro | 69 |
| 6.1 Conclusões..... | 69 |
| 6.2 Trabalho futuro | 71 |
| Bibliografia | 73 |
| Anexo I - Artigos do Código do Trabalho..... | 77 |
| Anexo II – Código RDF gerado pelo Protégè..... | 83 |

Índice de Figuras

| | |
|---|----|
| Fig 1: Fluxograma do serviço de resolução de conflitos..... | 19 |
| Fig 2: Sistema ODR | 20 |
| Fig 3: Negociação ODR..... | 21 |
| Fig 4: Mediação ODR..... | 21 |
| Fig 5: Arbitragem ODR..... | 22 |
| Fig 6: Modelo Web 3.0 adaptado a partir do modelo actual de Web | 26 |
| Fig 7: Modelo Conceptualizado da Web 3.0..... | 28 |
| Fig 8: Definição do vocabulário ontológico | 30 |
| Fig 9: Procedimento de desenvolvimento <i>Ontology Development 101</i> | 32 |
| Fig 10: Processo de desenvolvimento <i>On-to-Knowledge</i> | 33 |
| Fig 11: Processo de desenvolvimento e ciclo de vida da <i>Methontology</i> | 34 |
| Fig 12: Protégé | 36 |
| Fig 13: Descrição simplificada de um Empregado em RDF..... | 38 |
| Fig 14: Representação de um <i>triple</i> | 39 |
| Fig 15: Representação de um <i>triple</i> sobre a forma de grafo..... | 39 |
| Fig 16: Grafo RDF mais detalhado | 39 |
| Fig 17: Descrição simplificada de uma propriedade em OWL. | 40 |
| Fig 18: Arquitectura da API Jena | 42 |
| Fig 19: Hierarquia e relacionamentos das interfaces..... | 43 |
| Fig 20: Definição de partes num caso..... | 45 |
| Fig 21: Definição de partes num caso..... | 45 |
| Fig 22: Saída produzida após a inferência..... | 46 |
| Fig 23: Actividades de planeamento da ontologia | 50 |
| Fig 244: Lista de termos | 52 |
| Fig 255: Gramática da ontologia | 53 |
| Fig 266: Representação de Classes..... | 54 |

1 Introdução

Este documento enquadra-se no domínio das ontologias, sendo o objecto de estudo o Direito Laboral Português. Pretende-se realçar o potencial papel das ontologias no desenvolvimento de uma nova geração de ferramentas de resolução de conflitos baseadas na tecnologia.

1.1 Motivação

O constante crescimento dos casos de divergências entre entidades empregadoras e trabalhadores, impulsionado fortemente pela crise que estamos a viver nos dias de hoje, contribui para que os Tribunais estejam sobrelotados com casos deste tipo, fazendo-os arrastar-se por longos períodos de tempo. Para minimizar o número de casos em litígio nos Tribunais prevê-se que a utilização da *Web*, num futuro próximo, possa contribuir activamente na resolução rápida e simples deste tipo de conflitos. No entanto, para que tal aconteça é fundamental existir um modelo estruturado e organizado de dados para que os conflitos possam ser entendidos e resolvidos por um computador, minimizando a intervenção de mão humana na sua resolução.

O estudo das ontologias é bastante importante, pois com recurso a estas será possível colocar agentes inteligentes a comunicarem entre si, conseguindo resolver com uma maior

brevidade casos que se poderiam arrastar durante anos nos tribunais. Assim, para que tal possa acontecer existe a necessidade de criar as ontologias, de forma robusta e fiável, para que ao longo do tempo se tornem cada vez mais sólidas através dos dados que lhe vão sendo inferidos.

Estes foram os principais factores que constituíram a motivação fundamental desta dissertação. O ponto de partida é o Código do Trabalho Português, mais especificamente os artigos relativos aos direitos e deveres dos empregadores e empregados.

1.2 Visão da Dissertação

O desenvolvimento da *Internet* e os seus serviços tem estimulado uma mudança radical na sociedade em geral. Uma sociedade baseada na produção industrial tornou-se numa sociedade baseada na informação. Nesta nova realidade o domínio legal não é excepção. Actualmente os tribunais não se encontram preparados para esta realidade, pois ainda têm muita documentação em papel, não estando assim preparados para que exista troca de informação entre o juiz e as partes por meios electrónicos.

No entanto, é fundamental que esta lacuna seja rapidamente ultrapassada, de modo a permitir, por um lado, que os tribunais se ajustem e acompanhem o desenvolvimento global, e, por outro lado, que os tribunais agilizem o seu funcionamento e reduzam o tempo de resolução dos litígios.

Numa primeira fase surge, então, a resolução alternativa de conflitos com o objectivo de retirar dos tribunais alguns dos litígios mais simples de forma a tornar a sua resolução mais rápida. Surge, deste modo, a resolução de litígios em tribunais arbitrais e julgados de paz.

A necessidade de informatizar dados e simplificar ainda mais a resolução deste tipo de litígios conduz à resolução *online* de conflitos. A resolução *online* de conflitos é um ramo da resolução alternativa de conflitos que recorre à tecnologia como intermediário para facilitar a resolução de diferendos entre as partes. [1] Esta resolução pode obter-se por três vias: a via da negociação, a via da mediação ou a via da arbitragem, no entanto existe ainda a possibilidade, em caso de necessidade, de recorrer a uma combinação destas três vias para obter a resolução do conflito. [2].

O conhecimento deve-se encontrar estruturado de forma a ser compreendido pelo sistema de resolução *online* de conflitos [3]. Nesse sentido a utilização de ontologias afigura-se como sendo a via mais indicada.

Essas ontologias devem conter uma estrutura bem definida, elaborada com recurso à ajuda de profissionais dentro da área do direito, para que sejam sólidas e tenham todo o conhecimento necessário para que sirvam de base do sistema de resolução *online* de conflitos.

A necessidade de realizar consultas e inferir novo conhecimento à ontologia conduz à utilização de uma ferramenta de apoio onde seja possível visualizar alguns dados relevantes para o litígio, como por exemplo: o histórico associado às disputas tanto do empregador como do empregado, as notas de culpa, bem como os dados que levaram ao litígio entre as partes, tornando-se essa ferramenta bastante importante num sistema de resolução *online* de conflitos.

1.3 Projecto UMCourt

O projecto UMCourt está a ser desenvolvido na Universidade do Minho no âmbito do projecto TIARAC (Telemática e Inteligência Artificial na Resolução Alternativa de Conflitos). Este projecto visa explorar as potencialidades do uso combinado das novas tecnologias com técnicas de Inteligência Artificial e da Lei, no domínio da resolução alternativa de conflitos. [4]

O UMCourt é um sistema multi-agente, baseado em padrões e tecnologias abertas, tornando-se assim flexível, dinâmico e expansível. Pretende ser mais que uma simples ferramenta de gestão de informação jurídica, sendo o seu núcleo um grupo de agentes que incorporam técnicas inteligentes de resolução de problemas.

A plataforma resultante é capaz de fornecer serviços tais como: determinar as possíveis soluções para um conflito e as respectivas probabilidades; fornecer informações relevantes em tempo real; criar automaticamente documentos legais; calcular valores de conceitos legais para cada disputa; entre outros.[5]

Peritos em Direito Laboral trabalham em estreita cooperação com a equipa de desenvolvimento do UMCourt para que a plataforma resultante tenha a abordagem correcta. A definição de padrões de representação da informação ou a formalização de argumentação são desafios com os quais a área jurídica se confronta frequentemente e que são também abordados neste projecto.

1.4 Projectos Relacionados

Pequenos procedimentos fazem o ponto de passagem entre a formalidade dos tribunais (litígios formais) e a resolução alternativa de conflitos *ADR* (do inglês *Alternative Dispute Resolution*), ou seja, quando as disputas em questão envolvem pequenas coisas podem ser resolvidas recorrendo ao *ADR*, tornando a resolução mais rápida, com um custo reduzido e com menos formalidade que a que existiria no tribunal. Nesta secção procede-se à apresentação de um conjunto de projectos de resolução alternativa de conflitos.

European Small Claims Procedure

O *ESCP* (*European Small Claims Procedure*) é um conjunto de procedimentos elaborados pela Comissão Europeia que permite superar a principal limitação existente nos processos *ADR*, ou seja, o facto dos procedimentos se encontrarem restritos a uma jurisdição especial. A implementação do *ESCP* aconteceu em todos os Estados Membros da União Europeia em Janeiro de 2009. A sua principal vantagem, é a contemplação da execução de decisões em qualquer um dos Estados membros sem a necessidade de apresentar o reconhecimento formal das mesmas [6].

Grandes expectativas são colocados no *ESCP*, este procedimento será um grande desafio, porque ao contrário do *UDRP* (*Uniform Domain Names Dispute Resolution Policy*), que é um processo totalmente *online* para lidar com queixas específicas, o *ESCP* vai lidar com vários litígios em tanto em matéria civil, como comercial, sendo que o seu principal objectivo é a criação de um procedimento eficiente e de baixos custos aplicáveis, que só pode ser conseguido através de um procedimento escrito, assistido por formulários electrónicos, por *email* ou videoconferência.

O regulamento permite a utilização de novas tecnologias para a transferência de informações e provas entre os tribunais dos Estados Membros, no entanto fica a questão: "Será que estes decidirão através dos seus próprios regulamentos ou pelas normas definidas no procedimento da *ESCP*?" De salientar que, dado que a *ESCP* é um procedimento e não uma directiva, esta questão é discutível, e desta forma tem-se deixado muitos aspectos ao critério dos Estados Membros, o que poderá pôr em causa a segurança jurídica que se pretende que seja única e baseada num regulamento europeu.

Com a implementação do *ESCP* um sistema *ODR* poderá vir a surgir na Europa no decorrer deste ano. Espera-se ainda que o *ESCP* contribua para mitigar o problema de legitimidade que também dificulta o surgimento de *ODR*.

e-Justice Centre no Second Life

O Ministério da Justiça Português já deu os primeiros passos nesta abordagem e possui já um *e-Justice Centre* no *Second Life*, que foi desenvolvido em conjunto pelo Departamento de Comunicação e Arte da Universidade de Aveiro e a Faculdade de Direito da Universidade Nova de Lisboa. Este centro fornece serviços de mediação e arbitragem para todos os utilizadores da plataforma *Second Life*, resolvendo assim litígios resultantes de relações de consumo ou de qualquer contrato assinado entre as partes [7].

A questão mais interessante é que o *e-Justice Centre*, é um centro de mediação e arbitragem no mundo virtual 3D *Second Life*, por isso os primeiros passos já estão dados, muito brevemente surgirão *Web Services ODR* para resolução de litígios reais.

British Columbia ODR

Dois projectos de *ODR* estão actualmente a decorrer em *British Columbia* [11]. O primeiro é um projecto patrocinado pela mediadora MediateBC e destina-se a casos de direito da família que envolvem questões entre dois cônjuges, companheiros ou parceiros, tais como:

- Onde e com quem as crianças vão viver.
- Como partilhar o custo da criação dos filhos.
- Como dividir os bens da família ou pensões.
- O que fazer com uma casa de família.

O processo é privado, informal e envolve um mediador de direito da família (um mediador de direito de família deve frequentar uma formação especial em *British Columbia* para ser capaz de usar esta designação). Nenhuma tecnologia especial é necessária, basta as partes terem um computador com uma *webcam*, em seguida, o mediador pode usar ferramentas de conferência *web*. Alternativamente, as partes podem usar *email* ou até mesmo apenas um telefone.

O aspecto importante deste projecto-piloto é tentar unir as pessoas com as tecnologias existentes para que a distância não seja um obstáculo, a fim de ajudar a resolver os verdadeiros problemas humanos e demonstrar "prova de conceito" da ideia em *British Columbia*.

O segundo projecto-piloto, também patrocinado pela MediateBC em conjunto com a Procuradoria-Geral, visa a resolução de outros litígios em que as disputas sejam de baixo valor. Embora o final deste projecto-piloto tenha sido declarado em 31 de Março de 2011, acredita-se que o mesmo ainda está em curso e a aceitar novos casos.

Para serem aceites num processo deste tipo ambas as partes devem:

- Ser residentes em *British Columbia*.
- Actualmente estar envolvidos numa disputa de baixo valor (não familiar); por exemplo, litígios do tipo *business-to-business* ou *business-to-client*.
- Encontrar-se dispostos a participar voluntariamente num sistema de litígio *ODR*.
- Estar dispostos a dar *feedback* sobre a sua experiência, por exemplo, através do preenchimento de um pequeno inquérito.
- Querer completar a sua parte do processo de *ODR*, entre outras condições, como por exemplo ter acesso a um computador com acesso à *Internet*.

General Electric Co

A divisão de petróleo e gás da companhia *General Electric*, está a efectuar testes num sistema de resolução alternativa de conflitos do tipo *ODR*, com o objectivo de redução de tempo na resolução dos conflitos e de gastos em advogados.

A divisão da GE, que fornece equipamentos e serviços para empresas petrolíferas e empresas de construção, exige que milhares de fornecedores concordem com *cybersettlements* em disputas simples. Esta abordagem encontra-se em experiência em Itália com o intuito de resolver as divergências com os fornecedores até ao valor máximo de 50.000€.

Existe uma preocupação sobre o custo actual da arbitragem comercial, pois esta está a ter preços elevados, deste modo a criação de um sistema de *ODR* foi um passo muito importante para reduzir gastos [12].

Este sistema da GE envolve licitação cega, para ver se as partes chegam a um consenso sobre um montante de liquidação. Existem regras de arbitragem, mas o árbitro só comunica *online* e sem audiência.

Como exemplo pode referir-se que a GE procurava ser reembolsada por uma empresa fornecedora, pois esta tinha-lhe vendido 48 tubos que eram defeituosos. A disputa não se resolveu sobre o sistema inicialmente proposto e foi para um sistema de arbitragem *online*. A GE ganhou 3.160 €, sendo que 3.036 € foram atribuídos para os materiais com defeito e o restante para cobrir algumas das taxas que pagou. De salientar que todo este processo foi rapidamente resolvido e sem custo elevado para as partes.

A empresa está a monitorar de perto os resultados do programa e diz que é cedo demais para dizer se *cybersettlement* veio para ficar ou quanto dinheiro vai ser economizado, mas ao nível de tempo já se verifica alguma economia, pois todos os casos foram resolvidos num período de tempo inferior a 80 dias.

Singapore online mediation program

O poder jurídico de Singapura no final do ano 2000 afirmava ser o primeiro no mundo a introduzir um mecanismo de mediação *online*. Este mecanismo permitia que fossem resolvidos conflitos de disputas comerciais, relacionados com liquidação de pagamentos *online*, sem a necessidade das partes recorrerem aos tribunais.

Neste sistema o queixoso vai ao site de *ODR*, preenche os seus dados, incluindo informações para contacto da outra parte e propõe uma solução para o problema, o réu obtém os dados relativos ao caso e após consulta informa o moderador se aceita a resolução do caso num sistema *ODR*. Caso aceite é ouvido pelo moderador e apresenta a sua versão dos factos, via formulário define a sua opção para solução do problema e a partir daí começa o processo de mediação *ODR*, até que ambas as partes cheguem a um acordo [13].

Outros Projectos Universitários no Mundo

Várias universidades estão a criar grupos de investigação na área de *ODR*. A Universidade de Genebra tem uma equipa de investigação em *ODR*, em arbitragem e em tecnologias da informação. Um dos projectos desenvolvidos foi o estudo de quais os campos necessários para

um sistema de *ODR* ser funcional. Este grupo de pesquisa continua a fazer investigar activamente sobre esses temas, principalmente impulsionado pelo director executivo Dr. Thomas Schultz. Os resultados podem ser vistos em muitos artigos publicados e livros [8].

O *eBusiness McMaster Research Center* em Ontário tem como missão assegurar a liderança e apoio de infra-estruturas para pesquisa *eBusiness* para parceiros académicos e industriais. Para tal o *MeRC* concentra-se em três actividades principais: pesquisa, ensino e crescimento. A pesquisa é vista como uma forma de identificar e realizar uma investigação multidisciplinar nas áreas de *eBusiness* e garantir os recursos para apoiá-la. Por sua vez, a estimulação do ensino, através do fornecimento de seminários e *workshops* em *eBusiness*, tem o intuito de aumentar o conhecimento para os negócios. Finalmente, o *MeRC* que visa fornecer uma interface e reforçar o diálogo entre a comunidade empresarial e académica.

Existe, actualmente, um grande número de projectos de pesquisa em curso no *MeRC* que cobrem uma vasta gama disciplinar, desde o *eCommerce* ao roubo de identidade *online*, passando por projectos na área da negociação *online*. Nesses projectos são investigadas as melhores e mais seguras formas de fazer transacções *online*, bem como os métodos de resolução de conflitos a serem aplicados quando estas falham [9].

O *National Center for Technology and Dispute Resolution*, localizado na Universidade de Massachusetts, fornece uma abordagem interdisciplinar para o estudo do direito e da sociedade. Uma das principais figuras impulsionadoras na investigação *ODR* nesta universidade é professor Ethan Katsh que além de ser professor de estudos jurídicos nesta universidade também é o director do Centro Nacional de Tecnologia e Resolução de Conflitos. Entre todos os seus trabalhos neste campo salienta-se a publicação de três livros, todos relacionados com o novo papel que a tecnologia tem na lei, nomeadamente no tópico de resolução de conflitos [10].

Outros Projectos não universitários

A *SquareTrade* é uma empresa dos EUA que presta serviços na área de resolução de conflitos em leilões *online*. Esta empresa atribui um mediador que ajuda a resolver qualquer disputa, independentemente do valor que represente. A principal característica do sistema é que nenhuma acusação é feita ao reclamante, o serviço é rápido e facilmente disponível aos consumidores. A *SquareTrade* foi contratada pelo *eBay* para servir como seu mediador em disputas *business-to-business*. O número médio de mediações da *SquareTrade* foi de 2500

mediações por mês no ano de 2000. Por enquanto, este serviço é gratuito para utilizadores do *eBay* que efectuem transacções acima de 100 dólares [14].

A *World Intellectual Property Organisation (WIPO)* possui um centro de mediação e arbitragem, onde mais de duas mil queixas já foram tratadas com recurso a este procedimento. A modalidade de mediação *WIPO* está a trabalhar na infra-estrutura técnica para fornecer comunicações em tempo real, recursos de áudio e vídeo, bem como a criação de estruturas que possuem um conhecimento jurídico.

O *Word & Bond* é um sistema de arbitragem imparcial que fornece um padrão próprio para um sistema de vendas interactiva, combinando um conjunto de princípios universalmente aplicáveis e conhecidos como práticas *Word & Bond*. As promessas em relação à venda de bens e serviços têm de cumprir as seguintes condições definidas: o comprador necessita de a ter a certeza quanto ao bem que está a adquirir; e o bem adquirido tem que ser entregue dentro do tempo estabelecido. Caso o vendedor não consiga cumprir com estas condições previamente acordadas, vê-se obrigado a concordar com o início de um processo de arbitragem *ODR*, aceitando a decisão do árbitro seja ela qual for.

1.5 Objectivos

1.5.1 Objectivo Global

Pretende-se criar um sistema que seja capaz de resolver conflitos laborais suportado por tecnologias baseadas na *Web*. Actualmente existem já alguns modelos de *ODR*, no entanto estes são muito rudimentares, tratando-se de modelos em que ambas as partes comunicam com o sistema por uma *webcam* ou através de *email*.

Assim sendo, o trabalho documentado nesta dissertação foi desenvolvido no âmbito do projecto *UMCourt*, tendo como principal objectivo investigar métodos de criação de ontologias, de forma a desenvolver um protótipo para uma ontologia na área do direito laboral.

1.5.2 Objectivos Específicos

- Investigação dos métodos e ferramentas de desenvolvimento ontológico;
- Elaboração de uma lista de termos relevantes na área do direito laboral;
- Definição da gramática da ontologia, baseada na lista de termos previamente definida;
- Desenvolvimento de um protótipo da ontologia;
- Inserção de Leis, indivíduos e conflitos na ontologia;
- Criação de um motor de consulta e inferência de conhecimento.

1.6 Metodologia da Investigação

Esta metodologia, denominada de *action-research*, começa por identificar o problema para que uma hipótese possa ser formulada, na qual será baseado o desenvolvimento. Posteriormente a informação recolhida é recompilada, organizada e analisada, de uma forma contínua para que seja possível efectuar a construção de uma proposta para resolver o problema identificado. Finalmente, podem obter-se as conclusões com base nos resultados obtidos durante a investigação.

Assim sendo, para alcançar os objectivos planeados para este modelo de pesquisa, foram definidas cinco etapas complementares, que se encontram adiante descritas:

- Especificação do problema e suas características;
- Actualização constante e incremental e revisão do estado da arte;
- Idealização e desenvolvimento gradual e interactivo do modelo proposto;
- Estudo e implementação de uma solução através do desenvolvimento de um protótipo;
- Análise dos resultados e formulação de conclusões.

1.7 Estrutura do Documento

Este documento encontra-se redigido e estruturado em seis capítulos, sendo que esta introdução diz respeito ao primeiro capítulo.

O segundo capítulo descreve o domínio do problema, explicando os conceitos de resolução alternativa de conflitos e resolução *online* de conflitos, sendo que o maior enfoque é dado à resolução *online* de conflitos. O terceiro capítulo aborda o tema da *Web Semântica*, uma vez que a tendência da *Web* que se utiliza na actualidade é convergir para a *Web* semântica, tal acontecerá logo que as ontologias possuam uma utilização mais abrangente. Neste mesmo capítulo é ainda abordada a arquitectura ideal para a *Web* semântica, bem como a forma de adaptar a *Web* actual a um formato idêntico ao da *Web* semântica. O quarto capítulo define o conceito de ontologia e descreve algumas técnicas e ferramentas para representação e definição de ontologias. O quinto capítulo relata a OntoLab e descreve o trabalho efectuado desde a criação da ontologia à criação do motor de inferência e pesquisa. Por fim, o sexto capítulo apresenta as conclusões deste documento, bem como o trabalho futuro a elaborar e uma avaliação dos resultados obtidos.

2 Domínio do Problema

Neste capítulo são abordadas as temáticas chave desta dissertação: a resolução alternativa de conflitos e a resolução *online* de conflitos, sendo que o principal relevo é atribuído à resolução *online* de conflitos.

A resolução alternativa de conflitos já se encontra em utilização com bastante frequência actualmente, de referir que os julgados de paz são o meio mais utilizado. Em relação à resolução *online* de conflitos a sua utilização actual é ainda reduzida, mas prevê-se que a curto prazo muitos casos transitem dos tribunais tradicionais para sistemas de resolução *online* de conflitos, onde o tempo de espera pela resolução de um conflito é encurtado e os gastos inerentes bastante reduzidos.

2.1 Resolução Alternativa de Conflitos

O objectivo da resolução alternativa de conflitos é a redução do número de litígios em tribunal, encaminhando os que envolvam valores inferiores a 5.000€ para julgados de paz ou para meios não judiciais, onde os litígios são resolvidos através da negociação, da mediação ou

da arbitragem. Independentemente da via de resolução escolhida os litígios são solucionados de forma mais simples, mais rápida e mais barata.

2.1.1 Negociação

Na negociação a participação é voluntária e não existe mais ninguém envolvido, a não ser as partes em litígio. Não existe a participação de nenhum juiz, conselheiro, assistente social ou amigo experiente, que tente facilitar o processo de resolução ou impor uma solução rápida e que satisfaça ambas as partes.

Assim, a negociação é entendida como um processo colaborativo e informal pelo qual as partes comunicam, sem qualquer influência externa, de modo a alcançarem um resultado que possa satisfazer ambas [15]. Este método é amplamente utilizado em áreas muito distintas, como processos judiciais, divórcios ou situações de reféns. Trata-se de um processo não obrigatório, o que significa que os litigantes não têm de aceitar o resultado. Quanto à organização as técnicas de negociação podem ser classificadas como distributivas ou integrativas [16].

Na negociação distributiva a abordagem do conflito visa distribuí-lo pelas partes envolvidas, de maneira a maximizar a satisfação de ambos. Um excelente exemplo pode ser um divórcio em que as partes têm de se sentar para dividir os bens que possuem, não querendo nenhum deles ficar a perder.

Por sua vez, na negociação integrativa o problema pode ter mais soluções do que aquelas que são visíveis à primeira vista. Neste tipo de conflito as partes tentam trazer para a mesa os seus interesses. O objectivo é maximizar o valor em disputa e cada uma das partes vai tentar obter a maior parcela desse valor. Esta abordagem é mais utilizada em situações em que as partes têm entre si uma relação com continuidade e que lhes poderá trazer mais-valias no futuro.

Se utilizarmos a negociação em processos de conflitos laborais, onde os sindicatos são o negociador entre as partes (Empregado e Empregador), estes vão defender sempre os interesses dos trabalhadores, tentando maximizar as receitas e otimizar as condições de trabalho, mas por outro lado, os Empregadores tentam maximizar o lucro da empresa. O ponto comum é que a empresa precisa dos trabalhadores e os trabalhadores necessitam do trabalho sendo em torno

desse ponto que a negociação vai seguir. Neste tipo de negociação, o que um ganha o outro perde, ou seja, se os trabalhadores ganham um salário mais elevado a empresa vai perder esse dinheiro e assim diminuir o lucro, logo o objectivo é chegar a uma situação óptima para ambas as partes, relação ganhar-ganhar.

2.1.2 Mediação

A mediação é uma forma de resolução alternativa de conflitos em que os litigantes são guiados por uma entidade neutra e independente que tenta conduzir o processo, a fim de alcançar um resultado que satisfaça a ambas as partes em litígio [17].

Neste tipo de abordagem, como também é o caso da negociação, as partes é que têm o poder de decisão sobre o desfecho do conflito, pois esta não é imposta por nenhuma outra entidade, ao contrário do que acontece no tribunal. Existe uma parte neutra que é escolhida por ambos os litigantes e não tem autoridade para decidir sobre o resultado da disputa, só tem autoridade para a dirigir. Deve conduzir a disputa de forma a manter as partes focadas no objecto/situação que levou ao litígio, para facilitar toda a interacção e comunicação entre as partes. Os mediadores são, portanto, muito importantes, porque as suas habilidades e competências podem representar o sucesso ou o fracasso do processo de resolução de litígios.

Um mediador deve, portanto, ser capaz de comunicar e ser entendido por ambas as partes, dando instruções claras para cada uma. Deve fornecer conselhos estratégicos na hora certa, com o objectivo de facilitar o processo sem se esquecer de manter a sua neutralidade. Para este efeito, é importante que o mediador mostre diferentes perspectivas, reconheça as expectativas e frustrações das partes e consiga reagir a elas, tenha a capacidade de acalmar as partes quando a discussão fica mais sentimental e incentivá-las a alcançar um resultado satisfatório.

Assim, um mediador deve ter acima de tudo, boa capacidade de comunicação para que as suas ideias sejam correctamente transmitidas às partes. O mediador não deve ser um estranho frio que simplesmente examina os factos e decide sobre eles, tem de ser capaz de ler os sentimentos das partes [18].

O processo de mediação, a partir da perspectiva do mediador, geralmente apresenta três fases: apresentação do pedido por parte do requerente, contacto com ambas as partes em conflito e conhecimento e mediação do conflito.

Na primeira fase, após a apresentação do pedido pelo requerente, o mediador deve apresentar as partes caso não se conheçam e expor como ocorrerá o processo. Na segunda fase, o mediador deve conhecer o problema e compreendê-lo, necessita de saber quais os objectos em disputa, qual o crédito e as expectativas que cada parte demonstra. É muito importante nesta fase que o mediador aborde muito de perto cada uma das partes de modo a que entenda as características do conflito. Na terceira fase do processo, o mediador deve conhecer as regras do processo e definir uma estratégia para atingir a meta com base nas observações feitas na fase anterior.

Se o mediador compreendeu bem as singularidades do problema, a estratégia por ele adoptada deve funcionar, caso contrário, se o objectivo não está a ser alcançado, o mediador pode adaptar a sua estratégia durante o processo. Portanto, é importante que o mediador tenha a sensibilidade necessária para perceber como o processo está a ser interpretado por cada uma das partes, de modo a efectuar ajustes na estratégia, se necessário.

Um dos exemplos mais recentes de mediação conhecido é o processo do Médio Oriente, em que os EUA se encontram a mediar o conflito entre a Palestina e Israel, tendo como objectivo alcançar um cessar-fogo de longa duração [19].

2.1.3 Arbitragem

Na arbitragem ambas as partes podem utilizar a ajuda de uma organização independente e neutra para resolver o litígio, mas, ao contrário do que acontece na mediação, a entidade não tem um papel activo tentando ajudar as partes no processo [20].

Nesta abordagem, a entidade independente está lá somente para dar a sua opinião experiente no assunto, os árbitros simplesmente ouvem as partes e, com base nos factos apresentados, tomam uma decisão sem influenciar os litigantes durante as suas apresentações dos factos. O resultado do processo de arbitragem é também original porque pode ser obrigatório ou não obrigatório. Se num caso particular o resultado não é obrigatório, a arbitragem

é similar à mediação, excepto para o papel de entidade independente. No entanto, se a decisão do árbitro é obrigatória, as partes terão que aceitá-la como se fosse uma decisão do tribunal.

A arbitragem também apresenta algumas desvantagens, como por exemplo quando se trata da compra de um determinado objecto ou da assinatura de um contrato de trabalho, uma vez que as partes não são informadas que a única maneira de resolver qualquer conflito deste tipo é através de um processo de arbitragem obrigatória, sem a possibilidade de ir a tribunal, constituindo uma séria ameaça aos direitos humanos.

2.2 Resolução Online de Conflitos

Numa primeira geração do *ODR* a ideia principal era ter o ser humano como peça central tanto no planeamento como na tomada de decisões, assim os mediadores tinham um papel determinante, com grande responsabilidade e por esse motivo eram escolhidos criteriosamente de acordo com os seus conhecimentos e historial de casos.

No *ODR* actual está prevista a utilização *Web Service ODR*, sem qualquer intervenção humana no processo de decisão, sendo que o processo actual pode ser organizado em três etapas.

Assim, inicia-se quando surge um contacto entre uma das partes e o *Web Service ODR*, manifestando a intenção de iniciar um processo de resolução de um conflito, servindo o *interface Web* de informador sobre os litigantes. Nesse momento o sistema deve notificar as partes envolvidas no conflito e coloca-las em contacto, para ver se há disponibilidade de participarem na sua resolução. Se todas as partes concordarem, o sistema avança para a etapa seguinte, tentando reunir o máximo de informação relativa ao conflito.

Nesta segunda etapa as partes têm que fornecer todos os detalhes que considerem relevantes, desde documentos, objectos ou valores monetários que possam fornecer provas ou factos importantes para a resolução do caso. Por esse motivo é muito importante que o sistema entenda as expectativas que ambas as partes têm em relação ao desenrolar do conflito, sendo esta provavelmente a tarefa mais complicada para o *Web Service*, pois é totalmente autónomo, sem qualquer intervenção humana [21, 22].

Por fim, na terceira etapa o *Web Service* procede à mediação do conflito e à resolução do mesmo com os litigantes, sendo que, tal como no *ADR*, não existe obrigatoriedade de aceitar a decisão, cabendo às partes em litígio essa mesma aceitação ou não.

O *ODR* é uma abordagem nova, relativa à resolução de litígios já que as tecnologias em que está assente são também recentes. Existe no entanto um crescimento rápido na sua utilização, devido às inúmeras vantagens que possui: é fácil de usar; não existe necessidade do litigante se deslocar a nenhum local, podendo resolver o seu litígio em sua própria casa, necessitando unicamente de uma ligação *Web*; as interfaces são maioritariamente intuitivas, escondendo toda a complexidade relativa às leis e formalidades existentes por trás dos processos. A possibilidade de existir comunicação de forma síncrona e assíncrona, torna este tipo de ferramentas de fácil utilização, pois mesmo que os litigantes se encontrem em partes distintas do mundo é possível resolverem o seu litígio recorrendo a este *Web Service*, sem terem a necessidade de se deslocar à presença de um juiz, permanecendo assim num ambiente menos intimidativo, tornando as pessoas mais verdadeiras, retirando-lhes o receio de falar abertamente do que aconteceu.

A grande acessibilidade do *ODR*, aliada à sua rapidez de execução, à sua economia e à forma transparente como resolve os casos são as vantagens mais referidas, no entanto alguns pontos considerados vantajosos podem, ao mesmo tempo, ser um inconveniente. Se em alguns casos o acesso à *Web* é simples, existem outros em que isso não é verdade, existe um grande número de população que não têm este tipo de acesso ou não possui formação específica para utilizar ferramentas deste tipo, gerando-se assim desigualdades. Poderá existir uma formação específica para esse tipo de população, mas o tempo despendido para formação implicaria atrasos nos processos.

Por outro lado, se é verdade que na maioria dos países desenvolvidos as tecnologias utilizadas pelo *ODR* são baratas, nos países em desenvolvimento não é necessariamente verdadeiro o que pode levar a custos mais elevados, tornando-se assim uma tecnologia menos interessante.

Deste modo o grande objectivo é simplificar ao máximo a aplicação facilitando manuseamento e o acesso para uma grande parte da população.

Ferramentas electrónicas são, evidentemente, utilizadas mas são vistas como mero auxílio, sem qualquer autonomia, a sua utilidade é facilitar a gestão de informação e as comunicações entre as partes envolvidas. Nesta primeira geração as principais tecnologias

utilizadas são as mensagens instantâneas, vídeo conferência, chamadas telefônicas, fóruns de discussão e numa fase mais recente o *VideoPresence*. Estes sistemas têm como suporte uma página *Web* e representam o primeiro passo da tecnologia *ODR*, antes de se avançar com a autonomia dos sistemas inteligentes [23].

2.3 Modelação de Serviços ODR

Na figura 1, pode observar-se como funciona um serviço de resolução de conflitos, onde a primeira fase é a apresentação da queixa. Após esta primeira fase podem ocorrer duas situações distintas: existir resposta por parte do réu ou não existir resposta por parte do réu.

Caso ocorra resposta por parte do réu existe a possibilidade de contra resposta por parte do queixoso, para que se entre no processo de mediação, que levará à decisão e posterior encerramento do litígio.

Caso não exista resposta passa-se directamente à fase de decisão do litígio, nesta fase existe a possibilidade de apelo por parte do réu, mas maior parte dos casos são encerrados por decisão sumária, sem dar possibilidade de resposta ao réu, já que este não apresentou nenhuma contra proposta.

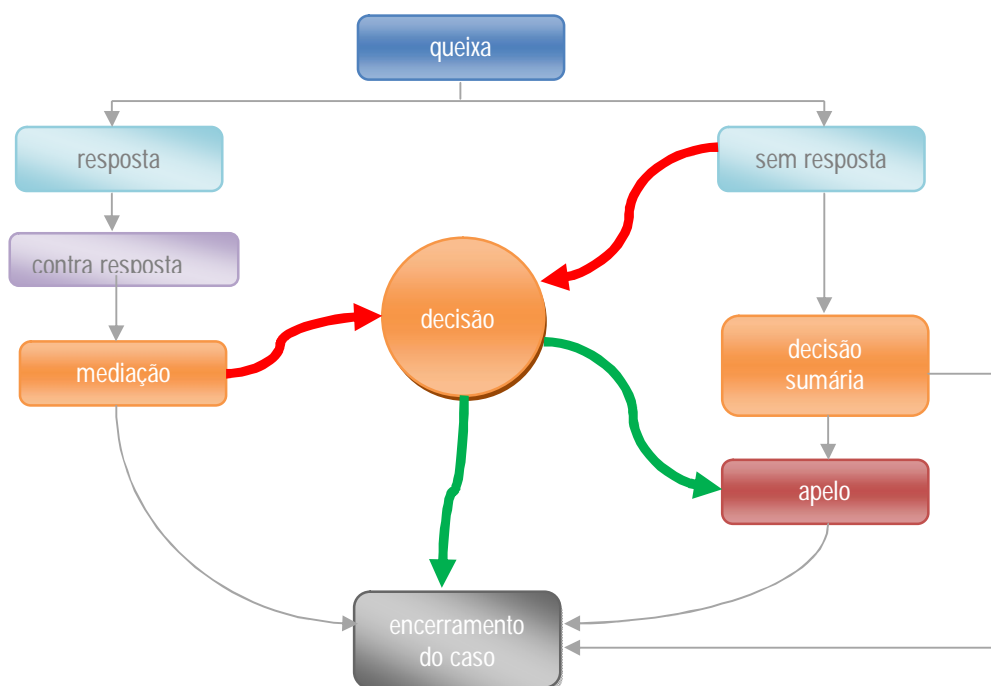


Fig 1: Fluxograma do serviço de resolução de conflitos

2.3.1 Partes Implicadas

Os diferentes actores implicados no sistema *ODR* são:

- Partes.
- Árbitro.
- Mediador.
- Peritos/Especialistas.
- Testemunhas.
- Administrador do *ODR*.
- O Administrador do Sistema.
- Visitante.

Podemos distinguir três tipos de actores: os actores que estão directamente implicados no processo de *ODR* (partes em cada processo, o árbitro, peritos ou testemunhas num determinado processo de mediação ou mesmo o mediador desse processo); os actores de gestão do sistema *ODR*, também são muito importantes (administrador do sistema e o administrador *ODR*); e, finalmente, os visitantes.

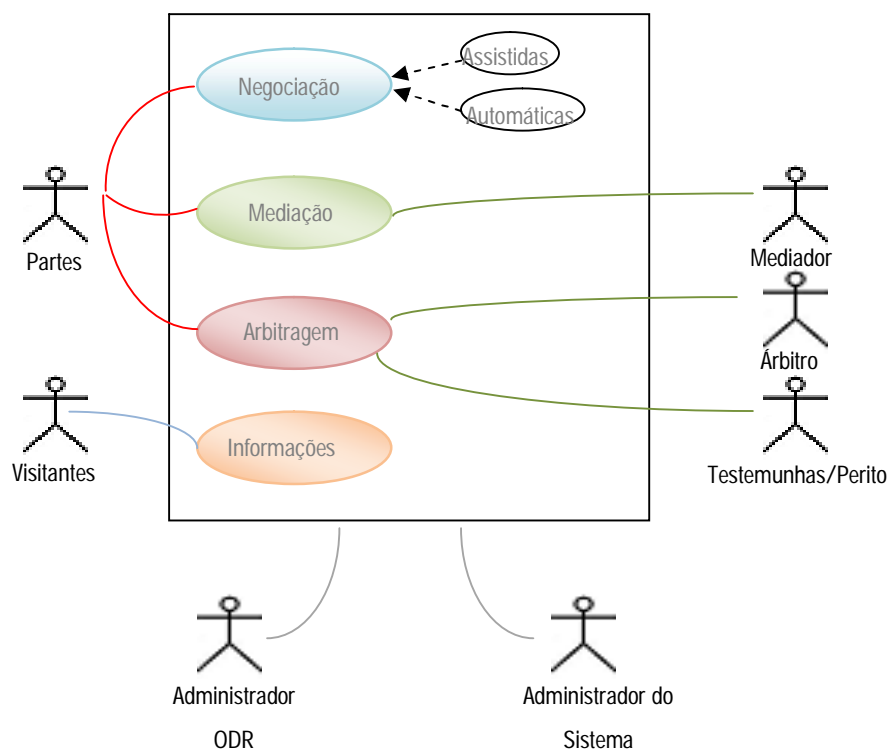


Fig 2: Sistema ODR

2.3.2 Tipos de Comunicação

Nos diagramas seguintes estão descritos todos os tipos possíveis de comunicação entre os intervenientes existentes no *ODR*.

Abaixo, representa-se um caso de negociação, neste caso ambas as partes interagem directamente uma com a outra, e o negociador é simplesmente uma interface *Web* que possui regras *ODR* bem definidas e consegue resolver o litígio sem ter necessidade de recorrer à presença física de um juiz que seja capaz de gerir o processo de negociação (figura 3).

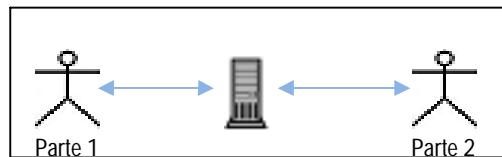


Fig 3: Negociação ODR

A mediação *ODR* (figura 4) é um processo um pouco mais complexo, deve existir um mediador entre as partes, tentando chegar a um consenso, de modo a que ambas fiquem satisfeitas com a decisão, sendo que neste tipo de processo quem o coordena é o administrador do *ODR*.

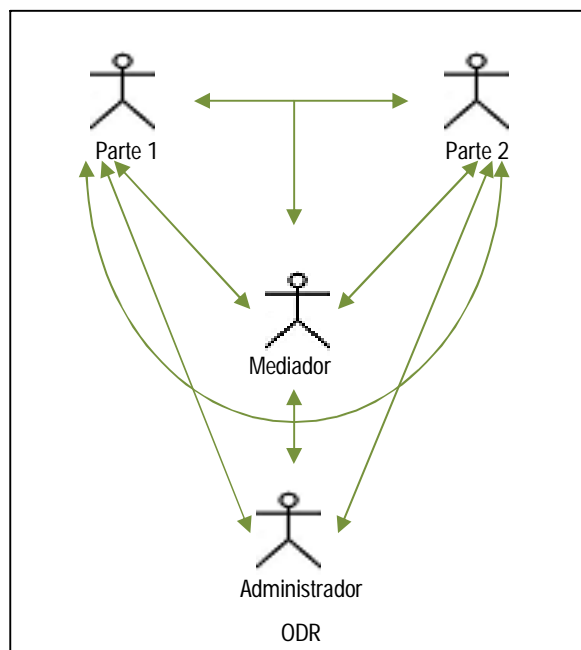


Fig 4: Mediação ODR

Finalmente, a arbitragem (figura 5) permite que ambas as partes interajam uma com a outra, mas sempre sobre o controlo do administrador ODR, bem como de um árbitro, as testemunhas estão também presentes para que possam existir mais dados relevantes para que o processo seja mais facilmente resolvido, existe também um conselheiro experiente no processo de arbitragem, cuja função é ajudar na resolução do problema com recurso à vasta experiência que possui.

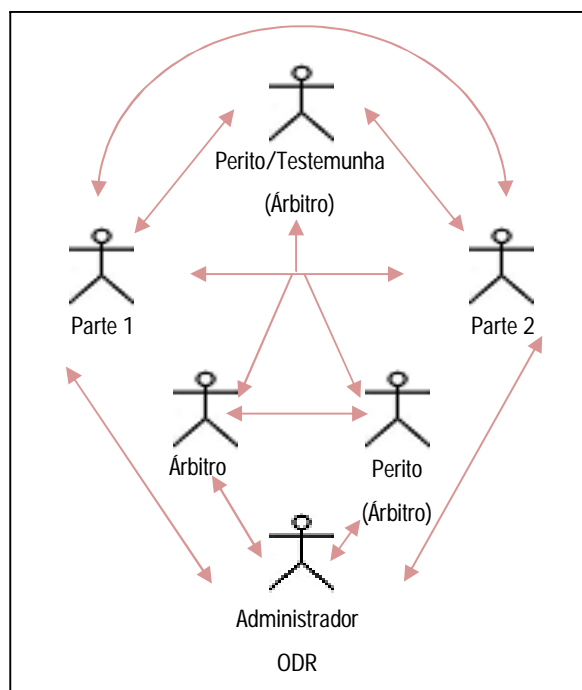


Fig 5: Arbitragem ODR

2.4 Inteligência Artificial no ODR

A Inteligência Artificial lida com o desafio de implementar comportamentos inteligentes em máquinas, com o objectivo destas imitarem os humanos em tarefas diárias, como por exemplo a capacidade de compreender e usar a linguagem, efectuar reconhecimento de imagens ou outros problemas. Entre as características que os investigadores esperam conseguir incutir nas máquinas encontram-se o raciocínio, o conhecimento, o planeamento, a aprendizagem, a comunicação, a percepção e a capacidade de mover e manipular objectos [24].

Tendo todas estas capacidades os computadores encontram-se no mesmo patamar que os seres humanos [25]. Na maioria das vezes, a inteligência não é alcançada por um único

agente inteligente, mas através da interacção de um grupo de agentes que comunicam. Este paradigma leva eventualmente ao conceito de Inteligência Artificial Distribuída (IAD) [26]. O uso desses agentes inteligentes no domínio específico da lei e de resolução de conflitos levanta muitas questões a partir do ponto de vista legal, nomeadamente na definição das relações entre os agentes e os partidos em nome dos quais agem [27].

No entanto, a pesquisa da IA levou ao desenvolvimento de muitas tecnologias que são utilizadas actualmente, na maioria das vezes na sombra dos grandes sistemas que constroem. Essas tecnologias são, em geral, utilizadas para otimizar o trabalho dos domínios do conhecimento, para tornar os produtos mais fáceis de utilizar através da adopção de interfaces inteligentes ou a automatização de tarefas. Entre os principais problemas atacados por IA, podemos enumerar alguns como por exemplo: a dedução, o raciocínio, a resolução de problemas, a representação do conhecimento, o planeamento, a aprendizagem, a inteligência social, os sentimentos ou a criatividade.

Tendo como foco o ODR verifica-se que há um número crescente de projectos nesta área, que actualmente utilizam técnicas de Inteligência Artificial. Em todos eles estas técnicas são usadas com o objectivo de imitar o comportamento de um especialista humano ou de uma das partes envolvidas. Por outro lado, alguns projectos não utilizam explicitamente técnicas de IA, mas revelam um comportamento inteligente.

Exemplo disto é o *Rule-based Legal Decision-making System (LDS)*, que é um projecto antigo, datado de 1980, e que foi um dos pioneiros dos sistemas de negociação apoiada [28]. O domínio deste sistema foi a lei de responsabilidade. Este campo do direito mantém distribuidores e fabricantes de produtos responsáveis pelos danos que os seus produtos podem causar. O projecto teve o objectivo de formalizar os processos de tomada de decisão de advogados envolvidos em litígios de responsabilidade do produto. Os autores formalizaram o conceito estrito de responsabilidade sobre a linguagem ROSIE, de modo a que o réu pode-se ou não ser considerado responsável.

O EXPERTIUS é outro sistema existente, trata-se de um sistema de suporte à decisão que aconselha juizes e oficiais de justiça mexicanos sobre a determinação de uma pensão financeira por uma das partes, ajudando também a determinar o valor dessa pensão [29]. O sistema possui três módulos principais: um módulo tutorial, um módulo de inferência e um módulo financeiro. O módulo tutorial orienta o juiz ou oficial de justiça através da realização de diversas tarefas. O módulo inferencial avalia as evidências baseadas em pesos que o utilizador atribui a

cada pedaço. Por fim, o módulo financeiro auxilia o utilizador no cálculo do valor das pensões de acordo com alguns critérios pré-existentes. Para efectuar o raciocínio nestes termos, o EXPERTIUS tem uma maneira de representar o amplo conhecimento sobre os vários parâmetros.

Nestes projectos de *ODR* a Inteligência Artificial possui um papel preponderante, pois um agente inteligente não é mais que uma entidade abstracta com a capacidade de realizar comportamentos inteligentes complexos. Este tipo de agentes ainda está em desenvolvimento e as suas habilidades encontram-se em crescimento, quando atingirem um nível mais elevado de desenvolvimento espera-se que tomem um papel mais importante e activo em todo o processo de *ODR*. Esse papel mais predominante acabará por levar aos sistemas de segunda geração *ODR* em que os agentes terão funções específicas. Espera-se, por exemplo, que os agentes sejam capazes de mediar conflitos sem intervenção humana, ou seja, que sejam capazes de ler o ambiente (as partes, o domínio do problema e as características, a lei e outros parâmetros) e resolver o litígio.

No entanto, para que os agentes consigam ter um papel mais activo e predominante na resolução de conflitos é fundamental que possuam boa capacidade de comunicação, o que lhes permite comunicar convenientemente com as partes envolvidas no litígio. Os agentes têm de estar dotados de mecanismos de representação de conhecimento capazes de armazenar os dados recolhidos durante todas as fases do processo (que podem incluir dados sobre as leis, o domínio do problema, os pontos em conflito, entre outros). Os agentes têm ainda de possuir a capacidade de tirar conclusões, definir estratégias ou dar conselhos às partes envolvidas, para tal o agente necessita de entender a linguagem natural, pois só assim consegue comunicar com os humanos, sendo neste ponto que entram as Ontologias.

Num futuro muito próximo o *ODR* em conjunto com a IA irá ajudar os tribunais na resolução de conflitos, numa fase inicial acredita-se que só processos pequenos vão poder ser resolvidos por sistemas *ODR*, mas quanto mais estes sistemas se especializarem mais tipos de conflitos serão capazes de resolver ou ajudar na resolução.

3 Web Semântica

Web semântica será o futuro da *Web* que conhecemos actualmente, pois com a informação organizada será mais fácil as máquinas interagirem entre si, através de ontologias, que é o que propõem a *Web* semântica.

3.1 O que é a Web Semântica?

Estando actualmente muito dispersa a informação presente na *Web*, a pesquisa é dificultada, vendo-se os utilizadores na necessidade de recorrer a uma selecção manual da informação. Nesse sentido começa-se a dar alguns passos, ainda pequenos, no sentido de catalogar essa informação existente nas páginas *Web*.

Sendo a *Web* semântica uma das tecnologias com maior potencial para a catalogação da informação da *Web*, muitos domínios ontológicos beneficiariam da sua utilização. Contudo neste capítulo irá ser abordado o domínio da economia. Neste capítulo pretende-se evidenciar as tomadas de decisão que foram usadas para a demonstração da tecnologia na investigação, assim como demonstrar o potencial do uso da *Web* semântica neste domínio.

Com a primeira abordagem à tecnologia, muitos sépticos poderão ser levados a pensar que com o uso de bases de dados normais, seria possível fazer-se o mesmo que está a ser feito com o recurso a documentos *RDF*. Contudo, com o uso de bases de dados normais não exista possibilidade de inferir novo conhecimento, limitando assim os utilizadores a poder somente

efectuar pesquisas, enquanto com a capacidade fornecida na tecnologia da *Web* semântica a inferência de novos conhecimentos a partir de factos e de regras já existentes revela-se bastante útil e importante.

3.2 Arquitectura Ideal para a Web Semântica

Na figura 6, pode-se observar como seria o modelo ideal de *Web* num futuro que poderá vir a ser bastante longínquo, pois para ser possível utilizar-se este modelo seria necessário efectuar-se uma reestruturação de toda a *Web* de modo a catalogar toda a informação actualmente existente.

Ao comparar-se a *Web* actualmente existente, com esta que se antevê, pode concluir-se que será dado um passo muito grande no que diz respeito à organização da informação agora existente na *Web*, fazendo com que esta passe a ser utilizada de forma inteligente. Estando a informação organizada, a pesquisa tornar-se-á mais precisa, deixando de lado os algoritmos de pesagem actualmente existentes (exemplo do *Google*), cuja finalidade é a de catalogar a informação, baseando-se no contexto sintáctico em que esta surge. No futuro essa catalogação será feita à custa do tipo de relações entre os diversos recursos, tornando-se assim a pesquisa mais fiável e com maiores níveis de precisão.

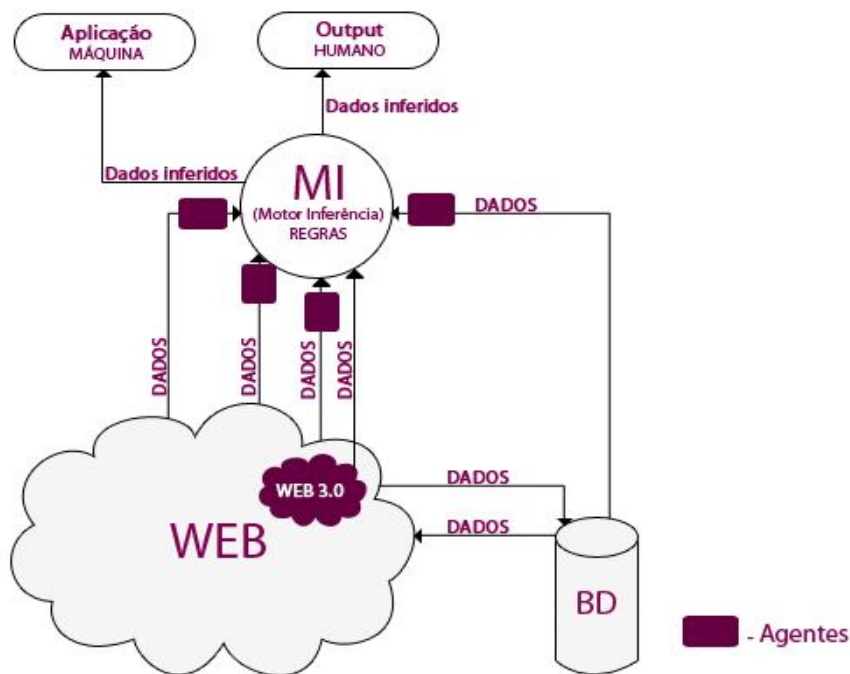


Fig 6: Modelo Web 3.0 adaptado a partir do modelo actual de Web

Observando a figura 7 conclui-se também que este modelo conceptual, considerado como “ideal”, tem como principal vantagem tornar a informação existente na *Web* interpretável pelas máquinas, tornando-a de certa maneira “inteligente”.

Pegando na *Web* totalmente catalogada, existirá a possibilidade de lhe inferir conhecimento com recurso a motores de inferência, motores esses que pegam em dados bem definidos, em regras estruturadas e comprovadas por indivíduos e geram novo conhecimento. Estes indivíduos são profissionais das várias áreas ontológicas, possuindo o conhecimento necessário e comprovado, para a implementação e descrição de regras capazes de inferir novo conhecimento.

Os motores de inferência pegam automaticamente nos dados provenientes de diversas fontes existentes na *Web*, e através da aplicação das várias regras ontológicas, têm a capacidade de poder inferir novo conhecimento, isto é, encontrando-se os dados bem estruturados e catalogados na *Web*, com regras bem definidas por indivíduos especializados, seria possível averiguar quais os factores que de uma forma directa ou indirecta poderiam influenciar a variação de um determinado caso, isto é, seria por exemplo possível prever o valor da empresa no mercado de acções. Esta é apenas uma das possíveis soluções e possibilidades que seria permissível ter perante uma *Web* semanticamente bem estruturada, com a informação descrita e detalhada num formato comum.

3.3 Arquitectura Actual para a Web Semântica

A tecnologia hoje disponibilizada para a sustentação da *Web* semântica, oferece já mecanismos para organizar e estruturar a informação. Com o modelo apresentado na figura 6, pretende-se demonstrar uma das formas possíveis para a integração da *Web* semântica na *Web* actual, sem a necessidade de a reestruturar, tornando-a assim uma arquitectura próxima da apresentada anteriormente.

Pela análise do modelo da figura 7, relativamente ao modelo descrito na secção 3.2, é possível destacar as caixas que interligam a *Web* actual ao motor de inferência, estas caixas são designadas por agentes. Um agente, tem como função recolher a informação existente na *Web* e tentar perceber o seu significado, de modo a inserir a informação recolhida num determinado

contexto. Estes dados são então utilizados posteriormente nos motores de inferência para a manipulação de conhecimento, conjugados com regras.

Da análise do modelo destaca-se também a existência de uma ínfima parte da *Web*, já com os dados devidamente estruturados e catalogados, num formato comum, nomeadamente no formato *RDF*, o acesso a estes dados seria feito de forma directa por parte do motor de inferência.

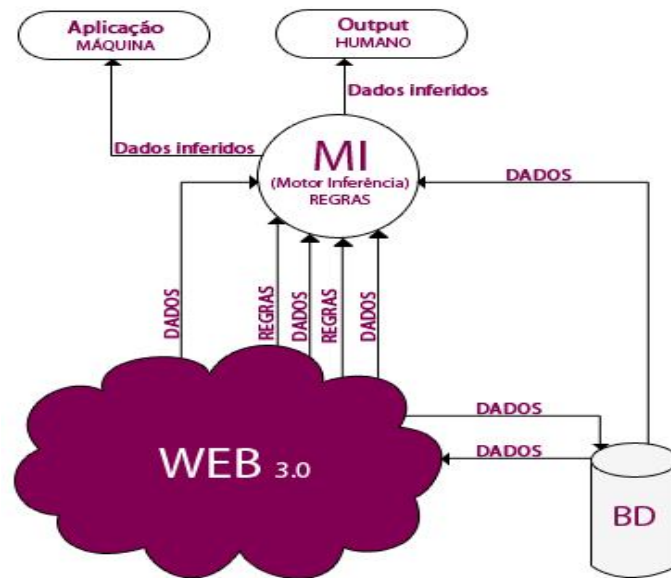


Fig 7: Modelo Conceptualizado da Web 3.0

É também possível observar que através da aplicação de agentes, estaríamos mais perto de uma *Web* não só de nível sintáctico mas também de nível semântico, isto é, não teríamos uma *Web* somente para a troca e partilha de dados, mas teríamos uma *Web* capaz de utilizar esses dados para gerar novo conhecimento que permitisse auxiliar na tomada de decisões, tanto na perspectiva do homem como da máquina.

As desvantagens adjacentes à utilização destes agentes para que seja possível adaptar a *Web* actual a uma *Web* semântica são bastantes, pois mesmo que se consiga ter esses agentes, a implementação não será trivial, existirá sempre a morosidade do processo de “alimentação” de dados aos motores de inferência por exemplo. Contudo, os agentes teriam dificuldades e fariam a filtragem de informação com um elevado grau de incerteza, o que poderia resultar na criação de conhecimentos enganosos, bem como temporalmente descontextualizados. O próprio processamento computacional requerido por este tipo de agentes e pelos motores de inferência é extremamente exigente e praticamente impossível de ser realizada em tempo real.

4 Ontologias

Ontologia é uma reunião de conhecimento compartilhado numa área específica. Portanto, esse conhecimento pode resolver problemas de comunicação entre as aplicações, sendo esse um dos grandes objectivos desta dissertação. Um estudo recente diz que as ontologias são acordos em matéria de conceitualizações compartilhadas, essas conceitualizações incluem estruturas de domínio do conhecimento e de modelação de conteúdos específicos dos protocolos utilizados para comunicação entre agentes.

As ontologias são usadas há vários anos pela inteligência artificial, mas, mais recentemente, começaram a ser utilizadas para a localização e recuperação de informação no âmbito das Ciências da Computação e das Ciências da Informação [30]. Podem ser também propostas para fornecer um entendimento comum e compartilhado de algum domínio que pode ser compreendido por pessoas e computadores [31], mas uma ontologia é uma investigação dos conceitos que torna possível às pessoas o conhecimento e determinação de objectos reais [32].

Uma ontologia não é mais que um modelo de dados que representa um determinado domínio do conhecimento. As ontologias assumem-se como muito importantes no âmbito da Web Semântica e têm vindo a assumir-se como cruciais em diversas áreas, das quais são exemplo a representação de conhecimento, a recuperação de informação, o processamento de linguagem natural, entre outros.

Pode também ser usado o termo ontologia para um conjunto de conceitos e termos interligados que são utilizados para descrever uma determinada área do conhecimento ou construir uma representação para o conhecimento [33].

A camada mais importante neste diagrama (figura 8) é a camada **Ontologia**, esta é constituída pelo vocabulário ontológico que permite atribuir significado a termos e teorias de uma determinada área de conhecimento, processando e integrando a informação existente sem problemas de indefinição ou conflito de terminologia. A linguagem *RDF Schema* permite a construção de ontologias simples e limitadas, enquanto a *Web Ontology Language*, mais reconhecida pelo acrónimo de *OWL*, estende o vocabulário do *RDF Schema* para a inclusão de elementos com maior poder de expressividade e de inferência.

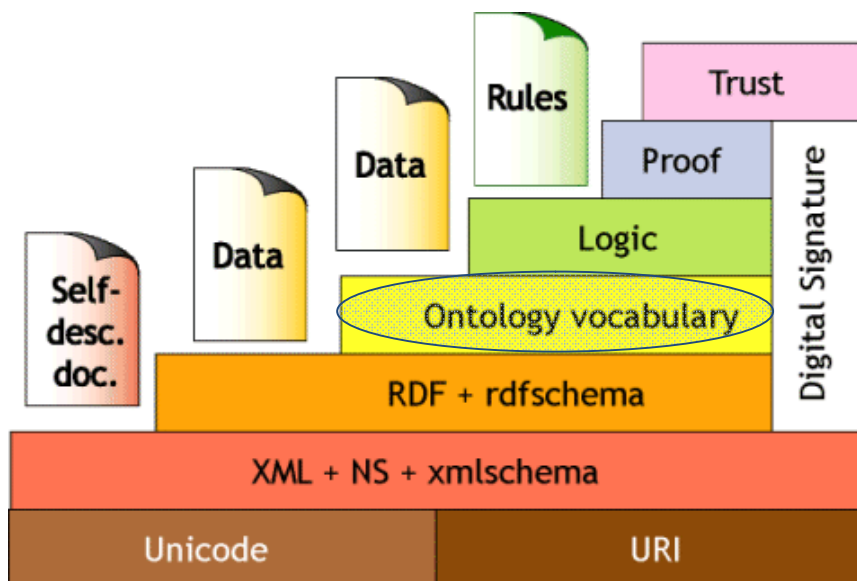


Fig 8: Definição do vocabulário ontológico

Actualmente existe uma grande necessidade de modelação, estruturação e articulação do conhecimento, de modo a apoiar adequadamente a sua integração flexível e apresentação personalizada para o consumidor. A resposta a estas necessidades são as ontologias, pois destinam-se a capturar o conhecimento e a fornecer uma compreensão comum de um determinado domínio, podendo esse conhecimento ser reutilizado e compartilhado.

Em [34] define-se ontologia como os termos básicos e os relacionamentos que constituem o vocabulário de uma área do conhecimento, bem como as regras para combinar termos e

relacionamentos para definir extensões ao vocabulário. Formalmente, uma ontologia pode ser representada por:

$$O = (C, H, I, R, P, A)$$

- C representa o conjunto de entidades da ontologia, ou seja representa as entidades a modelar no domínio. É designada por um ou mais termos da linguagem natural.
- H representa o conjunto das relações entre os conceitos, sendo que essas relações definem a hierarquia dos conceitos.
- I representa o conjunto de relacionamentos entre classes e instâncias.
- R representa o conjunto de relacionamentos que não são entre conceitos nem entre classes ou instâncias.
- P representa o conjunto de propriedades das entidades de uma ontologia.
- A representa o conjunto de axiomas, regras que permitem verificar a consistência da ontologia e inferir novo conhecimento.

4.1 Métodos de Desenvolvimento Ontológico

Existe a necessidade de ter uma metodologia rígida e disciplinada na criação de ontologias, desse modo existem diversas tarefas a executar até chegar ao desenvolvimento de uma ontologia.

Como não existe um método óptimo para o desenvolvimento de ontologias investigou-se algumas das metodologias já existentes de forma a seleccionar a mais adequada para criar a uma metodologia para o desenvolvimento da OntoLab.

4.1.1 Ontology Development 101

Em [35] é sugerida a *Ontology Development 101* como um procedimento que consiste num guia de tarefas a executar, essas tarefas têm a função de ajudar no desenvolvimento da ontologia, pois orientam esse desenvolvimento até chegar ao objectivo final, uma ontologia

completa e funcional. O modelo *Ontology development 101* é dividido em 7 etapas, descritas na figura 9.

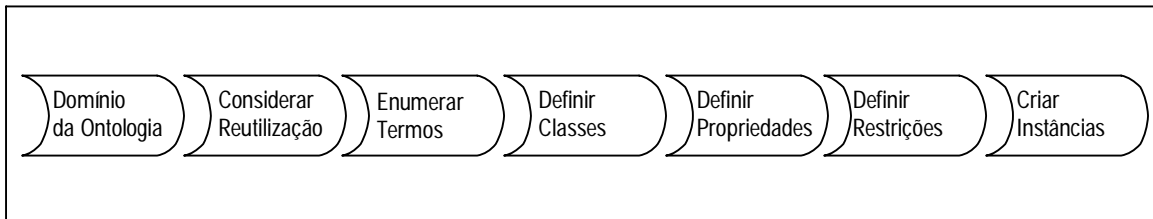


Fig 9: Procedimento de desenvolvimento *Ontology Development 101*

No domínio da ontologia pretende-se identificar com clareza os propósitos da ontologia, bem como os possíveis cenários de utilização. As respostas às questões “Qual a abrangência do domínio da ontologia?”, “Qual a sua utilização?”, “A que questões vai responder a ontologia?”, “Quais os utilizadores e quem irá manter a ontologia?”, ajudam a determinar qual o domínio da ontologia a desenvolver.

Deve-se considerar a reutilização de ontologias existentes, ou seja, é necessário verificar a existência de ontologias que possam ser reutilizadas no desenvolvimento do novo projecto, pois não existe necessidade de “reinventar” o que já está feito.

A enumeração dos termos presentes na descrição do domínio da ontologia é bastante importante para que seja possível, à posteriori, proceder à definição correcta de classes, de propriedades e de instâncias.

A definição das classes e a sua hierarquia é feita a partir da lista de termos, sendo que todos os termos que descreverem objectos são representados como classes. A hierarquia é criada após a definição de todas as classes quando se procede à verificação e selecção de quais são as classes que possuem um âmbito mais geral ou um âmbito mais específico.

Pegando nos termos restantes definem-se as propriedades, mas é necessário ter atenção para separar propriedades de relações entre classes. Caso seja uma propriedade é necessário definir o seu tipo de dados (texto, inteiro, percentagem), caso seja uma relação deve definir-se quais as classes que relaciona, verificar a cardinalidade e os valores válidos que pode ter.

Por fim, procede-se à criação de instâncias, entendidas como sendo população de uma determinada classe.

4.1.2 On-to-Knowledge

É uma metodologia (figura 10) de desenvolvimento criada pela cooperação de várias entidades europeias, sendo o seu principal objectivo desenvolver ontologias que tenham utilidade na área da gestão do conhecimento [36, 37].

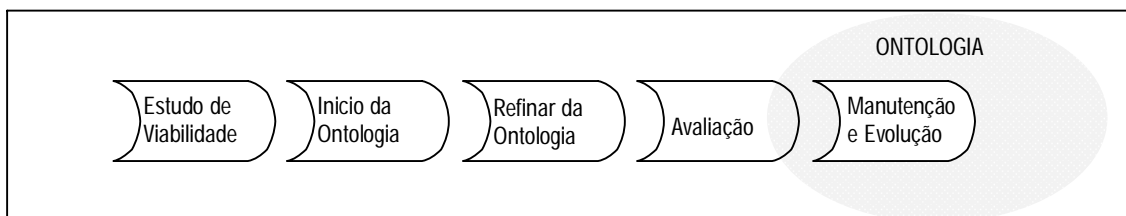


Fig 10: Processo de desenvolvimento *On-to-Knowledge*

Esta metodologia defende um estudo inicial de viabilidade, que pode ser efectuado com recurso ao *CommonKADS*, que é uma metodologia que fornece métodos úteis para efectuar uma análise detalhada do conhecimento, tarefas e processos. O estudo de viabilidade destina-se a mapear as reais necessidades do desenvolvimento de uma ontologia.

Após mapear as necessidades passa-se à produção de documentação com a especificação de requisitos, define-se o domínio e os objectivos da ontologia, identificam-se as fontes de conhecimento, definem-se as classes, as relações e as propriedades.

Como é necessário que a ontologia seja robusta, é fundamental que esta seja refinada com base na documentação proveniente da especificação de requisitos, o responsável pelo desenvolvimento da ontologia nesta etapa vai interagir com especialistas do domínio ontológico, moldando e estendendo a ontologia em busca de uma versão robusta.

Após refinar a ontologia, ou num primeiro momento criar a ontologia base, é necessário avaliar a ontologia desenvolvida, só assim se consegue verificar se esta se encontra de acordo com a documentação produzida na especificação de requisitos. Nesta etapa existe um constante recuo à etapa de refinar, até se atingir a ontologia pretendida e estabilizada.

Quando se desenvolve uma ontologia é necessário prever que esta necessita de manutenção e evolução ao longo do tempo, pois só assim continuará a ter utilidade.

A principal característica desta metodologia é a preocupação que é demonstrada nas etapas de estudo de viabilidade e início da ontologia, sendo estas as etapas chave para o desenvolvimento de uma ontologia estável.

4.2.3 Methontology

A *Methontology*, é uma metodologia de desenvolvimento criada por um grupo de investigadores da Universidade Politécnica de Madrid. Com duas grandes influências de áreas distintas: a Engenharia de Software e a Engenharia do Conhecimento, esta retira o melhor de cada uma delas [38].

Esta metodologia prevê um ciclo de vida baseado na prototipagem de ontologias, baseando-se na evolução das actividades de desenvolvimento (Especificação, Conceitualização, Formalização, Implementação, Manutenção). Esta metodologia baseada em ciclos de vida de *software*, tem como actividades privilegiadas as actividades de gestão, desenvolvimento e suporte do ciclo de vida da ontologia, como se pode observar na figura 11.

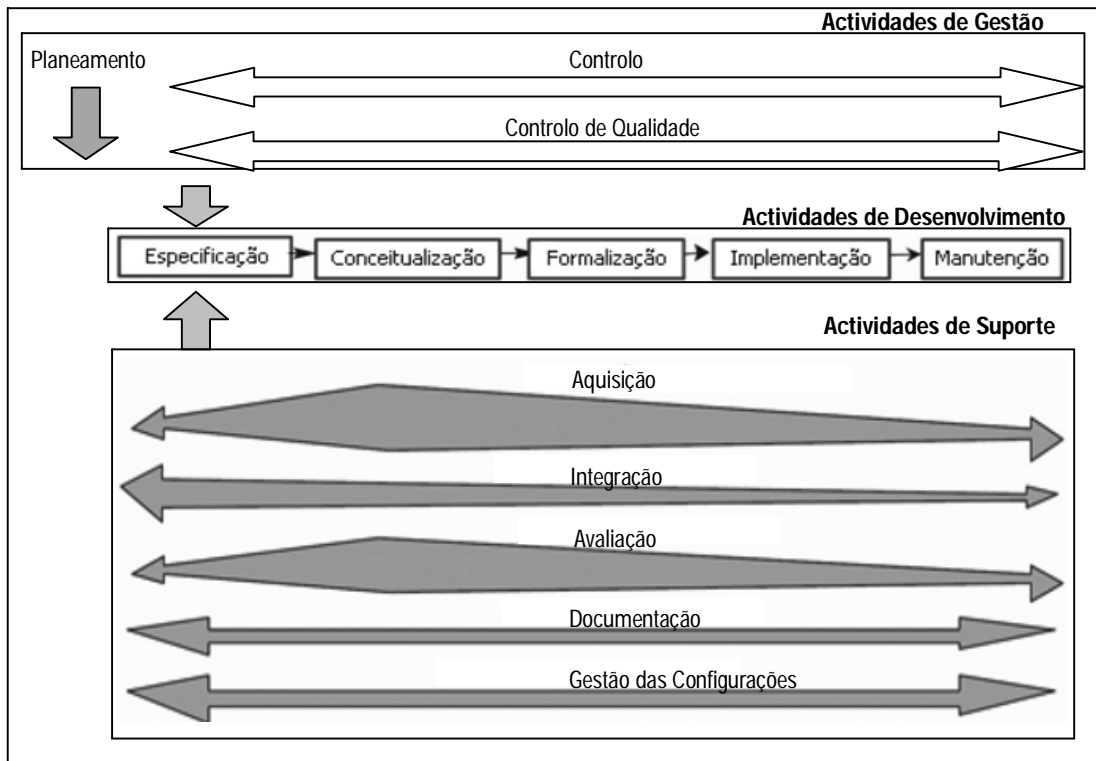


Fig 11: Processo de desenvolvimento e ciclo de vida da *Methontology*

4.2 Ferramentas de Desenvolvimento Ontológico

As ontologias podem ser representadas através de várias ferramentas, desde o *UML*, passando por uma qualquer linguagem orientada a objectos, *RDF*, *DAML + OIL*, *OWL* ou qualquer outra representação que possa definir os objectos, propriedades e suas relações. O *OWL (Web Ontology Language)* é considerado pela comunidade *Web* semântica como o melhor método para representação de ontologias.

As ontologias não são todas construídas da mesma forma. Existem várias formas de as representar que vão desde a lógica aos grafos. Ultimamente várias linguagens evoluíram para apoiar a definição de ontologias, como exemplo temos o *OKBC (Open Knowledge Base Connectivity)*, o *KIF (Knowledge Interchange Format)* ou a *FIPA (Foundation for Intelligent Physical Agents)*. Existem também várias linguagens com base numa forma de pensamento lógico para serem especialmente utilizadas por computadores, conhecidas como linguagens lógicas descritivas.

Quando se pretende comparar linguagens ontológicas, pretende-se que estas tenham requisitos como a simplicidade e a expressividade. A linguagem só precisa de ser tão rica e expressiva quanto o necessário para representar a complexidade do conhecimento, que é o grande propósito da ontologia [39].

O vasto leque de informações que residem na *Web* tem dado um impulso ao uso de ontologias e linguagens de ontologia, que dependem cada vez mais tecnologicamente do *W3C*, que em conjunto com a *DARPA* e a União Europeia está a seguir o desenvolvimento das linguagens nesta área. Ontologias fornecem a capacidade de dizer "o meu mundo é assim" e são a base que vai permitir que os programas possam "raciocinar" sobre diferentes ambientes fazendo conexões entre eles.

4.2.1 Protégé

O *Protégé* é uma ferramenta para criação e edição de ontologias e bases de conhecimento, de código aberto, desenvolvida pelo grupo de Informática Médica da Universidade de Stanford.

É uma ferramenta muito simples de utilizar, pois é muito visual, em contraste com as linguagens de programação existentes para a definição de ontologias, como por exemplo a linguagem *RDF (Resource Description Framework)* e *OWL (Web Ontology Language)*, ambas sucessoras de *DAML OIL*.

O *Protégé*, está muito próximo da metodologia *CommonKADS*, abrangendo uma grande colecção de métodos estruturados para construção de sistemas de Representação de Conhecimento, concentrando-se especificamente na modelação desse tipo de representação.

Esta ferramenta (figura 12) é a escolhida para efectuar o desenvolvimento da ontologia *OntoLab*, a possibilidade de exportar as ontologias desenvolvidas para a linguagem *OWL* ou *RDF*, trás uma mais-valia a esta ferramenta, existe também a possibilidade de mostrar graficamente as ontologias desenvolvidas através de grafos.

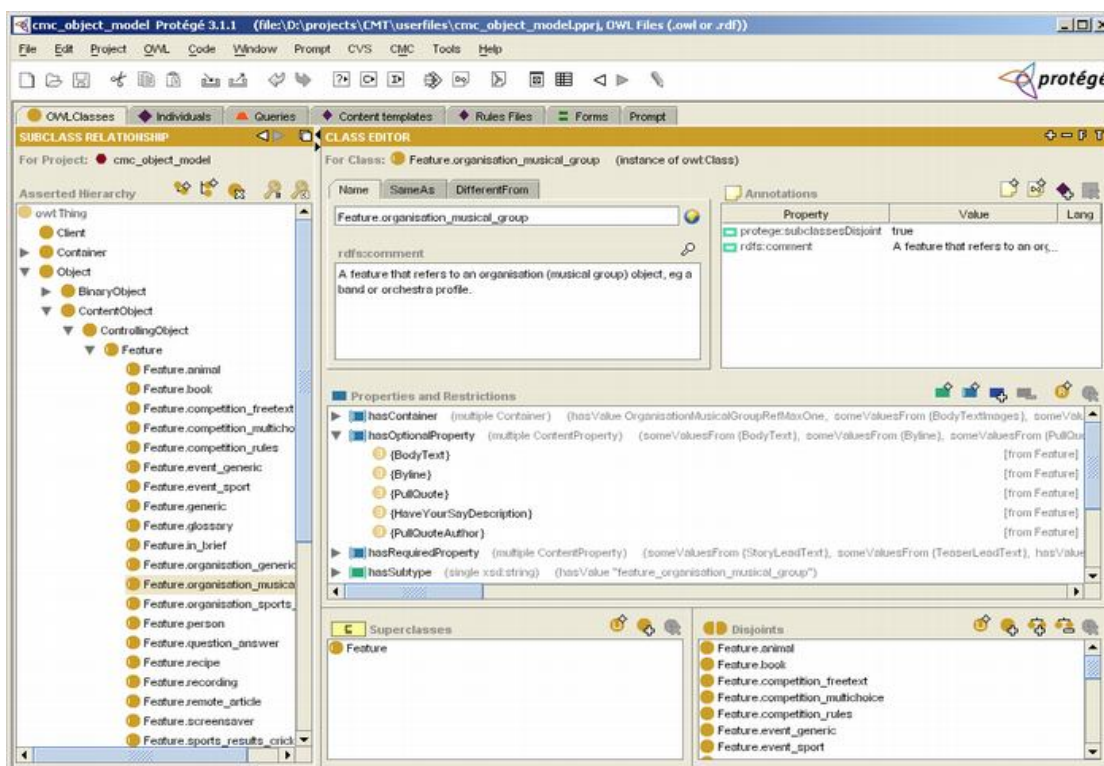


Fig 12: Protégé

4.2.2 eXtensible Markup Language (XML)

XML é uma linguagem *standard* de comunicação entre as aplicações. O uso de XML, que é uma linguagem estruturada, de fácil compreensão e grande usabilidade, torna possível a obtenção de níveis mais elevados de solidez na ontologia. Isso levou a que a ontologia criada em

Protégé fosse gravada num XML *schema*, pois com recurso a este XML *schema* é facilitada a inferência de conhecimento, bem como outras funcionalidades a implementar num *Web Service* de simulação de *ODR*, pois assim a informação está bem estruturada.

A utilização do XML foi, desde sempre, entendida como de elevada importância no desenvolvimento de ontologias, desta forma, para constatar a viabilidade da ideia pré concebida utilizaram-se alguns XML *schemas*, nomeadamente o *RDF* e o *OWL*, para verificar as funcionalidades que estes podiam trazer na construção da ontologia.

4.2.3 Resource Description Framework (RDF)

O *RDF* (*Resource Description Framework*) é uma plataforma inicialmente proposta pela *W3C* (*World Wide Web Consortium*), apresentando-se assim como uma nova forma de representação de informação na *WEB*. Esta descrição de dados é baseada na aplicação da linguagem XML (*eXtensible Markup Language*), destinando-se essencialmente a representar metadados de modo a que estes possam ser analisados e interpretados por máquinas, em vez de serem somente disponibilizados para as pessoas. Vários recursos podem ser descritos, desde serviços ou conteúdos na *WEB* a objectos do mundo real, como, por exemplo, pessoas. Este formato permite descrever em *RDF* qualquer conceito identificável, sendo assim uma boa plataforma para a gravação e partilha de conhecimentos na *WEB*. O *RDF* torna assim possível modelar informação de uma forma suficientemente consistente para que esta seja processada uniformemente por máquinas.

O *RDF* automatiza o processamento de recursos, tornando mais simples as trocas de conhecimento na *WEB*, recorrendo a *URIs*, que associados a elementos *RDF* se tornam semelhantes aos *links* das páginas *WEB*. A padronização deste formato, como se pode observar pela figura 13, permite que seja estabelecido um modelo de dados e de sintaxe para que seja possível representar, codificar e transmitir metadados, podendo ser estabelecida segundo dois documentos:

- **Resource Description Framework (RDF)**: tem a função de descrever o modelo de dados *RDF*; [40]

- **Resource Description Framework Shema Specification (RDFs):** funciona descrevendo primitivas de modelagem utilizadas para descrição de um domínio particular de interesse. [41]

```
<rdf:Description rdf:about="http://www.litigio.net/Empregado1">
  <rdf:type rdf:resource="http://www.litigio.net/Empregado"/>
  <pr:nome>Jorge Gonçalves</pr:nome>
  <pr:BI rdf:datatype="http://www.w3.org/2001/XMLSchema#int">12365700</pr:BI>
  <pr:NIF rdf:datatype="http://www.w3.org/2001/XMLSchema#int">12365700</pr:NIF>
</rdf:Description>
```

Fig 13: Descrição simplificada de um Empregado em RDF

O modelo *RDF* básico não representa uma linguagem mas sim um modelo baseado em vocabulário *URI* e numa sintaxe XML. A característica principal deste modelo assenta na forma como este utiliza metadados para a descrição de recursos, com o intuito de atribuir um significado ao recurso. A utilização de grafos permite a representação visual do *RDF*, utilizando os objectos abaixo referidos:

- **Resources:** Um recurso pode ser caracterizado por um objecto (físico ou lógico) desde que este possa ser identificável por um determinado *URI*, nomeadamente uma página *Web*, parte de uma página, várias páginas, um determinado elemento dentro de um HTML ou XML, ou também qualquer outro objecto de informação acerca de um conceito desde que este se encontre correctamente definido.
- **Properties:** Uma propriedade serve para descrever uma característica, um atributo ou uma relação qualquer utilizada num recurso, recorrendo ao *RDF Schema* ou *RDFs*.
- **Literals:** É um valor qualquer que um objecto pode assumir de acordo com a propriedade em questão.
- **Statements:** Fazem corresponder a um determinado recurso as suas propriedades bem como os seus respectivos valores, designando-se assim estas três partes respectivamente por sujeito, predicado e objecto.

É utilizada a indicação de um *triple*, (predicado, [sujeito], [objecto]), de forma a permitir a ligação entre os *literals* e os *resources*, cuja principal função é a de fornecer uma maior representação de modelos complexos. Um exemplo deste tipo é o seguinte (figura 14):

(Empregado, [http://www.galp.pt], Jorge)

Fig 14: Representação de um *triple*

Além de poder ser apresentado como um *triple*, o modelo *RDF* permite uma visualização através de grafos, convencionados pela *W3C*. Os *resources* são assim representados através de entidades ovais, as *properties* através de arcos e os *literals* por rectângulos. A figura 15 demonstra a representação gráfica do *triple* apresentado anteriormente.

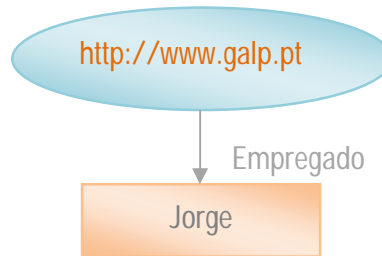


Fig 15: Representação de um *triple* sobre a forma de grafo

A representação gráfica do modelo *RDF* em grafos tem a capacidade de a um determinado recurso associar novas ramificações, de forma a especificar o modelo num contexto geral de uma forma mais detalhada. Esta característica encontra-se melhor representada na figura 16.

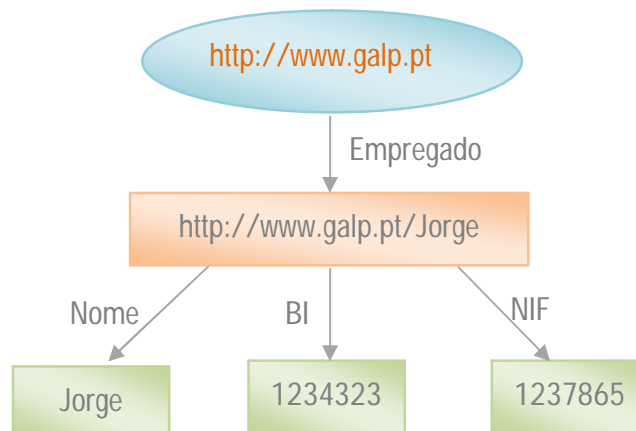


Fig 16: Grafo *RDF* mais detalhado

4.2.4 RDF Schema

O *RDF*, abordado anteriormente, foi caracterizado como um modelo padrão para descrição de recursos com propriedades. Porém a viabilidade para representar a interligação das propriedades com os outros recursos mostra-se bastante limitada. Com isso, surgiu o *RDFS* que se trata de uma extensão ao modelo *RDF*, que introduz primitivas básicas com a capacidade de permitir falar de classes, subclasses, domínio e restrições de valores das propriedades no contexto da *Web*. Permite assim que os utilizadores definam um vocabulário básico para dados *RDF* e especificar o tipo de objectos aos quais se aplicam esses atributos.

O *RDFS* proporciona um mecanismo para a criação de ontologias simples, utilizando para isso termos definidos como: *Class*, *subPropertyOf* e *subClassOf*. Os objectos *RDF* podem ser definidos como instâncias de uma ou mais classes usando a propriedade denominada *type*.

4.2.5 Web Ontology Language (OWL)

OWL não é mais que uma linguagem para definir e instanciar ontologias na *Web*, adiciona mais vocabulário para descrever as propriedades e classes. É também uma linguagem para representação do conhecimento.

A *OWL* é vista como uma tecnologia importante para a futura implementação da *Web* semântica. Esta tecnologia tem vindo a ocupar um papel importante num número cada vez mais significativo de aplicações e tem sido foco de pesquisa para ferramentas, técnicas de inferências, fundamentos formais e extensões de linguagem.

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

Fig 17: Descrição simplificada de uma propriedade em OWL.

A linguagem *OWL* é desenhada para ser utilizada por aplicações que necessitem de processar o conteúdo da informação, em vez de apenas o apresentar aos humanos. Possibilita

melhor interpretação pela máquina dos conteúdos *Web* que XML, *RDF*, e *RDFs* ao proporcionar vocabulário adicional a par com as semânticas formais. A linguagem *OWL* divide-se em:

- ***OWL Lite***: suporta aqueles utilizadores que necessitam principalmente de uma classificação hierárquica e de características de restrições simples.
- ***OWL DL***: suporta os utilizadores que querem expressividade máxima, sem perderem a totalidade de capacidade de decisão computacional dos sistemas de raciocínio. Foi desenhada para suportar o segmento de negócio existente e tem propriedades computacionais desejáveis para os sistemas de raciocínio.
- ***OWL Full***: suporta os utilizadores que querem expressividade máxima e liberdade sintáctica de *RDF*, sem garantias computacionais. É improvável que qualquer *software* de raciocínio seja capaz de suportar qualquer característica desta linguagem.

4.2.6 JENA

Jena trata-se de uma *Framework* de código aberto para a linguagem *Java*. Fornece uma *API (Application Programming Interface)*, para a linguagem de programação *Java*, que faculta as ferramentas e os meios necessários para o desenvolvimento de aplicações para a Web Semântica.

Proporciona assim um ambiente de programação para *RDF*, *RDFs* e *OWL*, *SPARQL* e inclui ainda um mecanismo próprio de regras, que permite efectuar a inferência de conhecimento sobre um determinado modelo.

Foi desenvolvida por Brian McBride, funcionário da HP (*Hewlett-Packard*), baseando-se num projecto denominado *SIRPAC API* que se encontrava já com alguns conceitos bem fundamentados.

A arquitectura da *API* de ontologias do *Jena* é composta por três camadas (ver figura 18).

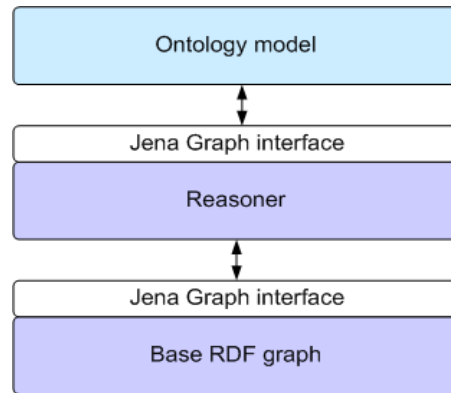


Fig 18: Arquitectura da API Jena

Todas as camadas apresentam os módulos básicos de modo a permitirem que a *API* seja vista mais genericamente.

A primeira camada designada por *Ontology Model*, contém todas as classes necessárias para trabalhar com ontologias descritas em *OWL*, *DAML OIL* ou *RDFs*. O módulo seguinte denominado *Reasoner*, permite inferir conhecimentos sobre modelos ontológicos nomeadamente *OWL* e *RDF*. O uso da inferência sobre modelos semânticos tem como objectivo permitir obter informação adicional sobre as ontologias. A camada designada *Base RDF graph* é responsável pela manipulação de ontologias.

Sendo importante referir que todas estas camadas comunicam entre si através da *Jena Graph Interface* que representa um modelo genérico de dados e que permite a comunicação entre as camadas/módulos da *API*.

A ferramenta *Jena* caracteriza-se pela capacidade de criação e manipulação de grafos *RDF*, representando-os pelos seus recursos (*resources*), propriedades (*properties*) e literais (*literals*), formando triples que irão originar os objectos criados pelo *JAVA*.

Na *Jena*, um grafo *RDF* denomina-se por modelo e é representado pela *interface Model*. Na figura 19 [42] pode-se observar a hierarquia das interfaces da *API Jena*.

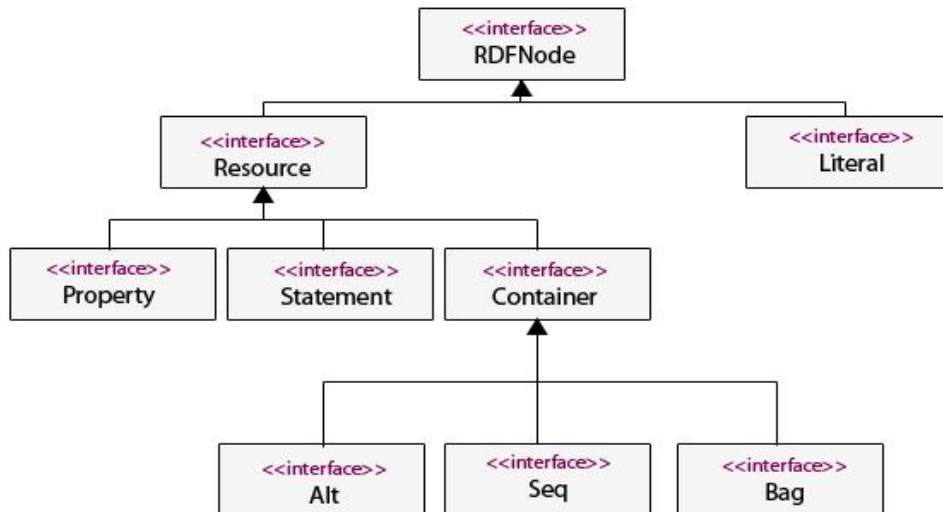


Fig 19: Hierarquia e relacionamentos das interfaces

A interface de programação que manipula e cria novos elementos é visualizada da seguinte forma:

- **RDFNode** tem a função de ligar todos os elementos do *RDF*, de modo a formar *triples* para uma perfeita criação dos elementos.
- **Resource** trata-se de um objecto que pode ser representado como um *URI*.
- **Literals** são utilizados como novos objectos, podendo também ser atribuições dentro de *triples* (*predicate*, *subject*, *object*) ou resultam de acessos a métodos que convertem *literals* em vários tipos de dados (*string*, *int* e *double*).
- **Properties** são as propriedades (*predicate*) inseridas nos *triples* utilizados.
- **Statement** é definida como uma expressão que pode dar origem a um novo triple.
- **Container** são conjuntos de objectos (contentores), apresentados de maneira adequada (*alt*, *bag* e *seq*) cujo principal objectivo é guardar os triples.

A classe central à inferência é o *Reasoner*, é a partir desta que se pode recolher conhecimento da ontologia, isto é, apesar da informação descrita no modelo poder ser bastante sintetizada, o *Reasoner* disponibiliza mecanismos baseados em regras, que permitem extrair conhecimento da ontologia. Essas regras têm que ser definidas em linguagem de primeira ordem, muito semelhantes à linguagem Prolog. O *Jena* disponibiliza dois *reasoners*, sendo estes,

o *forward chaining engine* e o *backward chaining engine*. Contudo a sua *API* permite ainda a integração de outros *reasoners* implementados por terceiros. [43]

- **Forward Chaining Engine:** é o raciocínio no sentido, factos → conclusões resultantes, pode ser apresentado como uma dedução.

(?A p ?B) -> (?A é ?C) - Verifica a regra (?A p ?B) e não chega ao objectivo.

- **Backward Chaining Engine:** é o raciocínio no sentido inverso, a partir de uma hipótese que tem que ser provada, até aos factos que suportam essa conclusão, como que se de uma indução se tratasse.

(?A p ?B) -> (?A é ?C) - Obtém todas as regras que satisfaçam (?A é ?C) e verifica que a condição não é verificada.

- **Hybrid Chaining Engine:** trata-se da junção entre os dois motores de inferência descrito anteriormente.

Para uma melhor compreensão e ilustração de como se podem aplicar as regras será utilizado o modelo ontológico de definição de partes num caso de *ODR*, como demonstrado na figura 20.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://OntoLab/relacao/" >
  <rdf:Description rdf:about="http://OntoLab/pessoa/Jorge">
    <j.0:Empregado rdf:resource="http://OntoLab/pessoa/Antonio"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://OntoLab/pessoas/Antonio">
    <j.0:Empregador rdf:resource="http://OntoLab/pessoa/Jorge"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://OntoLab/pessoas/Manuel">
```

```
<j.0:Colega rdf:resource="http://OntoLab/pessoa/Jorge"/>
</rdf:Description>
</rdf:RDF>
```

Fig 20: Definição de partes num caso

As regras enunciadas anteriormente, podem assumir o formato da figura 21, bem como reconhecer e estabelecer as restantes relações entre as pessoas descritas no modelo.

```
@prefix pess: http://OntoLab/pessoa/
[Regra1:
  (?p pess:Empregador ?q) -> (?q pess:Empregado ?p)
]
[Regra2:
  (?a pess:Empregador ?b)
  (?b pess:Colega ?c)
  ->
  (?a pess:Empregador ?c)
  (?c pess:Empregado ?a)
]
```

Fig 21: Definição de partes num caso

O uso da instrução *"@prefix pess: http://OntoLab/pessoa/"*, permite identificar o espaço de nomes que se está a abordar para a inferência de conhecimento.

A saída produzida após a inferência (figura 22), como se pode verificar, permitiu fazer com que o modelo ontológico ficasse mais enriquecido a nível de conhecimento.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://OntoLab/relacao/" >
  <rdf:Description rdf:about="http://OntoLab/pessoa/Jorge">
```

```

<j.O:Empregado rdf:resource="http://OntoLab/pessoa/Antonio"/>
<j.O:Colega rdf:resource="http://OntoLab/pessoa/Manuel"/>
</rdf:Description>
<rdf:Description rdf:about="http://OntoLab/pessoas/Antonio">
<j.O:Empregador rdf:resource="http://OntoLab/pessoa/Jorge"/>
<j.O:Empregador rdf:resource="http://OntoLab/pessoa/Manuel"/>
</rdf:Description>
<rdf:Description rdf:about="http://OntoLab/pessoas/Manuel">
<j.O:Empregado rdf:resource="http://OntoLab/pessoa/Antonio"/>
<j.O:Colega rdf:resource="http://OntoLab/pessoa/Jorge"/>
</rdf:Description>
</rdf:RDF>

```

Fig 22: Saída produzida após a inferência

A ferramenta *Jena* disponibiliza primitivas que permitem efectuar certas operações sobre os recursos e propriedades de um determinado modelo. Pois permite não só representar dados (factos), como também possibilita a representação de regras (o que não é possível numa base de dados convencional). Faculta também, através do motor de inferência, a aplicação de regras sobre os factos, com o objectivo de inferir novo conhecimento.

4.2.7 SPARQL

Existe um mecanismo, que pode ser utilizado em conjunto com esta *API*, que não é nada mais que consultas construídas sobre documentos XML/RDF que permitem visualizar os metadados como uma base de conhecimento, de forma a ser possível explorar a semântica dos dados num modelo *RDF* ou num modelo *RDF Schema*. O *Jena* possui assim classes para executar *queries* feitas em *SPARQL* e/ou *RDQL* sobre modelos ontológicos. Concentremo-nos portanto no *SPARQL*, tendo em conta que esta linguagem é mais avançada que o *RDQL*.

SPARQL é um acrónimo recursivo de *Query Language RDF* e foi padronizada pela *Data Access Working Group* (DAWG) da *World Wide Web Consortium* (W3C). A maior parte das utilizações do *SPARQL* refere-se a consultas em documentos RDF. Para este tipo de uso, o

SPARQL usa uma estrutura sintáctica semelhante à linguagem SQL, inclui características que permitem consultar triples, conjunções, disjunções e padrões facultativos.

O *SPARQL* trata-se assim de uma linguagem que permite invocação remota a ficheiros *RDF*, sendo assim especificada uma interface simples que pode ser apoiada tanto via *HTTP* ou *SOAP* de modo a que o cliente possa usar qualquer uma destas tecnologias para emitir consultas em *SPARQL*.

Nesta linguagem as variáveis são definidas por um prefixo "?" ou "\$" de modo a ser possível executar as pesquisas. O *SPARQL* procura triples que coincidem com quatro padrões, fazendo corresponder as variáveis da consulta às variáveis existentes em cada *triple*. Para as consultas serem exactas e curtas, a linguagem *SPARQL* permite definir prefixos com base em *URIs*, ou seja, com o uso de prefixos é desnecessária a repetição dos longos caminhos definidos pelos *URIs*.

O *SELECT* tem a função de devolver os registos seleccionados de um ou mais campos existentes no modelo *RDF* definido no prefixo. Existe também a primitiva *WHERE* que serve para definir as pesquisas a serem feitas pelo *SPARQL*. Existem outros comandos importantes, tais como: o *FILTER*, que permite filtrar os valores que vão ser devolvidos pelo *SPARQL*, após análise do código *RDF*; e a primitiva *ORDER BY*, que permite que os valores pesquisados sejam ordenados por um determinado campo do *RDF*.

5 OntoLab

A importância das ontologias na resolução de problemas do quotidiano encontra-se em franco crescimento. De facto, existe um crescente aumento do número de pedidos de criação de ontologias em certos domínios. Estas estão a ser utilizadas para melhorar a compreensão dos *Web Services* em áreas distintas, ou para fazer simples trocas de informações entre máquinas, para que estas interajam sem a necessidade de intervenção de um ser humano.

O mecanismo associado aos *Web Services* torna mais fácil a resolução de problemas relacionados com o direito do trabalho que é o foco desta tese. Trata-se de uma máquina que estará em interacção directa com as partes em litígio tornando o caso julgado mais claro, mais rápido e possivelmente mais barato.

A situação que se pretende retratar neste trabalho é um pequeno conflito entre empregador e empregado. Inicialmente procedeu-se à criação de dois protótipos: uma ontologia e um motor de pesquisa/inferência.

A ontologia, desenvolvida em *Protégé* e exportada para *RDF*, foi criada com as classes empregado, empregador, visitante, litígio e lei, sendo que a classe lei contém informações relativas aos artigos 127.º, 128.º e 129.º do Código do Trabalho Português, para que sirva como base de conhecimento para uma melhor resolução do conflito entre as partes. Para o desenvolvimento da ontologia utilizou-se a metodologia *Methontology*, por se tratar de uma metodologia mais completa.

O motor de inferência tem como objectivo possibilitar a inferência de novo conhecimento à ontologia, conhecimento adquirido com base em regras externas, e efectuar pesquisas ao conhecimento já existente na ontologia.

A ontologia desenvolvida visa servir de base para um sistema de *ODR*, sendo que a base do conhecimento será o direito laboral português, esta ontologia será *input* do sistema de inferência e consulta de dados também desenvolvido.

5.1 Planeamento da Ontologia

De acordo com a metodologia seleccionada, *Methontology*, o processo de desenvolvimento da ontologia OntoLab iniciou-se com a fase de planeamento, onde se planificaram as principais tarefas a executar, onde se definiu qual a sua organização e onde se verificaram quais os recursos necessários para o seu desenvolvimento (em termos de *software*, *hardware* e pessoas) (figura 23).

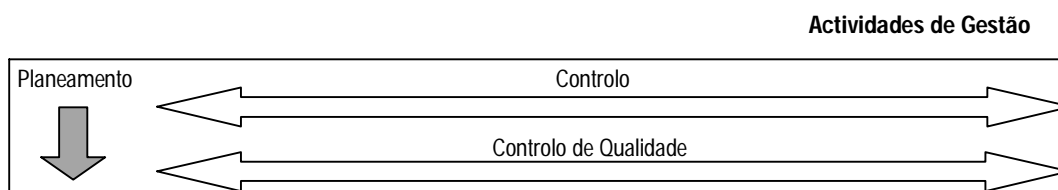


Fig 23: Actividades de planeamento da ontologia

Relativamente às tarefas a executar e à sua organização ficou definido o seguinte plano de trabalho, sendo que as mesmas foram executadas pela ordem apresentada:

- Investigação dos métodos e ferramentas de desenvolvimento ontológico a utilizar.
- Elaboração de uma lista de termos relevantes na área do direito laboral.
- Definição da gramática da ontologia, baseada na lista de termos previamente definida.
- Desenvolvimento de um protótipo da ontologia.
- Inserção de Leis, indivíduos e conflitos na ontologia.

No que diz respeito aos recursos necessários para o desenvolvimento da ontologia definiu-se que: relativamente ao *software* a ferramenta utilizada seria o *Protégé*; relativamente às

peessoas existiu a necessidade de constituir uma equipa multidisciplinar constituída por um programador e um especialista na área do direito do trabalho.

Ainda na fase de planeamento definiu-se as regras relativas ao controlo e ao controlo de qualidade, quanto ao controlo definiu-se que o mesmo ocorrerá ao longo de todo o ciclo de vida da ontologia de modo a evitar desvios ao cronograma inicial; quanto ao controlo de qualidade definiu-se que o mesmo é responsável por assegurar a qualidade de cada *output* produzido pela ontologia.

5.2 Desenvolvimento da Ontologia

Nesta secção definiu-se a lista de termos e criou-se uma gramática que servirá para organizar os termos definidos, de forma a facilitar a transformação destes em classes, propriedades e relações entre classes.

5.2.1 Especificação da Ontologia

A ontologia OntoLab foi desenvolvida com o intuito de resolver, com recurso às TIC, conflitos da área do direito do trabalho, tendo como finalidade integrar um sistema *ODR* que poderá ser utilizado tanto por empregados como por empregadores.

Desta forma, a OntoLab pretende minimizar o tempo, os recursos e o número de litígios em tribunal, através da simplificação da resolução dos conflitos de ordem laboral.

5.2.2 Gramática da Ontologia

Antes de criar a gramática que servirá de base para a ontologia, é necessário criar a lista de termos (figura 24) que será a base da gramática da ontologia. A lista de termos é criada a partir das fontes de conhecimento existentes, neste caso os artigos do código de trabalho e os dados necessários para que exista um conflito laboral.

Lista de termos OntoLab

- Pessoa
- Empregado;
- Empregador;
- Tipo de empregador;
- Nome;
- NIF (Número de contribuinte);
- BI (Número de Bilhete de Identidade);
- Especialista/Perito;
- Tipo de especialidade;
- Artigo;
- Alínea do artigo;
- Número de artigo;
- Descrição do artigo;
- Visitante;
- Litígio;
- Descrição do litígio;

Fig 244: Lista de termos

Da análise dos artigos do código do trabalho retirou-se termos como artigo, alínea e descrição; da análise de um processo de conflito laboral retirou-se termos como partes em litígio, entidade patronal, empregado, nome, número de bilhete de identidade, número de contribuinte (tanto para a entidade patronal como para o empregado), tipo de entidade patronal (sendo as opções singular ou colectiva) e, por fim, testemunhas de ambas as partes.

É importante referir que a ontologia também terá de contemplar dados relativos aos peritos/especialistas que podem ser necessários durante o processo de *ODR*, bem como as pessoas que podem consultar o processo durante o seu decorrer.

Após criar a lista de termos, passa-se à definição de uma gramática para tornar mais simples a visualização do que poderá ser representado em classes, em propriedades ou mesmo em relações entre classes. Para a representação da gramática (figura 25) vai-se utilizar um

método de representação próprio, onde cada produção (P) vai conter uma classe e os seus atributos/propriedades.

```
P1: Pessoa → Empregado
      → Empregador
      → Visitante
P2: Empregado → [Nome],[BI],[NIF];
P3: Empregador → [Nome],[BI],[NIF],[Tipo (Singular,Colectivo)];
P4: Visitante → Especialista
      → Testemunha
P5: Especialista → [Nome],[BI],[Especialidade];
P6: Testemunha → [Nome],[BI];
P7: Lei → Artigo;
P8: Artigo → [Nr],[Alínea],[Descrição];
P9: Litigio → [Descrição];
```

Fig 255: Gramática da ontologia

Nesta pequena gramática, com nove produções, pode observar-se a disposição de todos os termos. Esta gramática vai servir como protótipo da ontologia a ser implementada, tentou-se que esta fosse o mais simples possível, mas sem nunca perder de vista a robustez, de modo a explorar mais aprofundadamente as capacidades das ferramentas a utilizar.

Analisando a gramática pode observar-se que existem nove classes, uma por cada produção (*Pessoa*, *Empregado*, *Empregador*, *Visitante*, *Especialista*, *Testemunha*, *Lei*, *Artigo* e *Litigio*), das nove classes existem seis subclasses (*Empregado*, *Empregador*, *Visitante*, *Especialista*, *Testemunha* e *Artigo*), algumas propriedades (*Nome*, *BI*, *Alínea*) e restrições às propriedades, como é o caso das restrições *Singular ou Colectivo*, pertencentes à propriedade *Tipo*.

5.2.3 Classes

Classes OWL são interpretadas como conjuntos de indivíduos. As classes são descritas usando linguagem formal e as descrições devem indicar com precisão os requisitos para a adesão da classe. Por exemplo, a classe Empregado deverá conter todos os indivíduos que são Empregados no domínio de interesse. As classes podem ser organizadas numa hierarquia, que também é conhecida como uma taxonomia.

As subclasses são classes especializadas da super classe, no caso da subclasse Empregado, poderia existir uma super classe Pessoa, retratando que todos os Empregados são pessoas, logo todos os membros da classe Empregado pertence à classe Pessoa. A figura 26 mostra a representação de algumas classes contendo indivíduos - classes são representados como círculos, um pouco à semelhança dos conjuntos em diagramas de Venn.

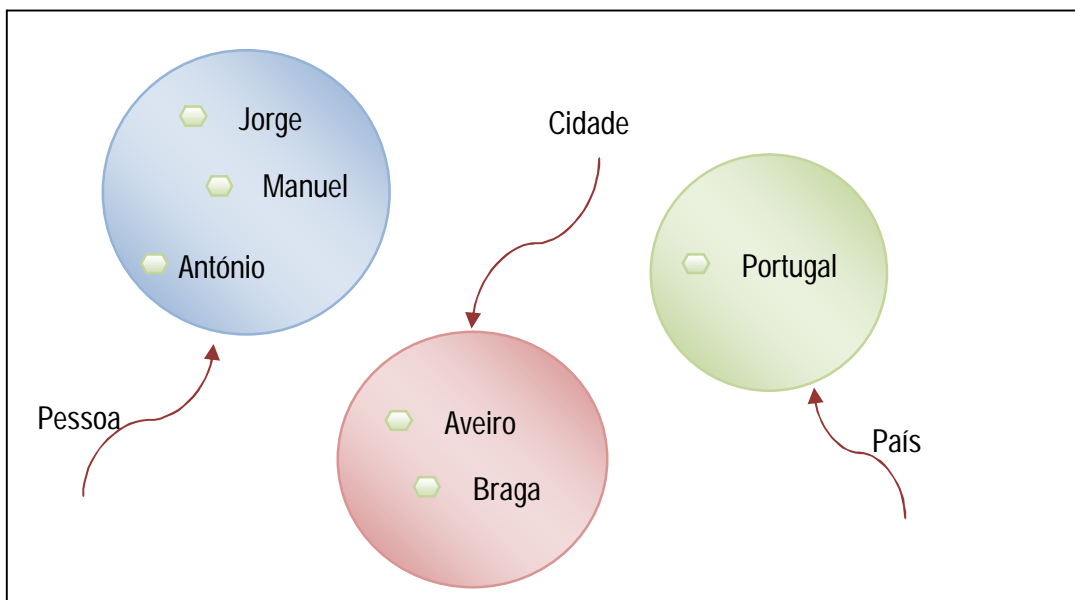


Fig 266: Representação de Classes

Seguindo a gramática criada para servir de guião na criação da ontologia, temos as nove classes representadas em *Protégé*. Na figura 27 podemos ver representadas as três super classes: classe Pessoa, classe Lei e classe Litígio.

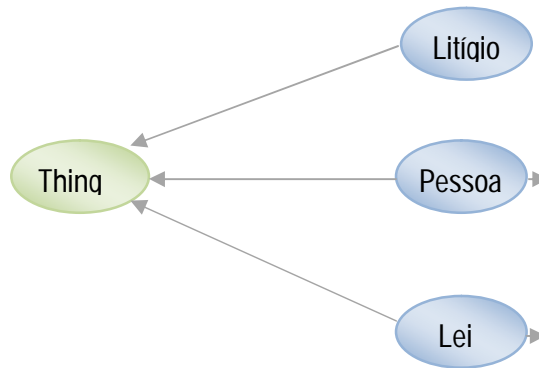


Fig 27: Super classes da OntoLab

A super classe Pessoa (figura 28) vai ter como subclasses as classes Empregado, Empregador e Visitante, estas classes herdam as características da classe Pessoa, a classe Empregado é disjunta da classe Empregador, por sua vez a classe Visitante pode ser tanto um empregado, como um empregador de outra empresa, ou pode ser mesmo uma pessoa fora do círculo empresarial.

A super classe Pessoa foi escolhida pois facilita a expansão da ontologia, um dia mais tarde, caso se pretenda alargar a índole desta ontologia, por exemplo para o direito da família, bastava criar duas subclasses que herdassem as características da classe Pessoa, algo que seria rápido de fazer.

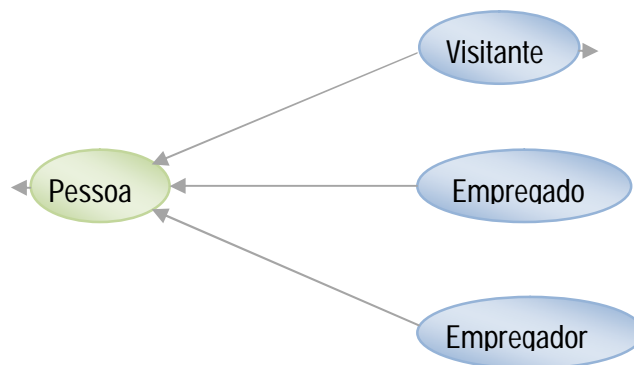


Fig 28: Classe Pessoa e Subclasses associadas

Na figura 29, podemos observar que a classe Visitante é subdividida em duas, sendo uma a classe Especialista e outra a classe Testemunha. A classe Especialista será utilizada para

definir visitantes que tenham alguma especialidade que possa ser utilizada no decorrer o litígio, podem ser especialistas de informática, que num litígio relacionado com fugas de informação são chamados para ajudarem na busca de provas e analisarem as existentes. A classe testemunha serve para definir indivíduos que sejam testemunhas no litígio, as testemunhas num sistema de *ODR* são vistas da mesma forma que num tribunal comum.

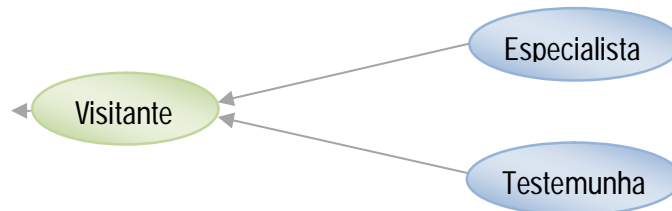


Fig 29: Classe Visitante e Subclasses associadas

A classe Lei (figura 30) será o suporte de todos os artigos do código do trabalho, terá o número do artigo, a alínea e a descrição de cada alínea.

Neste momento, esta classe só possui a informação relativa aos artigos 127.º, 128.º e 129.º do Código do Trabalho Português, pois estes foram os artigos considerados mais relevantes para iniciar este protótipo. Nestes artigos temos a informação relativa aos direitos e deveres dos Empregados e Empregadores, que deve ser utilizada como fonte para a resolução do litígio.

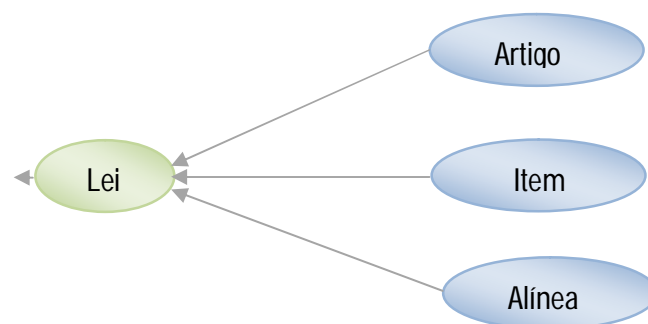


Fig 30: Classe Lei e Subclasses associadas

Na figura 31 pode observar-se totalmente o protótipo da ontologia desenvolvida, trata-se de um modelo simples que tem como objectivo demonstrar a utilidade das ontologias nos sistemas *ODR* e o código *RDF* gerado que serve de base ao motor de inferência desenvolvido.

Pois neste motor, com base nos dados existentes e com regras de inferência bem definidas, podemos acrescentar mais conhecimento ao modelo ontológico, que é o pretendido.

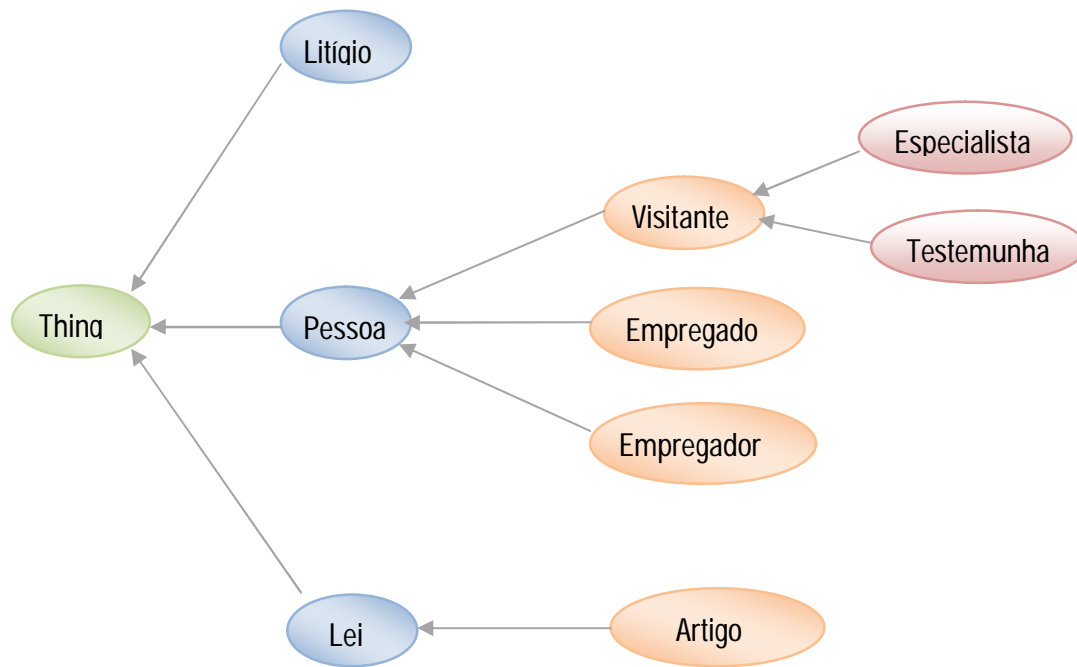


Fig 31: Estrutura de Classes da OntoLab

O modelo hierárquico de classes (figura 32) é algo bastante útil, uma vez que permite visualizar hierarquicamente todas as classes de determinada ontologia. A ferramenta utilizada para desenvolver a ontologia OntoLab possui esta funcionalidade. Assim sendo, o *Protégé* permite ao programador visualizar uma hierarquia de classes, para que este consiga ter uma visão global do modelo ontológico criado até ao momento, esta ferramenta permite ainda visualizar grafos da ontologia e navegar através das classes existentes.

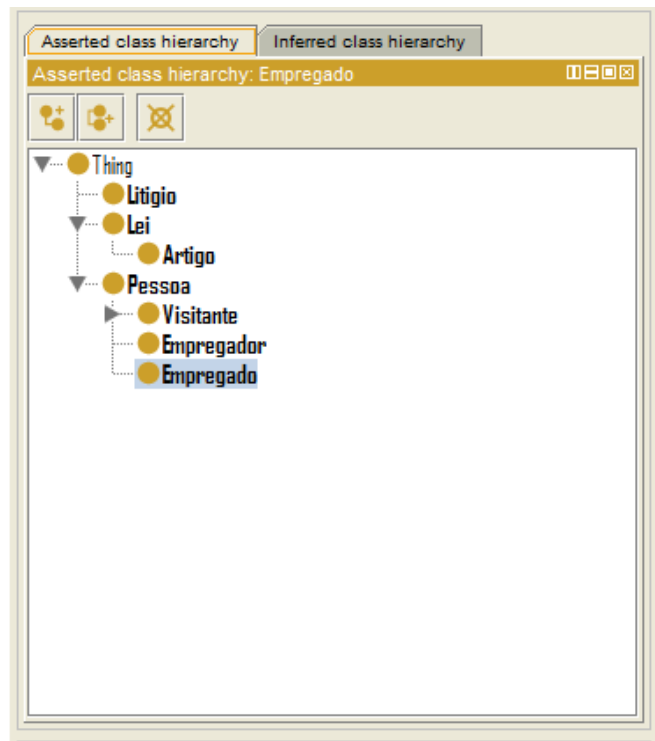


Fig 32: Hierarquia de Classes da Ontologia OntoLab (*Protégé*)

5.2.4 Indivíduos

Indivíduos não são mais que instâncias de classes que representam objectos num determinado domínio.

Uma diferença importante entre *Protégé* e *OWL*, é que *OWL* não utiliza o nome exclusivamente, ou seja, dois nomes diferentes podem, na verdade, referir-se ao mesmo indivíduo. Por exemplo, "*Queen Elizabeth*", "*A Rainha*" e "*Elizabeth Windsor*", são três nomes distintos mas que se referem a um mesmo indivíduo.

Assim, em *OWL*, deve ser explicitamente declarado que os indivíduos são os mesmos, ou diferentes uns dos outros, caso contrário o sistema não os identifica como sendo iguais ou diferentes entre si.

A figura 33 mostra a representação de alguns indivíduos pertencentes ao domínio da OntoLab.

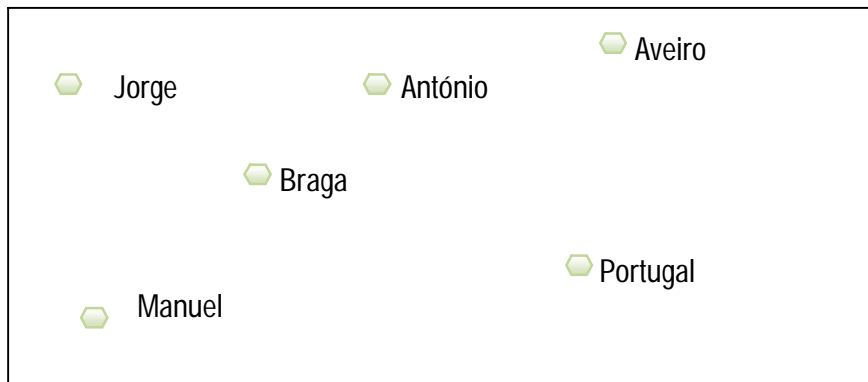


Fig 33: Representação de Indivíduos

No *Protégé* pode-se visualizar a quantidade de indivíduos por classe, bem como quais os indivíduos pertencentes a cada classe. Na figura 34 consegue-se rapidamente verificar que existe um indivíduo do tipo Empregado, um do tipo Empregador, um Litígio e uma Testemunha, sendo que três indivíduos pertencem à classe Pessoa.

Consegue-se também saber que o indivíduo pertencente à classe Testemunha é o Manuel, pois este é o único indivíduo pertencente à classe Visitante e como foi visto anteriormente esta classe é uma subclasse da classe Testemunha.

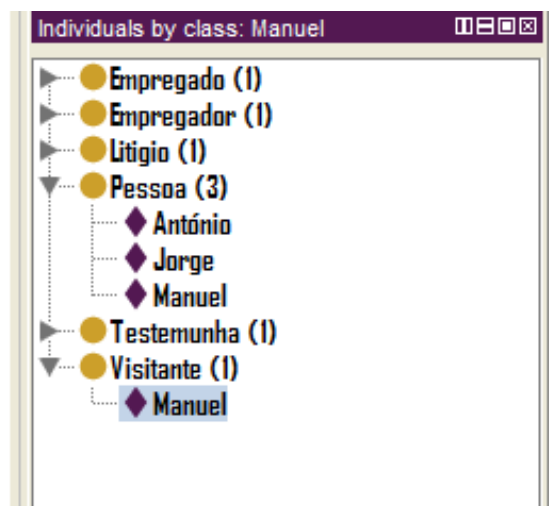


Fig 34: Distribuição de Indivíduos por Classe

É ainda possível verificar quais as propriedades que um indivíduo possui, ou seja, observando a figura 35, podemos verificar que o indivíduo Manuel, tem o valor "435321" na

propriedade BI e na propriedade Nome tem o valor "Manuel". É ainda possível verificar que este indivíduo pertence à classe Testemunha, Visitante e Pessoa, que como vimos anteriormente são classes pertencentes à mesma hierarquia. Pela análise das propriedades do indivíduo verifica-se que é um indivíduo semelhante ao Jorge e diferente do António.

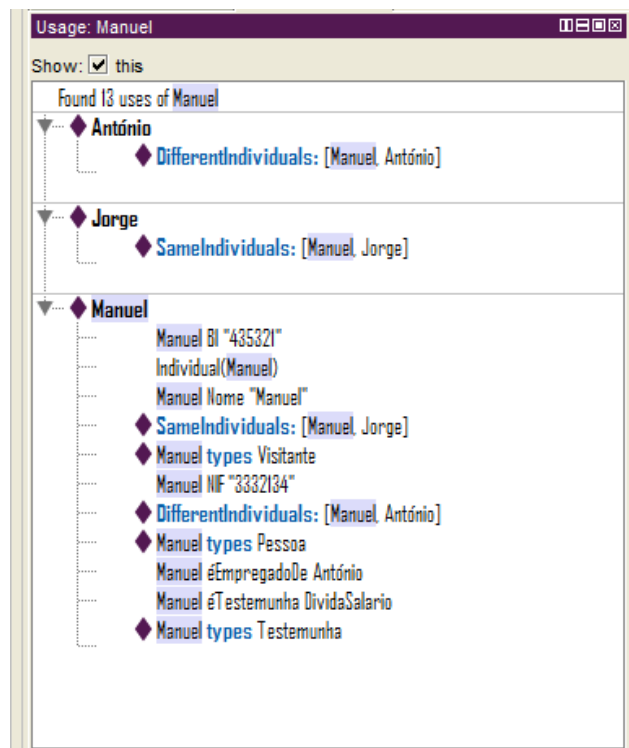


Fig 35: Representação do Indivíduo Manuel

5.2.5 Propriedades

Propriedades são relações binárias sobre os indivíduos, ou seja, as propriedades servem para ligar dois indivíduos. Por exemplo, a propriedade *éNaturalDe* faz a ligação entre o indivíduo Jorge e o indivíduo Aveiro, ou por exemplo a propriedade *éEmpregadoDe* vincula o indivíduo António ao indivíduo Jorge.

No entanto, as propriedades podem também ter o inverso, ou seja, pegando no exemplo da propriedade *éEmpregadoDe* que liga os indivíduos António e Jorge, sendo o primeiro Empregado do segundo, se fizermos uma propriedade inversa, do tipo *éEmpregadorDe*, podemos dizer que o indivíduo Jorge é Empregador do indivíduo António.

As propriedades podem também ser limitadas a um único valor, para que sejam funcionais, a simetria ou a transitividade são outras características associadas às propriedades. A figura 36 representa alguns exemplos de propriedades que ligam indivíduos.

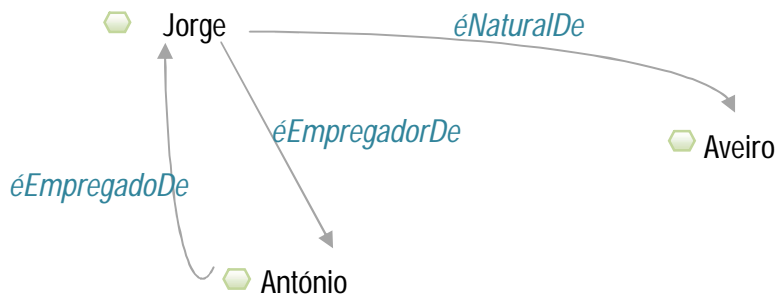


Fig 36: Representação de Propriedades

Tal como nas classes, nas propriedades o *Protégé* também tem uma hierarquia de propriedades, neste caso em particular todas as propriedades definidas estão no mesmo nível da hierarquia.

Existem dois tipos de propriedades que podem ser descritas em qualquer ontologia, as propriedades de dados, que servem para caracterizar um indivíduo e as propriedades dos objectos, que servem para ligar indivíduos.

A propriedade *TipoEmpregador* (figura 37), pertence às propriedades de dados e serve para definir se o Empregador é singular ou colectivo, trata-se de uma propriedade que tem um intervalo de valores associados.

Na *DescricaoLitigio* temos a descrição total do litígio, ou seja, é nesta descrição que estão incluídos os dados fornecidos pelas partes e pelos visitantes.

A propriedade *Nr* está relacionada com a classe *Artigo* e tem referência aos números dos artigos do código do trabalho, sendo que as propriedades *Alínea* e *DescricaoArtigo* estão também inteiramente relacionadas com a classe *Artigo*, pois a primeira tem as alíneas relativas ao artigo e a segunda a descrição das alíneas.

As propriedades *Nome*, *BI* e *NIF* pertencem à classe *Pessoa*, que por sua vez as atribui às suas subclasses.

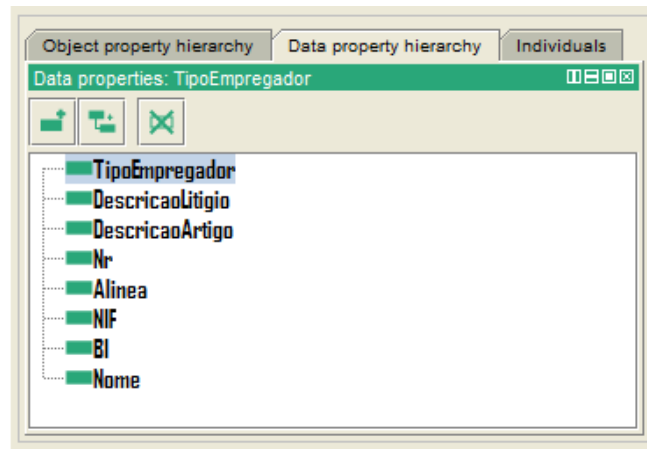


Fig 37: Representação de propriedades dos dados

As propriedades dos objectos servem para ligar indivíduos, no caso da propriedade *éArguido* e *éReu* estas servem para relacionar um indivíduo pertencente à classe Pessoa e um indivíduo pertencente à classe Litígio. Por sua vez as propriedades *éEmpregadorDe* e *éEmpregadoDe* servem para ligar dois indivíduos pertencentes à classe Pessoa, permitem definir quem é o Empregado e quem é o Empregador.

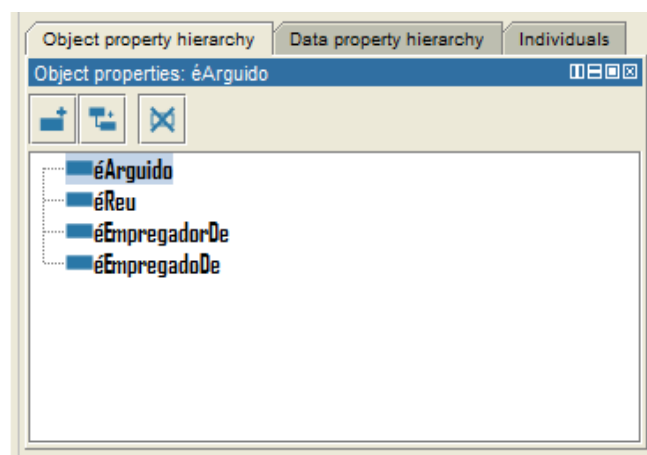


Fig 38: Representação de propriedades das classes

5.3 Motor de Inferência

O desenvolvimento desta aplicação experimental tem como princípio a demonstração do uso da *API* fornecida pela *Framework JENA*, no desenvolvimento de aplicações em linguagem *Java*.

A aplicação permite também proporcionar aos seus utilizadores um melhor ambiente para a manipulação de modelos *RDF*, quanto à sua inferência e consulta. A consulta destes modelos passa pela utilização de *queries* definidas na linguagem *SPARQL* e pelo uso de regras disponibilizadas na *API* para a inferência de conhecimento.

Este motor de inferência tem como objectivo permitir adicionar novo conhecimento à ontologia, sempre que tal se justifique ou se torne necessário.

5.3.1 Opções de Implementação

Para o desenvolvimento desta aplicação utilizou-se o *NetBeans*, um *IDE* (Ambiente de Desenvolvimento Integrado), desenvolvido pela *Sun Microsystems*. A sua escolha deveu-se sobretudo à rapidez e facilidade de desenvolvimento de uma aplicação *Java*, mas também pelo facto de se tratar de um *IDE* disponibilizado gratuitamente.

Na base de desenvolvimento da aplicação foi utilizado o modelo de projecto *Java Desktop Application* fornecido pelo *IDE NetBeans*. Neste modelo de projecto encontra-se implementado um esqueleto de uma aplicação para o padrão *Swing Application Framework*. O *Swing Application Framework* é uma *Framework* leve e simples de manusear, que permite a criação de aplicações *desktop* em *Java*.

5.3.2 Implementação da Aplicação

Na janela principal da aplicação (figura 38), encontram-se disponibilizados os meios necessários para carregar o conteúdo de ficheiros no formato *RDF* e *RULES*, para áreas de texto distintas. A utilização de áreas de texto para a apresentação do conteúdo dos ficheiros antevê a capacidade de poderem ser feitas alterações nos ficheiros, podendo essas alterações serem

guardadas no mesmo ficheiro ou num novo. Para guardar as alterações efectuadas são disponibilizados dois botões.

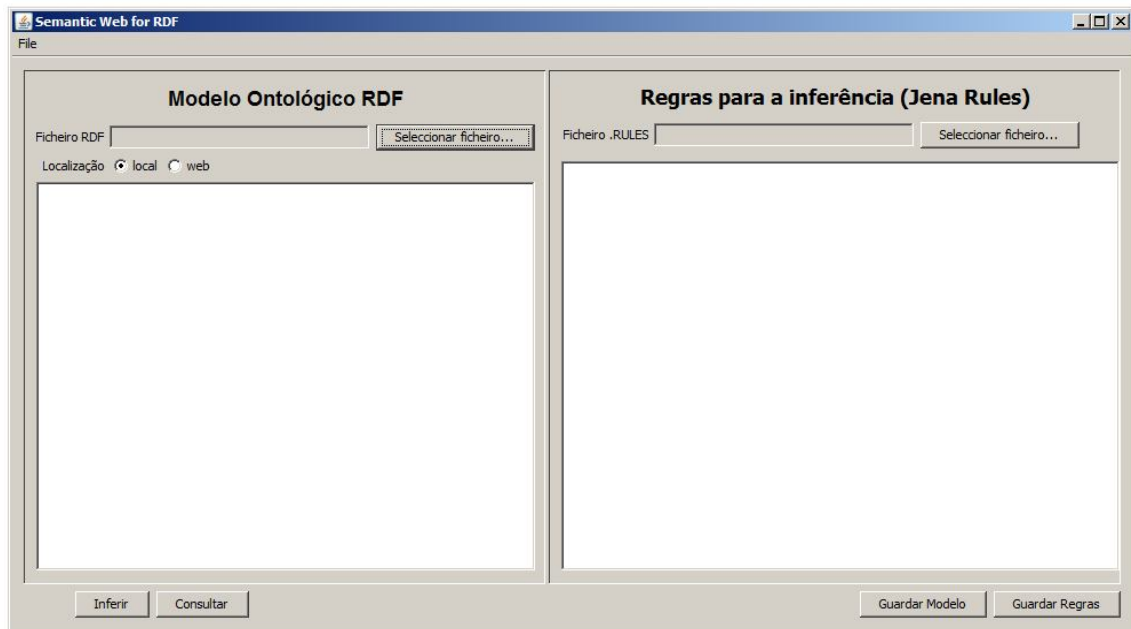


Fig 39: Janela principal do motor de inferência da OntoLab

Após a criação da Ontologia no *Protégé* existe a possibilidade desta ser gravada em *RDF*, tudo que criamos visualmente no *Protégé*, desde classes, propriedades e mesmo indivíduos é retirado em formato *RDF* para poder ser entendido neste protótipo de inferência.

Existe uma classe na *API* da *Jena* que permite armazenar o modelo ontológico em memória, trata-se portanto da classe *Model*. Quando se carrega no botão denominado "Seleccionar Ficheiro..." (*jButton2*) é criada uma instância do tipo *Model*, para onde o ficheiro *RDF* é carregado pelo uso dos métodos fornecidos na *API*.

Foi declarada inicialmente uma variável designada *modeloRDF* do tipo *Model*, onde será guardado o modelo ontológico. Como se pode verificar pelo código apresentado abaixo, com o método *createDefaultModel()* da class *ModelFactory* da *API*, criou-se uma instância auxiliar do tipo *Model*, esta é assim também uma forma de criar modelos em memória. Com a instrução:

```
model=FileManager.get().loadModel(caminhoRDF);
```

Pretende-se efectuar o carregamento do ficheiro *RDF* localizado no endereço *caminhoRDF* para o modelo em memória *model*.

Com o sistema de *try* e *catch* (figura 39) que a linguagem Java disponibiliza para capturar excepções, pode-se avaliar se o carregamento de um determinado modelo *RDF* foi efectuado com sucesso ou não.

```
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
    (...)
    String caminhoRDF = jTextField1.getText().toString();
    Model model = ModelFactory.createDefaultModel();
    try{
        model = FileManager.get().loadModel(caminhoRDF);
        (...)
        modeloRDF = model;
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null,"Erro a carregar o modelo, por favor verifique a
sintaxe do ficheiro.");
        (...)
    }
}
```

Fig 40: Sistema *try* e *catch* para avaliar se o modelo foi carregado

Uma vez que já se tem o modelo carregado em memória, apresenta-se agora uma das formas de inferir conhecimento utilizando para isso os métodos disponibilizados na *API* e o ficheiro de regras previamente seleccionado.

Foi definido um método denominado *runEngine*, que recebe como principais parâmetros a localização do ficheiro *Rules* e o modelo *RDF* (figura 40). Criou-se uma instância da classe *Writer* denominada *out*, para onde será enviado o modelo *RDF* já com os novos conhecimentos inferidos a partir das regras. Neste método é utilizada uma instância *configuration* do tipo *Resource*, onde serão adicionadas as propriedades necessárias para a configuração do *reasoner*, isto é, o *Resource configuration* que é responsável por indicar qual o mecanismo de inferência a usar (*Forward*, *Backward* ou *Hybrid*), e qual a localização do ficheiro de regras.

O *reasoner* é então criado com o ficheiro *configuration*. Após já estar definido o mecanismo de como será feita a inferência de conhecimento sobre o modelo *RDF* é criado um *InfModel infModel*, com o modelo pretendido e o *reasoner* a usar. Com o método *prepare()* é efectuado algum processamento e as operações de cache necessárias para a inferência.

A instrução `infModel.write(out,"RDF/XML")` permite serializar para o ficheiro *out* o modelo *RDF* com os novos dados inferidos, guardando a informação na codificação "RDF/XML".

```
public void runEngine(String ruleFile,String filename, Model rdfModel){
    Writer out = null;
    try{
        configuration = rdfModel.createResource();
        configuration.addProperty(ReasonerVocabulary.PROPruleMode, "hybrid");
        configuration.addProperty(ReasonerVocabulary.PROPruleSet, ruleFile);
        reasoner = GenericRuleReasonerFactory
            .theInstance().create(configuration);
        infModel = ModelFactory.createInfModel(reasoner, rdfModel);
        infModel.prepare();
    }catch(Exception ex) {
        JOptionPane.showMessageDialog(null, "Surgiu um problema na inferência. Por favor
verifique a sintaxe das regras.");
    }
    (...)
    out = new BufferedWriter(new FileWriter(new File(nomeFicheiro)));
    infModel.write(out, "RDF/XML");
    out.close();
    (...)
}
}
```

Fig 41: Inferência de conhecimento no modelo de dados

Para a consulta de dados sobre o modelo *RDF* seleccionado chegou-se à solução presente na figura 41. Como se pode verificar existem duas áreas de texto sendo que a que se encontra

mais à esquerda será a responsável pela descrição dos comandos na linguagem *SPARQL* a serem usados para a consulta, enquanto que a que se encontra mais à direita será a responsável pela apresentação dos resultados, após aplicação dos comandos *SPARQL* sobre o modelo.



Fig 42: Janela de consultas do motor de inferência da OntoLab

Após se ter os comandos *SPARQL* pretendidos para a consulta, e clicado no botão com a designação “Correr” (jButton1), é então criada uma instância de *query* com esses comandos.

A execução dessa *query* sobre o modelo pretendido é efectuada recorrendo ao método *create()* da classe *QueryExecutionFactory*, onde é passada a *query* e o modelo como parâmetros do método. É usado o método *out* da classe *ResultSetFormatter*, este método é responsável por indicar o local para onde se pretende direccionar o *output* gerado pela consulta e o formato que este vai assumir na sua representação.

A figura 42 permite ilustrar a parte fundamental da execução de consultas *SPARQL* sobre modelos ontológicos *RDF*.

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt){  
    try{  
        (...)  
        Query query = QueryFactory.create(jTextArea1.getText());  
        QueryExecution qe = QueryExecutionFactory.create(query,model);  
        ResultSet results = qe.execSelect();  
    }  
}
```

```
    ByteArrayOutputStream fileOut = new ByteArrayOutputStream();
    ResultSetFormatter.out(fileOut,results,query);
    JTextArea2.setText(fileOut.toString());
    (...)
}catch(Exception ex){
    JOptionPane.showMessageDialog(null, "Surgiram problemas na
apresentação de resultados. Por favor verifique a linguagem SPARQL.");
}
}
```

Fig 43: Janela principal do motor de inferência da OntoLab

6 Conclusões e Trabalho Futuro

A terminar esta dissertação apresenta-se um resumo do que foi abordado na mesma, destacam-se as principais contribuições para a investigação nas áreas tratadas assim como algumas limitações confrontadas durante o desenvolvimento. Finalmente realçam-se os resultados deste trabalho. Apresentam-se ainda algumas linhas futuras de investigação.

6.1 Conclusões

Ao longo do desenvolvimento deste trabalho procurou-se realizar e atingir os objectivos específicos definidos previamente. No que respeita à conclusão dos mesmos pode referir-se que estes foram atingidos de um modo geral.

No que respeita à investigação dos métodos e ferramentas de desenvolvimento ontológico, após análise das opções existentes, a escolha recaiu sobre o Protégé, por se tratar de uma ferramenta bastante visual, no entanto a principal dificuldade encontrada foi a necessidade de exportar a ontologia desenvolvida para um formato *RDF* de forma a ser utilizada no motor de

inferência. Esta lacuna foi ultrapassada, pois o Protégé, após o desenvolvimento de ontologia, permite guarda-la em formato *RDF*.

Relativamente à elaboração da lista de termos e definição da gramática da ontologia verifica-se que grande parte da limitação das regras e factos utilizados se deve ao facto da grande dimensão do código de trabalho português e da sua mudança “constante”, geralmente sempre que cada governo inicia uma legislatura o código de trabalho é alterado. No entanto, como forma de ultrapassar esta limitação seleccionou-se uma sub-parte do código do trabalho, referente apenas aos direitos e deveres do empregado e do empregador.

Quanto à inserção de valores na ontologia (leis, indivíduos e conflitos) convém referir que esta foi manual, uma vez que não existe nenhum formulário electrónico para o efeito.

Por sua vez, quanto ao futuro do *ODR* na resolução de conflitos pode-se concluir que o este será bastante importante, pois a resolução de conflitos formal, tal como a conhecemos actualmente, nos tribunais ou mesmo nos julgados de paz, nunca vai ser substituída completamente, mas muitos casos vão passar para sistemas *ODR* em que a rapidez de resolução será bastante maior e com menos custos para os utilizadores.

Como vantagens associadas à utilização do *ODR*, embora tenhamos muito a aprender sobre como usar as *TIC* para maximizar as vantagens e minimizar as possíveis desvantagens, podemos já enunciar: a velocidade de resolução de conflitos; a conveniência; a facilidade de acesso, pois não necessita de se deslocar; a eficiência; os reduzidos custos associados e a facilidade de armazenamento de dados digitais. Cada uma das partes pode ditar quando e onde vai responder a um procedimento *ODR*.

Outra vantagem é que a comunicação do especialista de *ODR* com as partes fica mais facilitada que quando feita na frente de ambas as partes, pois dessa maneira o conflito é muito mais difícil de resolver. A separação espacial e temporal também permite que a hostilidade e a raiva sejam reduzidas, focando as partes nas questões de fundo.

Não se pode referir somente as vantagens encontradas, pois existem algumas desvantagens, tais como: a possibilidade de perda de pistas não verbais (linguagem corporal, tacto e olfacto, por exemplo); as nuances de expressão; o momento e a comunicação de enquadramento e da persuasão, que muitas vezes fazem a diferença entre o sucesso e o fracasso na negociação e mediação *ODR*.

Os protótipos a que se chegou serão de grande utilidade, sendo o primeiro passo para o desenvolvimento de um sistema *ODR*. A ontologia criada trata-se somente de um protótipo, que

tem como principal objectivo avaliar a utilização deste tipo de tecnologia nos sistemas *ODR*, sendo que, do que se pode verificar com a elaboração deste trabalho, as ontologias trazem mais-valias aos sistemas *ODR*, uma vez que os tornam mais “inteligentes” e autónomos.

6.2 Trabalho futuro

O trabalho desenvolvido no âmbito desta tese pode ser melhorado e continuado de diversas formas. Em termos de concretização seria interessante pegar nos protótipos realizados e construir aplicações que fossem úteis para a justiça. O sistema poderia ser assim colocado num ambiente *ODR* e servir numa primeira instância para casos mais simples, indo evoluindo para casos mais complexos dentro do direito laboral, até chegar a um ponto de confiança a que ontologia poderá ser alargada a outras áreas do direito.

A avaliação de alguns aspectos do sistema teria de ser efectuada a longo prazo, pois a sua base de conhecimento só com o decorrer do tempo é que terá dados suficientes para ajudar na inferência de novo conhecimento.

No entanto, para que esta avaliação a longo prazo possa ser realizada, é fundamental que se crie, com ajuda de especialistas na área do processamento de linguagem natural, um método de inserção de dados de forma automática, sem o qual o sistema não pode ser convenientemente testado a longo prazo.

Para permitir o alargamento a mais áreas do direito seria necessário tornar a ontologia mais robusta e criar um *Web server* que sirva de *interface* para as partes. Como são tarefas que exigem algum trabalho seria vantajoso contar com algum apoio institucional, como já acontece em projectos deste tipo em outros países.

A junção do motor de inferência criado com a ontologia, tornando-os um só é também um objectivo futuro, tal como a criação de relatórios automáticos com dados do litígio que as partes podem solicitar no *Web server*.

Uma vertente desta abordagem, que não foi adequadamente especificada, tem a ver com a fase da composição de ontologias. Apesar de serem conhecidas várias metodologias para construção de ontologias e também algumas técnicas para integração e união de ontologias, o método apresentado requer algum refinamento nesta questão. Será necessário definir um método mais detalhado, que oriente o utilizador (desenvolvedor da ontologia) na construção da

ontologia global que irá apoiar a operação de um sistema de *ODR*, isto é, que ajude na identificação da técnica mais adequada para integração de duas ou mais ontologias, e que face à escolha feita, especifique detalhadamente os passos a seguir.

A escrita de artigos científicos que reflectam o estado actual da investigação é também uma tarefa que pode ser realizada. É uma forma de complementar este trabalho, servindo para divulgar o projecto e obter *feedback* que será positivo para o tornar mais robusto e o ajudará a crescer.

Bibliografia

- [1] Ethan Katsh. *Online Dispute Resolution: Resolving Conflicts in Cyberspace*. San Francisco, CA: Jossey-Bass Publishers, 2001.
- [2] Benyekhlef, Karim and Gelinas, Fabien, Online Dispute Resolution (July 1, 2005). *Lex Electronica*, Vol. 10, No. 2, 2005.
- [3] Gaitenby, A. 2004. Online Dispute Resolution. *The Internet Encyclopedia*.
- [4] Carneiro D., Novais P., Neves J., Towards Domain-Independent Conflict Resolution Tools, The 2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2011), 22 – 27 August 2011, Campus Scientifique de la Doua, Lyon, France
- [5] Carneiro D., Novais P., Neves J., Toward Seamless Environments for Dispute Prevention and Resolution, International Symposium on Ambient Intelligence, Advances in Intelligent and Soft Computing, Springer, 2011, Volume 92/2011, 25-32, DOI: 10.1007/978-3-642-19937-0_4.
- [6] P. Cortés *Civil Justice Quarterly* [94-95]. – (P. Cortés "Does the Proposed European Procedure Enhance the Resolution of Small Claims?" (2008)
- [7] <http://www.e-arbitration-t.com/2008/02/28/e-justice-centre-odr-in-second-life/>
(Outubro 2011)
- [8] <http://www.mids.ch/school/university.html>
(Outubro 2011)
- [9] <http://www.merc-mcmaster.ca>
(Outubro 2011)
- [10] <http://www.odr.info>
(Outubro 2011)
- [11] www.xml.com/2001/05/23/graphics/jenainterface.jpg
(Outubro 2011)
- [12] David Bilinsky, A Tale of Two (ODR) Projects (2011)
- [13] <http://www.singaporelaw.sg/content/Mediation.html>, Singapore Academy of Law
(Outubro 2011)
- [14] Abernethy, Steve. Building Large-Scale Online Dispute Resolution & Trustmark Systems

- [15] Walton, P.R.E. & McKersie, R.B. (1991). A behavioral theory of labor negotiations. Cornell University Press.
- [16] Brickey, D.; Guha R. V. (eds.) Resource Description Framework (RDF) - Schema specification 1.0 Março 2003.
- [17] Raiffa, H. (2002). The Art and Science of Negotiation. Harvard University Press.
- [18] Brown & Marriott, ADR principles and practice.
- [19] Langer, A. (2005). The Importance of Mediators, Bridge Builders, Wall Vaulters and Frontier. Una Città.
- [20] Feste, K. A. (1991). Plans for peace. Greenwood Publishing Group.
- [21] Bennett, S. C. (2002). Arbitration: essential concepts. ALM Publishing.
- [22] Lodder, A. R., & Zeleznikow, J. (2005). Developing an Online Dispute Resolution Environment: Dialogue Tools and Negotiation Support Systems in a Three-Step Model. Harvard Negotiation Law Review. Vol. 10, pp. 287-337.
- [23] Chiti, G. & Peruginelli G (2002). Artificial intelligence in alternative dispute resolution. In Proceedings of LEA 2002. Workshop on the Law of Electronic Agents, pages 97–104. CIRSFID, Bologna.
- [24] David Carneiro, Paulo Novais, Andrade (2009). Online Dispute Resolution – TIARAC.
- [25] Russell, S., Norvig, P. Artificial Intelligence: A Modern Approach
- [26] Kurzweil, R. (2000). The Age of Spiritual Machines: When Computers Exceed Human Intelligence. Penguin.
- [27] Weiss, G. (2000). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press.
- [28] Peruginelli, G. (2002). Artificial Intelligence in Alternative Dispute Resolution. In Sartor, G. (Eds.) Proceedings of the workshop on the Law of Electronic Agents (LEA02).
- [29] Waterman, D.A. & Peterson, M. (1980). Rule-based models of legal expertise. In the Proceedings of the First National Conference on Artificial Intelligence, Stanford University.
- [30] <http://www.w3.org>
(Outubro 2011)
- [31] Garshol, L. M. Metadata? Thesauri? Taxonomies? Topic Maps! Making sense of it all. Ontopia (<http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html>)
(Outubro 2011)

- [32] Duineveld, A.J.; Stoter, R.; Weiden; Kenepa; Benjamins, V.R.; WonderTools? A comparative study of ontological engineering tools.
- [33] Santi 2000
- [34] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. and Swartout, W. R. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):16--36, 1991
- [35] Noy, N. F.; McGuinness, D. L. *Ontology Development 101: A Guide to Creating Your First Ontology*.
- [36] Swartout e Tate, *Intelligent Sysatems and their Applications*, IEEE.
- [37] Fensel, D.; Hermelem, F. *On-To-Knowledge: Content-Driven Knowledge Management Tools through Evolving Ontologies*.
- [38] Sure, Y.; Studer, R. *A Methodology for Ontology-based Knowledge Management*. In: Davies, John; et al (eds). *Towards The Semantic Web: Ontology-Driven Knowledge Management*. John Wiley & Sons]
- [39] Gómez-Pérez, A; Corcho, O.; Fernández-López, M. *Ontologic Engineering: with examples from the areas of knowledge management, e-commerce and the semantic web*.
- [40] Denny, *Ontology building: A survey of editing tools*
- [41] Lassila, O., Swick, R. *Resource Description Framework (RDF) Model and Syntax Specification - W3C (World-Wide WEB Consortium) Recommendation 22 February 1999*.
- [42] Vanessa O'Connell, *Oil-and-Gas Unit Tests Online Resolution to Control Costs (2011)*
- [43] Ribeiro, J. & Almeida, P (2008). *Utilização de tecnologias da Web Semântica para inferir conhecimento de índole económica*
- [44] *Código do Trabalho Português, Artigos 127 a 129*
- [45] W3C – World Wide WEB Consortium. *eXtensible Markup Language (XML)*.
- [46] W3C RDF Validator Service

Anexo I - Artigos do Código do Trabalho

Artigo 127.º

Deveres do empregador

1 – O empregador deve, nomeadamente:

- a) Respeitar e tratar o trabalhador com urbanidade e probidade;
- b) Pagar pontualmente a retribuição, que deve ser justa e adequada ao trabalho;
- c) Proporcionar boas condições de trabalho, do ponto de vista físico e moral;
- d) Contribuir para a elevação da produtividade e empregabilidade do trabalhador, nomeadamente proporcionando-lhe formação profissional adequada a desenvolver a sua qualificação;
- e) Respeitar a autonomia técnica do trabalhador que exerça actividade cuja regulamentação ou deontologia profissional a exija;
- f) Possibilitar o exercício de cargos em estruturas representativas dos trabalhadores;
- g) Prevenir riscos e doenças profissionais, tendo em conta a protecção da segurança e saúde do trabalhador, devendo indemnizá-lo dos prejuízos resultantes de acidentes de trabalho;
- h) Adoptar, no que se refere a segurança e saúde no trabalho, as medidas que decorram de lei ou instrumento de regulamentação colectiva de trabalho;
- i) Fornecer ao trabalhador a informação e a formação adequadas à prevenção de riscos de acidente ou doença;
- j) Manter actualizado, em cada estabelecimento, o registo dos trabalhadores com indicação de nome, datas de nascimento e admissão, modalidade de contrato, categoria, promoções, retribuições, datas de início e termo das férias e faltas que impliquem perda da retribuição ou diminuição de dias de férias.

2 – Na organização da actividade, o empregador deve observar o princípio geral da adaptação do trabalho à pessoa, com vista nomeadamente a atenuar o trabalho monótono ou cadenciado em função do tipo de actividade, e as exigências em matéria de segurança e saúde, designadamente no que se refere a pausas durante o tempo de trabalho.

3 – O empregador deve proporcionar ao trabalhador condições de trabalho que favoreçam a conciliação da actividade profissional com a vida familiar e pessoal.

4 – O empregador deve comunicar ao serviço com competência inspectiva do ministério responsável pela área laboral, antes do início da actividade da empresa, a denominação, sector de actividade ou objecto social, endereço da sede e outros locais de trabalho, indicação da publicação oficial do respectivo pacto social, estatuto ou acto constitutivo, identificação e domicílio dos respectivos gerentes ou administradores, o número de trabalhadores ao serviço e a apólice de seguro de acidentes de trabalho.

5 – A alteração dos elementos referidos no número anterior deve ser comunicada no prazo de 30 dias.

6 – Constitui contra-ordenação leve a violação do disposto na alínea j) do n.º 1 ou nos n.os 4 ou 5.

Artigo 128.º

Deveres do trabalhador

1 – Sem prejuízo de outras obrigações, o trabalhador deve:

- a) Respeitar e tratar o empregador, os superiores hierárquicos, os companheiros de trabalho e as pessoas que se relacionem com a empresa, com urbanidade e probidade;
- b) Comparecer ao serviço com assiduidade e pontualidade;
- c) Realizar o trabalho com zelo e diligência;
- d) Participar de modo diligente em acções de formação profissional que lhe sejam proporcionadas pelo empregador
- e) Cumprir as ordens e instruções do empregador respeitantes a execução ou disciplina do trabalho, bem como a segurança e saúde no trabalho, que não sejam contrárias aos seus direitos ou garantias;
- f) Guardar lealdade ao empregador, nomeadamente não negociando por conta própria ou alheia em concorrência com ele, nem divulgando informações referentes à sua organização, métodos de produção ou negócios;

- g) Velar pela conservação e boa utilização de bens relacionados com o trabalho que lhe forem confiados pelo empregador;
- h) Promover ou executar os actos tendentes à melhoria da produtividade da empresa;
- i) Cooperar para a melhoria da segurança e saúde no trabalho, nomeadamente por intermédio dos representantes dos trabalhadores eleitos para esse fim;
- j) Cumprir as prescrições sobre segurança e saúde no trabalho que decorram de lei ou instrumento de regulamentação colectiva de trabalho.

2 – O dever de obediência respeita tanto a ordens ou instruções do empregador como de superior hierárquico do trabalhador, dentro dos poderes que por aquele lhe forem atribuídos.

Artigo 129.º

Garantias do trabalhador

1 – É proibido ao empregador:

- a) Opor -se, por qualquer forma, a que o trabalhador exerça os seus direitos, bem como despedi-lo, aplicar -lhe outra sanção, ou tratá-lo desfavoravelmente por causa desse exercício;
- b) Obstar injustificadamente à prestação efectiva de trabalho;
- c) Exercer pressão sobre o trabalhador para que actue no sentido de influir desfavoravelmente nas condições de trabalho dele ou dos companheiros;
- d) Diminuir a retribuição, salvo nos casos previstos neste Código ou em instrumento de regulamentação colectiva de trabalho;
- e) Mudar o trabalhador para categoria inferior, salvo nos casos previstos neste Código;
- f) Transferir o trabalhador para outro local de trabalho, salvo nos casos previstos neste Código ou em instrumento de regulamentação colectiva de trabalho, ou ainda quando haja acordo;
- g) Ceder trabalhador para utilização de terceiro, salvo nos casos previstos neste Código ou em instrumento de regulamentação colectiva de trabalho;

- h) Obrigar o trabalhador a adquirir bens ou serviços a ele próprio ou a pessoa por ele indicada;
- i) Explorar, com fim lucrativo, cantina, refeitório, economato ou outro estabelecimento directamente relacionado com o trabalho, para fornecimento de bens ou prestação de serviços aos seus trabalhadores;
- j) Fazer cessar o contrato e readmitir o trabalhador, mesmo com o seu acordo, com o propósito de o prejudicar em direito ou garantia decorrente da antiguidade.

2 – Constitui contra-ordenação muito grave a violação do disposto neste artigo.

Anexo II – Código RDF gerado pelo Protégè

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY www "http://www.ONTOLAB.org#" >
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
```

```
<rdf:RDF xmlns="http://www.ONTOLAB.org#"
  xml:base="http://www.ONTOLAB.org"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:www="http://www.ONTOLAB.org#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <owl:Ontology rdf:about=""/>
```

```
<!--
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//////////
```

```
//
```

```
// Object Properties
```

```
//
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//////////
```

```
-->
```

```
<!-- http://www.ONTOLAB.org##233;Arguido -->
<owl:ObjectProperty rdf:about="##233;Arguido">
  <rdfs:domain rdf:resource="#Litigio"/>
  <owl:inverseOf rdf:resource="##233;Especialista"/>
  <owl:propertyDisjointWith rdf:resource="##233;Especialista"/>
  <owl:propertyDisjointWith rdf:resource="##233;Reu"/>
  <owl:propertyDisjointWith rdf:resource="##233;Testemunha"/>
  <owl:inverseOf rdf:resource="##233;Testemunha"/>
</owl:ObjectProperty>
```

```
<!-- http://www.ONTOLAB.org##233;EmpregadoDe -->
<owl:ObjectProperty rdf:about="##233;EmpregadoDe">
  <rdfs:domain rdf:resource="#Pessoa"/>
  <owl:inverseOf rdf:resource="##233;EmpregadorDe"/>
  <owl:propertyDisjointWith rdf:resource="##233;EmpregadorDe"/>
</owl:ObjectProperty>
```

```
<!-- http://www.ONTOLAB.org##233;EmpregadorDe -->
<owl:ObjectProperty rdf:about="##233;EmpregadorDe">
  <rdfs:domain rdf:resource="#Pessoa"/>
</owl:ObjectProperty>
```

```
<!-- http://www.ONTOLAB.org##233;Especialista -->
<owl:ObjectProperty rdf:about="##233;Especialista">
  <rdfs:domain rdf:resource="#Litigio"/>
  <owl:inverseOf rdf:resource="##233;Reu"/>
  <owl:propertyDisjointWith rdf:resource="##233;Reu"/>
  <owl:inverseOf rdf:resource="##233;Testemunha"/>
  <owl:propertyDisjointWith rdf:resource="##233;Testemunha"/>
</owl:ObjectProperty>
```

```

<!-- http://www.ONTOLAB.org#&#233;Reu -->
<owl:ObjectProperty rdf:about="#&#233;Reu">
  <rdfs:domain rdf:resource="#Litigio"/>
  <owl:inverseOf rdf:resource="#&#233;Arguido"/>
  <owl:propertyDisjointWith rdf:resource="#&#233;Testemunha"/>
</owl:ObjectProperty>

```

```

<!-- http://www.ONTOLAB.org#&#233;Testemunha -->
<owl:ObjectProperty rdf:about="#&#233;Testemunha">
  <rdfs:domain rdf:resource="#Litigio"/>
  <owl:inverseOf rdf:resource="#&#233;Reu"/>
</owl:ObjectProperty>

```

```

<!--

```

```

////////////////////////////////////

```

```

//////////

```

```

//

```

```

// Data properties

```

```

//

```

```

////////////////////////////////////

```

```

//////////

```

```

-->

```

```

<!-- http://www.ONTOLAB.org#Alinea -->
<owl:DatatypeProperty rdf:about="#Alinea">
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://www.ONTOLAB.org#BI -->
<owl:DatatypeProperty rdf:about="#BI">
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://www.ONTOLAB.org#DescricaoArtigo -->
<owl:DatatypeProperty rdf:about="#DescricaoArtigo">
  <rdfs:domain rdf:resource="#Artigo"/>
  <rdfs:range rdf:resource="&rdf;XMLLiteral"/>
</owl:DatatypeProperty>

<!-- http://www.ONTOLAB.org#DescricaoLitigio -->
<owl:DatatypeProperty rdf:about="#DescricaoLitigio">
  <rdfs:domain rdf:resource="#Litigio"/>
  <rdfs:range rdf:resource="&rdf;XMLLiteral"/>
</owl:DatatypeProperty>

<!-- http://www.ONTOLAB.org#NIF -->
<owl:DatatypeProperty rdf:about="#NIF">
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://www.ONTOLAB.org#Nome -->
<owl:DatatypeProperty rdf:about="#Nome">
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://www.ONTOLAB.org#Nr -->
<owl:DatatypeProperty rdf:about="#Nr">
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```
<!-- http://www.ONTOLAB.org#TipoEmpregador -->  
<owl:DatatypeProperty rdf:about="#TipoEmpregador">  
  <rdfs:domain rdf:resource="#Empregador"/>  
  <rdfs:range rdf:resource="&rdf;XMLLiteral"/>  
</owl:DatatypeProperty>  
<rdf:Description>  
  <rdf:type rdf:resource="&owl;Axiom"/>  
  <rdfs:label>Singular</rdfs:label>  
  <rdfs:label>Colectivo</rdfs:label>  
  <owl:subject rdf:resource="#TipoEmpregador"/>  
  <owl:object rdf:resource="&rdf;XMLLiteral"/>  
  <owl:predicate rdf:resource="&rdfs;range"/>  
</rdf:Description>
```

```
<!--
```

```
////////////////////////////////////
```

```
//////////
```

```
//
```

```
// Classes
```

```
//
```

```
////////////////////////////////////
```

```
//////////
```

```
-->
```

```
<!-- http://www.ONTOLAB.org#Artigo -->
```

```
<owl:Class rdf:about="#Artigo">
```

```
  <rdfs:subClassOf rdf:resource="#Lei"/>
```

```

</owl:Class>

<!-- http://www.ONTOLAB.org#Empregado -->
<owl:Class rdf:about="#Empregado">
  <rdfs:subClassOf rdf:resource="#Pessoa"/>
</owl:Class>

<!-- http://www.ONTOLAB.org#Empregador -->
<owl:Class rdf:about="#Empregador">
  <rdfs:subClassOf rdf:resource="#Pessoa"/>
</owl:Class>

<!-- http://www.ONTOLAB.org#Especialista -->
<owl:Class rdf:about="#Especialista">
  <rdfs:subClassOf rdf:resource="#Visitante"/>
</owl:Class>

<!-- http://www.ONTOLAB.org#Lei -->
<owl:Class rdf:about="#Lei"/>

<!-- http://www.ONTOLAB.org#Litigio -->
<owl:Class rdf:about="#Litigio"/>

<!-- http://www.ONTOLAB.org#Pessoa -->
<owl:Class rdf:about="#Pessoa"/>

<!-- http://www.ONTOLAB.org#Testemunha -->
<owl:Class rdf:about="#Testemunha">
  <rdfs:subClassOf rdf:resource="#Visitante"/>
</owl:Class>

<!-- http://www.ONTOLAB.org#Visitante -->

```



```

<owl:Class rdf:about="#Visitante">
  <rdfs:subClassOf rdf:resource="#Pessoa"/>
</owl:Class>

```

```

<!--

```

```

////////////////////////////////////

```

```

//////////

```

```

//

```

```

// Individuals

```

```

//

```

```

////////////////////////////////////

```

```

//////////

```

```

-->

```

```

<!-- http://www.ONTOLAB.org#Ant#243;nio -->

```

```

<Empregador rdf:about="#Ant#243;nio">

```

```

  <rdf:type rdf:resource="#Pessoa"/>

```

```

  <rdf:type rdf:resource="&owl;Thing"/>

```

```

  <Nome>Ant#243;nio</Nome>

```

```

  <TipoEmpregador>Colectivo</TipoEmpregador>

```

```

  <NIF>12343234</NIF>

```

```

  <BI>12432123</BI>

```

```

  <éReu rdf:resource="#DividaSalario"/>

```

```

  <éEmpregadorDe rdf:resource="#Jorge"/>

```

```

</Empregador>

```

```

<!-- http://www.ONTOLAB.org#DividaSalario -->

```

```

<Litigio rdf:about="#DividaSalario">

```

```
<rdf:type rdf:resource="&owl;Thing" />
<DescricaoLitigio
  >O Arguido Jorge reclama sal&#225;rios em atraso na presta&#231;&#227;o de
servi&#231;os ao Reu Ant&#243;nio.
```

```
Ser&#225; ouvido como testemunha o Manuel.</DescricaoLitigio>
```

```
<éReu rdf:resource="#Ant&#243;nio" />
<éArguido rdf:resource="#Jorge" />
</Litigio>
```

```
<!-- http://www.ONTOLAB.org#Jorge -->
```

```
<Empregado rdf:about="#Jorge">
  <rdf:type rdf:resource="#Pessoa" />
  <rdf:type rdf:resource="&owl;Thing" />
  <NIF rdf:datatype="&xsd:string">11111111</NIF>
  <Nome rdf:datatype="&xsd:string">Jorge</Nome>
  <BI>321231111</BI>
  <éEmpregadoDe rdf:resource="#Ant&#243;nio" />
  <éArguido rdf:resource="#DividaSalario" />
</Empregado>
```

```
<!-- http://www.ONTOLAB.org#Manuel -->
```

```
<owl:Thing rdf:about="#Manuel">
  <rdf:type rdf:resource="#Pessoa" />
  <rdf:type rdf:resource="#Testemunha" />
  <rdf:type rdf:resource="#Visitante" />
  <NIF>3332134</NIF>
  <Nome>Manuel</Nome>
  <BI>435321</BI>
  <éEmpregadoDe rdf:resource="#Ant&#243;nio" />
  <éTestemunha rdf:resource="#DividaSalario" />
  <owl:sameAs rdf:resource="#Jorge" />
</owl:Thing>
```

```

<!--

////////////////////////////////////
//////////
//
// General axioms
//

////////////////////////////////////
//////////

-->

<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDifferent"/>
  <owl:distinctMembers rdf:parseType="Collection">
    <rdf:Description rdf:about="#Manuel"/>
    <rdf:Description rdf:about="#Ant&#243;nio"/>
  </owl:distinctMembers>
</rdf:Description>
<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDifferent"/>
  <owl:distinctMembers rdf:parseType="Collection">
    <rdf:Description rdf:about="#Ant&#243;nio"/>
    <rdf:Description rdf:about="#Jorge"/>
  </owl:distinctMembers>
</rdf:Description>
</rdf:RDF>

```