Instituto Superior de Engenharia do Porto

População e Enriquecimento de Ontologias através de Web Scraping

Elisa da Conceição Marques Barreira

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Tecnologias do Conhecimento e Decisão

Orientador: Doutor Paulo Maio

Co-orientador: Doutor Nuno Silva

Júri:

Presidente:

Doutora Maria de Fátima Coutinho Rodrigues, Professora Coordenadora, DEI/ISEP

Vogais:

Doutora Isabel Cecília Correia da Silva Praça Gomes Pereira, Professora Adjunta, DEI/ISEP

Doutor Paulo Alexandre Fangueiro Oliveira Maio, Professor Adjunto, DEI/ISEP

Doutor Nuno Alexandre Pinto da Silva, Professor Cordenador, DEI/ISEP

Porto, Outubro 2014

Resumo

O surgir da World Wide Web providenciou aos utilizadores uma série de oportunidades no que diz respeito ao acesso a dados e informação. Este acesso tornou-se um ato banal para qualquer utilizador da Web, tanto pelo utilizador comum como por outros mais experientes, tanto para obter informações básicas, como outras informações mais complexas. Todo este avanço tecnológico permitiu que os utilizadores tivessem acesso a uma vasta quantidade de informação, dispersa pelo globo, não tendo, na maior parte das vezes, a informação qualquer tipo de ligação entre si. A necessidade de se obter informação de interesse relativamente a determinado tema, mas tendo que recorrer a diversas fontes para obter toda a informação que pretende obter e comparar, torna-se um processo moroso para o utilizador.

Pretende-se que este processo de recolha de informação de páginas web seja o mais automatizado possível, dando ao utilizador a possibilidade de utilizar algoritmos e ferramentas de análise e processamento automáticas, reduzindo desta forma o tempo e esforço de realização de tarefas sobre páginas web. Este processo é denominado Web Scraping.

Neste trabalho é descrita uma arquitetura de sistema de web scraping automático e configurável baseado em tecnologias existentes, nomeadamente no contexto da web semântica.

Para tal o trabalho desenvolvido analisa os efeitos da aplicação do Web Scraping percorrendo os seguintes pontos:

- Identificação e análise de diversas ferramentas de web scraping;
- Identificação do processo desenvolvido pelo ser humano complementar às atuais ferramentas de web scraping;
- Design duma arquitetura complementar às ferramentas de web scraping que dê apoio ao processo de web scraping do utilizador;
- Desenvolvimento dum protótipo baseado em ferramentas e tecnologias existentes;
- Realização de experiências no domínio de aplicação de páginas de super-mercados portugueses;
- Analisar resultados obtidos a partir destas.

Palavras-chave: World Wide Web, Web Scraping, Internet

Abstract

The rise of the World Wide Web has provided users with a lot of opportunities with regard to access to data and information. This access has become a banal act for any user of the Web by both the common user as for more experienced users, both for basic information, and more complex information. All this technological development has enabled users to have access to a vast amount of information scattered over the globe and they do not have, in most cases, any information linked. The need to obtain information of interest for a given topic, but having to use various sources to get all the information you want to obtain and compare it becomes a time consuming process for the user.

It is intended that this process of gathering information from web pages is as automated as possible, giving the user the possibility of using algorithms and analysis tools and automatic processing, thereby reducing the time and effort embodiment tasks on web pages. This process is called Web Scraping.

It is described an architecture for automated web scraping and configurable system based on existing technologies, particularly in the context of semantic web. For this, the developed work analyzes the effects of applying Web Scraping covering the following points:

- Identification and analysis of various tools for web scraping;
- Identification of additional process developed by man to supplement current web scraping tools;
- Design of a complement to the architecture of web scraping tool that gives support to the scraping web user process;
- Development of a prototype based on existing tools and technologies;
- Conducting experiments in the portuguese super markets domain;
- Analyzing the experiments' results.

Keywords: World Wide Web, Web Scraping, Internet

Agradecimentos

Durante todo este processo, muitas pessoas me influenciaram direta e indiretamente, estando do meu lado e apoiando-me das mais diversas formas.

Começo por agradecer aos meus Orientadores Professor Doutor Paulo Maio e Professor Doutor Nuno silva, que sempre foram incansáveis e sempre me prestaram todo o auxílio necessário para a realização deste trabalho.

Aos meus Pais, sempre presentes, que me apoiaram em todas as minhas escolhas durante o meu percurso académico, apesar de todas as dificuldades da vida, e que pela sua força e dedicação me tornaram na pessoa que sou hoje. À minha Irmã Sara, que sempre esteve do meu lado e a todos os meus familiares que sempre me deram o seu apoio e palavras de alento.

Ao meu Namorado Paulo, por toda a dedicação, paciência, ajuda e apoio dados durante estes meses. Sem eles não teria chegado tão longe neste processo.

E por ultimo, um agradecimento muito grande a todos os meus amigos e colegas de curso,que me acompanharam direta e indiretamente durante este percurso e que sempre me deram palavras de alento, coragem e conforto.

A todos vocês, um muito obrigado!

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

DOM	*Document Object Model*
HTML	*HyperText Markup Language*
JSON	*JavaScript Object Notation*
KB	Knowledge base
MAFRA	*MApping FRAmework*
OWL	*Web Ontology Language*
RDF	*Resource Description Framework*
RDFa	*Resource Description Framework in Attributes*
XHTML	*eXtensible Hypertext Markup Language*
XML	*eXtensible Markup Language*
XSD	*XML Schema Definition*
XSLT	*eXtensible Stylesheet Language for Transformation*
XPATH	*XML Path Language*
W3C	*World Wide Web Consortium*
W4F	*World-Wide Web Factory*
WWW	*World Wide Web*

1 Introdução

1.1 Problema

A informação é um bem essencial a qualquer atividade, de tal forma que a crescente necessidade de a obter leva à procura de formas mais simples, rápidas e úteis de a recolher.

Com o crescente crescimento da World Wide Web, o acesso à informação tornou-se muito mais facilitado, dado que se torna possível aceder a informação de uma forma mais abrangente e sem os limites que a informação em papel, rádio, televisão e outros media apresenta, uma vez que esta se encontra acessível permanentemente e em mais tipos de dispositivos.

Mas apesar disso torna-se também cada vez mais difícil obter informação abrangente e fidedigna, uma vez que os volumes de informação aumentam cada vez mais, e uma simples pesquisa pode retornar uma quantidade enorme de dados que dificilmente é processada em tempo útil pelo ser humano.

Com a crescente quantidade de dados, tornam-se necessários mecanismos que tirem partido da acessibilidade que a web disponibiliza, e que consigam fazê-lo de forma automática de modo a evitar o processo moroso de obter a informação manualmente.

Surge então o conceito de Web Scraping, que consiste numa técnica de extração de informação a partir da web, realizada de forma automática, por software. A sua utilização permite obter dados de páginas web através de regras de extração definidas pelo utilizador. Apesar de esta técnica possibilitar a ausência de uso humano para proceder à extração de dados, permitindo assim uma rápida recolha de dados e de forma automática, desconhece-se a sua utilização em áreas e domínios específicos.

Além disso, essas ferramentas não são ainda utilizadas para tarefas que exijam a integração de informação de diversas páginas e/ou domínios de conhecimento, ficando dessa forma a cargo do utilizador esses processos morosos e tecnicamente exigentes.

De facto, a representação da informação de forma a ser entendida (leia-se processada

segundo o significado atribuído pelo autor) pela máquina e ser útil para o utilizador requer mais do que ler páginas individuais, passa também por juntar informação de várias páginas distintas e complementares para gerar o conhecimento pretendido para tomar decisões. Por exemplo, quando é necessária a reserva de viagens de avião, facilmente se encontram aplicações/portais/páginas web que já agregam a informação referente a várias companhias aéreas, mas o mesmo não se passa em todos e qualquer domínio de conhecimento. O domínio das compras on-line de super-mercado (e.g. mercearia) é um caso onde o Web Scraping ainda não é utilizado, sendo um domínio onde existem diversas páginas disponíveis com informação, e para os quais não existem aplicações de apoio à tomada de decisão (e.g. procura, análise e compra).

A aplicação de web scraping poderia ser aplicada neste domínio e em muitos outros, facilitando as tarefas de tomada de decisão por parte do utilizador, pois facilita a procura de informação em várias páginas/portais distintas.

1.2 Definição

Dada a importância da Web para a obtenção de informação relevante, tornou-se necessária a criação de mecanismos que permitam transformar os dados não estruturados (i.e. páginas web para consulta pelo ser humano) em dados estruturados, para ser possível integrar os dados em bases de conhecimento e estes serem processados por algorimtos e sistemas de apoio à decisão. Este processo pode ser realizado com diferentes objetivos e com vários propósitos, desde aceder a conteúdos offline [1], obter informação para ser publicada em páginas de terceiros e outras aplicações para comparação de dados [1].

Normalmente, a informação presente na web encontra-se representada de forma não estruturada. Quando a informação é representada em HTML, é com o objetivo de ser apresentada na web e ser visualizada/consultada e utilizada por Humanos, uma vez que é interpretada pelo browser e é apresentada de forma visual [2].

Em contrapartida, a informação representada desta forma não se encontra adaptada para ser percebida pela máquina, dado que esta é uma linguagem direcionada para definir a visualização e descrever o conteúdo de páginas web, não sendo portanto uma linguagem direcionada para a interpretação pela máquina [2].

Uma vez que a linguagem HTML tem foco no aspeto visual e trata-se de informação não estruturada, não garante a representação explícita da sua semântica. Torna-se necessário efetuar uma transformação dos dados não estruturados em dados estruturados e garantir que os dados estruturados sigam e apresentem semântica, de forma a serem interpretados pela máquina.

A extração de informação pode ser realizada de diversas formas, sendo a forma mais simples a recolha manual, método utilizado pelos humanos, podendo esta também ser realizada por ferramentas que realizam as ações de forma automática [3].

À extração de informação de páginas Web dá-se o nome de Web Scraping ou Web Data Extraction [3][1][4]. Web Scraping é portanto uma técnica de extração de informação a partir da Web, assistida por ferramentas dedicadas [3], que processa o HTML das páginas Web requisitadas, podendo exportar esta informação para outros formatos [3].

Os dados estruturados seguem uma estrutura definida por um modelo. Este modelo pode ser o de uma base de dados ou através de Ontologias. Uma Ontologia é um modelo de dados que captura a semântica dum domínio de conhecimento, pela representação de conceitos, suas relacões e restrições de interpretação [5][6].

1.3 Objetivos

O desconhecimento da aplicação da técnica de Web Scraping na pesquisa, obtenção e extração de informação a partir da web serviu de mote para a realização desta dissertação.

A estruturação de dados pode ser utilizada para enriquecer as Ontologias nos vários níveis da sua estrutura, sendo este um dos principais objetivos. Este trabalho focar-se-á no enriquecimento ao nível das asserções, realizado com os dados estruturados, obtidos recorrendo à técnica de Web Scraping [3] para a recolha de dados.

Os principais objetivos desta dissertação são portanto:

- Efectuar um estudo sobre Web Scraping e um levantamento de ferramentas que permitam a utilização desta técnica
- Implementar a arquitetura criada para este problema, recorrendo a ferramentas e técnias estudadas, estruturar os dados obtidos de acordo com a ontologia definida, de forma a popular uma base de conhecimento
- Realizar testes no domínio das compras de supermercado, utilizando um sub-conjunto das ferramentas estudadas
- Análisar os resultados obtidos com as experiências realizadas.

1.4 Estrutura

Este documento divide-se em cinco capítulos. O primeiro capítulo introduz a temática do trabalho, apresenta algumas definições de conceitos fundamentais, os objetivos do trabalho e a estrutura do documento.

No segundo capítulo é apresentado o Estado da Arte, incluindo toda a descrição teórica subjacente a este trabalho, bem como resultados da pesquisa sobre ferramentas de Web Scraping.

No terceiro capítulo é apresentada a proposta de sistema, apresentando a arquitetura e descrição de implementação.

Posteriormente, no capítulo quatro são apresentadas as experiências realizadas e seus resultados, o que implica a descrição do domínio de conhecimento das experiências e a sua captura em ontologia.

Por fim, no capítulo 5 são apresentadas as conclusões, limitações da proposta e trabalho futuro.

2 Estado da Arte

Este capítulo apresenta o estado da arte sobre Web Scraping.

Web Scraping é uma técnica que pretende simular a uso humano da World Wide Web (WWW ou simplesmente Web), embora esta seja efetuada de forma automática por software [7][3]. A extração de informação pode ser realizada de diversas formas, podendo ser realizada por ferramentas que realizam as ações de forma automática [3], mas pode também ser realizada de forma manual.

2.1 Representação da Informação

Uma vez que a técnica de Web Scraping tem como objectivo a extração de informação de páginas web, torna-se também importante analisar a representação desta informação.

A informação encontra-se representada de diversas formas: de forma não estruturada, semi-estruturada ou estruturada. A forma como esta se encontra disponível para consulta, e os intervenientes que necessitam de a consultar, definem a linguagem em que esta informação é representada. A representação da informação assume vários formatos podendo estes ser de acordo com vários objetivos:

- Definição de visualização, como é o caso do HTML – tratam-se de linguagens que definem a estrutura e disposição dos dados na página web, definem a forma como os dados são visualizados. No caso específico do HTML, as “tags” HTML definem a forma como os dados são apresentados, a ordem de apresentação, a formatação do texto [8].

- Definição da descrição, como é o caso do XML – tratam-se de linguagens que se utilizam para descrever dados estruturados, dando aos dados relações estruturais entre eles. No caso específico da Linguagem XML, para definir o formato são definidas

"tags", com ligações entre si, de forma a estruturar os dados na descrição pretendida [9].

- Permite a persistência de dados, tal como Bases de Dados – trata-se de armazenamento de dados de uma forma permanente e não volátil, de forma a que, mesmo abandonando o sistema, os dados permaneçam intactos e disponíveis no futuro [10][11].

- Definição de Semântica, como RDFa e Microformatos – trata-se de uma forma de tratar a informação e permitir a sua interpretação pela máquina. RDFa é um conjunto de extensões que permite inserir anotações, embebidas em XHTML e atribuir descrições a propriedades [12], de forma a inserir dados semânticos e a permitir a extração de triplos RDF [13]. Microformato é uma convenção [14] utilizada na Web, com o objetivo de utilizar as tags de documentos HTML e XHTML para inserir metadados [15].

- Definição de Consultas, como o XPath – trata-se de uma linguagem de consulta (Query Language) que permite construir expressões que recorrem e processam um documento XML de modo semelhante a uma expressão regular. Pode também processar valores como números, valores Booleanos e cadeias de caracteres. O XPath foi definido no World Wide Web Consortium (W3C) [16].

O HTML é usado para visualização pelo ser humano, tornando as páginas apelativas e propícias para serem processadas pelo ser humano. Por contraponto a sua semântica é de difícil extração por algoritmos (i.e. os algoritmos têm que inferir o seu significado/semântica).

O XML é o oposto, uma vez que se trata de uma linguagem standadizada e que pode ser processada por qualquer máquina, independentemente do sistema e algoritmos de extração.

O RDFa, microformatos, e microdata combina os dois, mas não é amplamente usado.

O HTML é largamente maioritário, uma vez que os browsers estão preparados para intrepertar esta linguagem [17][18], e como tal é necessário extrair esse conhecimento para bases de dados/de conhecimento para que possa ser processado pelos algoritmos e segundo uma semântica conhecida por eles.

A combinação de XML e HTML permite que a utilização de RDFa e microformatos não necessite de realizar esta extração para bases de dados, uma vez que a sua utilização permite dar semântica aos dados, conseguindo assim ser interpretada por algoritmos, sem necessitar de extração.

2.2 Técnicas Web Scraping

Apesar da técnica de Web Scraping ainda não ser amplamente utilizada, existem diversas ferramentas que implementam esta técnica e têm como objetivo a extração de informação. Implementadas nas mais diversas linguagens, seguem diferentes diretrizes no que diz respeito à forma como são implementadas e ao modo de funcionamento.

Existem duas linhas de pesquisa relativamente à implementação de algoritmos e ferramentas de Web Scraping: Técnicas Tree-based e Machine Learning Approaches [19].

2.2.1 Tree-based

As técnicas Tree-based [19] tiram partido da natureza semiestruturada das páginas web, onde esta estrutura é tratada como uma árvore. A estrutura da página HTML (tags HTML) é representada em diferentes níveis, formando uma estrutura hierárquica, sendo esta estrutura explorada de forma a obter a informação nela contida. XHTML é uma combinação de Tags HTML com regras XML, garantindo uma estrutura em árvore bem formada em que as marcações/tags têm significado semântico para a visualização [20] equivalente ao do HTML original. Assim, uma vez que as páginas (X)HTML são definidas através de tags que possuem uma hierarquia, esta pode ser definida através de uma árvore, sendo para esse efeito utilizado o Document Object Module (DOM). DOM é uma convenção criada pela W3C, que permite visualizar a estrutura lógica de um documento, bem como alterar a sua estrutura, conteúdo e apresentação [21]. De uma forma genérica, o DOM permite definir a forma como o documento é acedido e manipulado [21].

Esta abordagem foi formalizada por Zhai e Liu [22][23][19], tendo estes autores desenvolvido um sistema de extração com base nela. A abordagem baseia-se na ideia de que a informação em documentos da Web normalmente é extraída em regiões contíguas da página, chamadas de “data record regions”. Esta estratégia consiste em identificar e extrair dessas regiões. Os autores foram buscar inspiração para esta técnica aos algoritmos de correspondência de árvores. O algoritmo funciona em duas etapas:

1. Segmentação - a página Web é dividida em segmentos, sem extrair quaisquer dados, sendo esta fase fundamental para a segunda etapa;

2. Alinhamento Parcial da Árvore – nesta fase é aplicado um algoritmo de alinhamento de árvore parcial aos dados divididos na primeira fase. Recorrendo ao XPath, os dados são estruturados segundo uma árvore.

Na figura 1 é apresentado um exemplo deste processo [24].

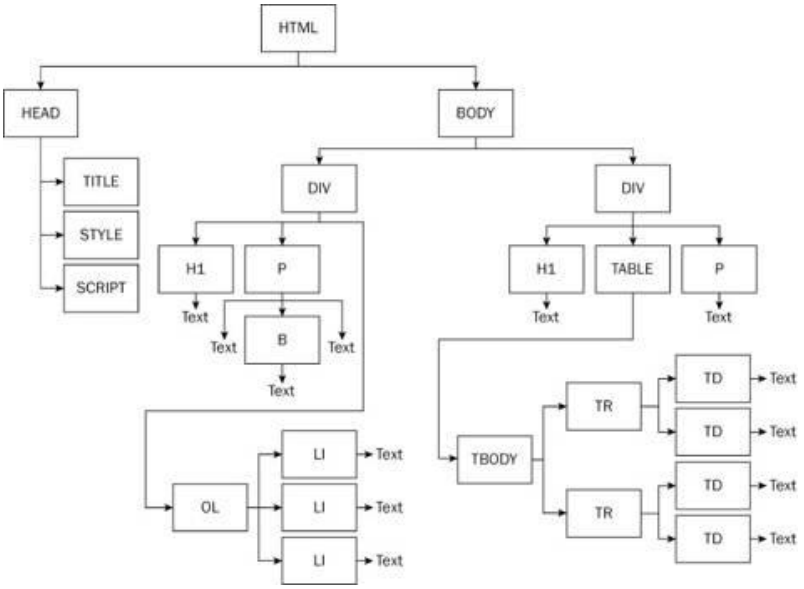

Figura 1 - Exemplo de criação de árvore DOM

Para ser possível tirar partido desta natureza, é utilizada a linguagem XPath para navegar pela árvore e explorar os nós.

Existem dois sub-tipos de diretrizes: Regular-expression-based aproach e Logic-based aproach.

2.2.1.1 Regular-expression-based aproach

Trata-se de uma das abordagens mais comuns, uma vez que expressões regulares são uma notação muito poderosa usada para identificar cadeias de caracteres ou padrões em texto não estruturado, com base em critérios de correspondência. A criação de regras pode ser bastante complexa e exige bastante experiência, sendo a grande vantagem das plataformas existentes o facto de elas possuírem uma interface que facilita ao utilizador a criação destas regras [19].

Um exemplo de aplicação desta técnica é a ferramenta W4F [25][26], que implementa expressões regulares para criar regras de extração.

No exemplo de código 1, é apresentado um exemplo do uso de expressões regulares na ferramenta W4F.

```
EXTRACTION_RULES ::
books = html.body.table[2].tr[0].td[1].ul[0].li[2].dl[0].dt[*]
 ( .b[0].a[0].pcdata[0].txt                              // title
#  .b[0].a[0].getAttr(href)                              // url
# ->dd[0].pcdata[0].txt, match /Published (19[0-9]{2})/   // year
# ->dd[0].pcdata[0].txt, match /(.*?)\//, split /, /     // authors
# ->dd[0].pcdata[1].txt, match /(\$[^ ]+)/               // price
  );
```

Código 1 - Exemplo de utilização de expressões regulares [25]

As expressões regulares geradas pela ferramenta W4F permitem extrair a informação pretendida pelo utilizador. Estas expressões são geradas automaticamente quando o utilizador seleciona a informação que pretende através da ferramenta.

2.2.1.2 Logic-based approach

Nesta abordagem, as páginas web não são entendidas como cadeias de texto, mas como árvores semiestruturadas, e o DOM representa a sua estrutura, onde os nós são elementos caracterizados pelas suas propriedades e seu conteúdo. A vantagem que esta abordagem apresenta é que as linguagens de programação dos Wrappers utilizadas para este efeito [19][27][28], como por exemplo a Linguagem *Lixto* [29] e a Linguagem *Elog*[30] podem ser definidas para explorar tanto a natureza semiestruturada das páginas como o seu conteúdo, necessitando de sistemas baseados em expressões regulares para levar a cabo o primeiro aspeto.

Na figura 2 é apresentado um exemplo da utilização desta técnica, recorrendo à Linguagem Elog [30].

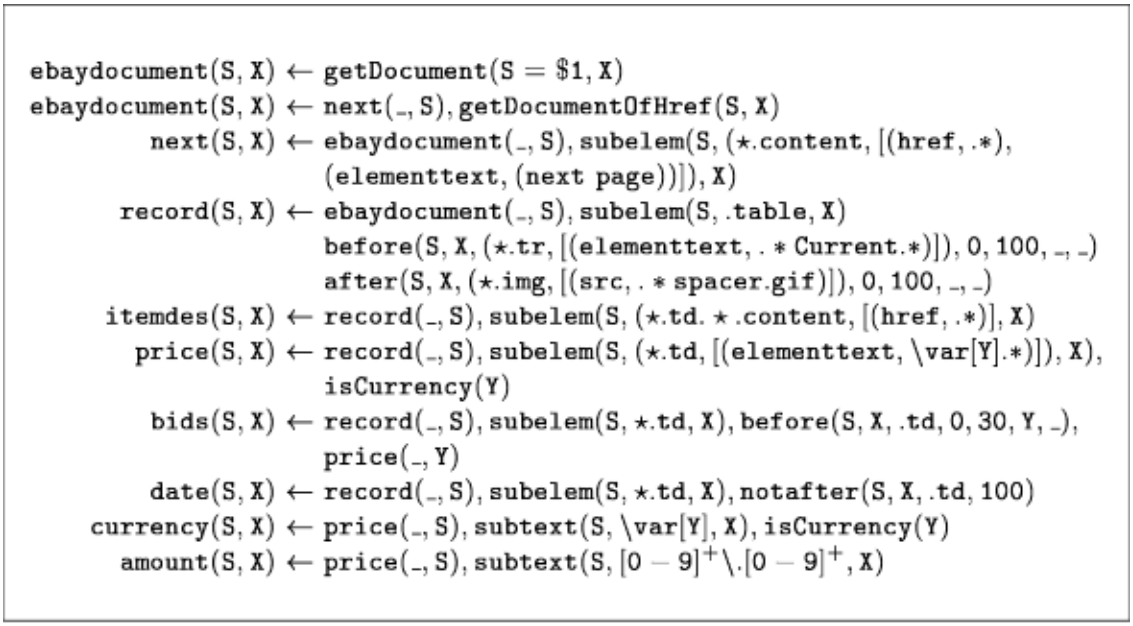

Figura 2 - Exemplo de utilização da Técnica Logic-Based Approach

Nesta fase, cada registo de dados é extraído da sua posição da subárvore DOM, constituindo a raiz de uma nova árvore única.

No exemplo apresentado no código 2 é apresentado o algoritmo referido inicialmente, e que foi formalizado pelos autores Zhai e Liu [23].

```
Sort trees in S in descending order according to the number of
data items that are not aligned;
Ts= the first tree (which is the largest) and delete it from S;
flag= false; R= ∅; I= false;
while(S≠∅)
```

```
Ti= select and delete next tree from S;
Simple_Tree_Matching(Ts, Ti);
L= alignTrees(Ts, Ti); // based on the result from line 6
if Ti is not completely aligned with Ts then
I = Insert IntoSeed(Ts, Ti);
if not all unaligned items in Ti are inserted into Ts then
Insert Ti into R;
endif;
endif;
if(L has new alignment) or (I is true) then
flag= true
endif;
if S= Ø and flag= true then
S= R; R= Ø;
flag= false; I= false
endif;
endwhile;
Output data fields from each Tito the data table based on the alignment
results.
```

Código 2 - Algoritmo Partial-Tree Alignment

2.2.2 Machine Learning

As técnicas de Machine Learning Approaches, de que são exemplo os Sistemas *WIEN* [31], WHISK [32] e *Rapier* [33][34], são técnicas de aprendizagem que exigem sessões de treino. Estas obtêm informação a partir das páginas web e ao mesmo tempo obtêm conhecimento sobre o domínio, servindo as sessões de treino para refinar a técnica de recolha e obtenção de conhecimento. Este tipo de técnicas requer, numa fase inicial, grande intervenção humana para catalogar as páginas web. Esta intervenção humana deve ser feita por peritos do domínio de forma a catalogar as páginas de forma correta para o treino. Estes sistemas foram desenvolvidos com base em duas abordagens:

- Conditional models [35] – Framework de inferência e Machine Learning que promove a aprendizagem recorrendo a modelos probabilísticos e discriminativos[35]. Esta Framework permite combinar conhecimento de domínio, fornecida por humanos, com modelos estatísticos [36];
- Adaptive Search [37] – Abordagem que aplica métodos de aprendizagem indutivos para a aquisição de conhecimento, gerando regras de extração a partir das páginas web de treino.

O sistema Rapier [33] baseia-se na aprendizagem de regras para extrair informação de páginas web. O sistema tem a capacidade de gerar estas regras automaticamente ao aceder à página que vai analisar, sem necessidade de uma análise prévia da página [19]. No código 3 é apresentado o algoritmo do sistema Rapier [33].

```
For each slot,S in the template being learned
```

```
SlotRules= most specific rules for S from example documents
Failures=0
While Failures<CompressLim
BestNewRule= FindNewRule(SlotRules,Examples)
If BestNewRule is acceptable
Add BestNewRule to SlotRules
remove empirically subsumed rules from SlotRules
else
Add 1 to Failures
```

Código 3 - Algoritmo Rapier

O sistema WHISK [32] baseia-se num algoritmo de aprendizagem supervisionada para gerar regras, com o objectivo de extrair informação de documentos de texto. Este sistema pode lidar com vários documentos de texto, desde documentos mais extruturados, como é o caso de páginas web que são definidas em HTML, a simples documentos de texto. As regras geradas pelo WHISK são compostas por dois componentes: o primeiro define em que contexto o texto pode ser considerado relevante e o segundo delimita o texto a ser extraído [19].

No código 4 é apresentado parte do algoritmo do WHISK [32].

```
WHISK(Reservoir)
  RuleSet=NULL
  Training=NULL
  Repeat at user's request
        Select a batch of NewInst from Reservoir
        (User tags the NewInst)
        Add NewInst to Training
        Discard rules with  errors on NewInst
        For each Inst in Training
              For each Tag of Inst
                    If Tag is not covered by RuleSet
                          Rule=GROW_RULE(Inst, Tag, Training)
Prune RuleSet
```

Código 4 - Algoritmo WHISK

O sistema WIEN [31] foi baseado em técnicas de aprendizagem indutiva, sendo o sistema capaz de atribuir rótulos às páginas de treino, não necessitando de tanta intervenção humana no processo de treino como outros sistemas. Uma limitação apresentada por este sistema é a dificuldade em lidar com valores em falta [19].

O sistema baseia-se em diversos algoritmos para proceder à aprendizagem. Na figura 3 é apresentado um dos algoritmos de aprendizagem do WIEN [31].

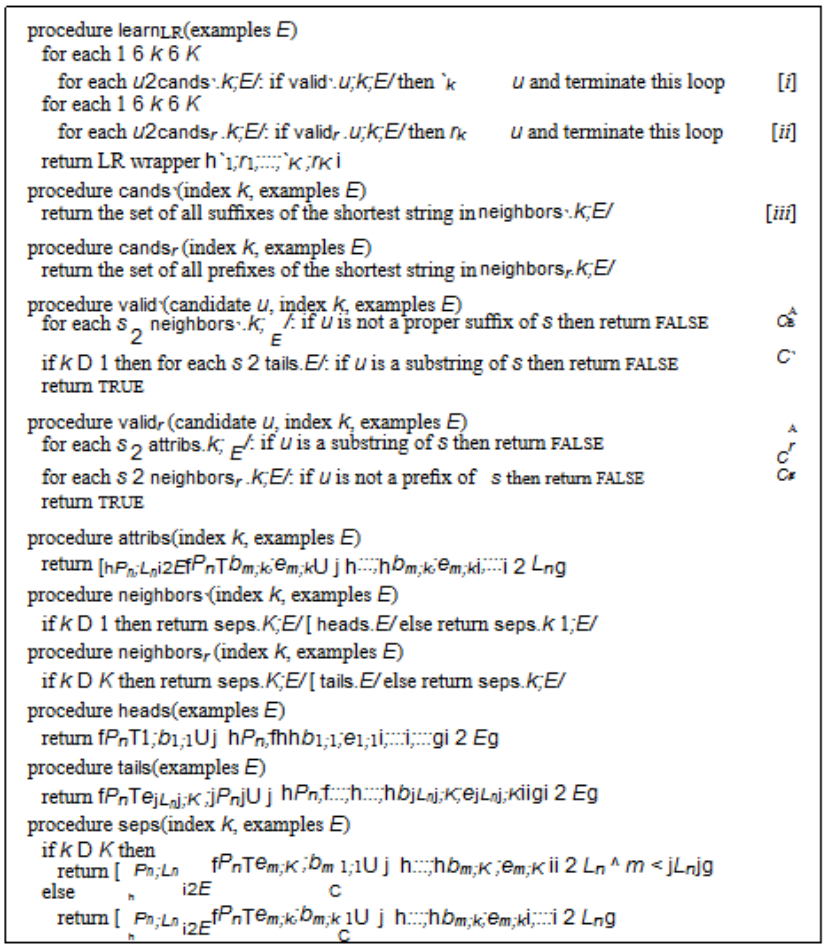

Figura 3 - Algoritmo de aprendizagem WIEN

Enquanto que os sistemas WHISK e Rapier se baseiam em regras para extrair informação de páginas Web, o sistema WIEN recorre a técnicas de aprendizagem indutiva para atribuir rótulos às páginas, seguindo uma abordagem diferente dos dois sistemas anteriores.

2.3 Ferramentas

Existem já diversas ferramentas disponíveis que implementam a técnica de Web Scraping.

2.3.1 WebSPHINX

WebSPHINX (Website-Specific Processors for HTML INformation eXtraction)[38] é uma biblioteca em Java e um ambiente de desenvolvimento de Web Crawlers. Web Crawler é um programa que pesquisa informação na Web automaticamente. WebSPHINX é composto por duas partes: *Crawler Workbench* e *WebSPHINX class library.*

A Crawler Workbench é uma interface gráfica que permite configurar e controlar um web

crawler personalizável. Usando o Crawler Workbench é possível:

1. Visualizar uma coleção de páginas web como um gráfico
2. Gravar páginas para o disco local para navegação offline
3. Concatenar páginas para visualização ou imprimi-las como um único documento
4. Extrair todo o texto correspondente a um certo padrão de um conjunto de páginas
5. Desenvolver rastreadores personalizados em Java ou Javascript que processa páginas web.

A *WebSPHINX class library* fornece suporte para a escrita de *crawlers* em Java. A biblioteca de classes oferece uma série de características:

1. Recuperação de página Web multithread
2. Um modelo de objetos que representam explicitamente páginas e links
3. Suporte para classificadores de conteúdo das páginas
4. Análise de HTML Tolerante
5. Suporte para o "robot exclusion standard"
6. A correspondência de padrões, incluindo expressões regulares, Unix shell wildcards, e expressões de tags HTML. As expressões regulares são fornecidas pela biblioteca de expressões regulares *Jakarta-regexp* da Apache[39]
7. Transformações HTML comuns, como a concatenação de páginas, gravar páginas para o disco, e renomeação de ligações

Esta ferramenta enquadra-se nas técnicas *Tree-based.*

Na figura 4 é apresentada a interface da aplicação WebSPHINX. Esta interface apresenta as principais opções que esta ferramenta oferece. A opção "File" permite a criação de um novo *crawler*, guardar o *crawler* criado e abrir Crawlers criados. Na opção "Crawl" é possível selecionar onde o *crawler* vai efetuar a busca. No campo "Starting URLs" deve ser inserido o URL que será ponto de partida da busca a realizar. No campo "Action" deverá ser escolhida a ação a realizar sobre os dados obtidos, podendo estes dados ser guardados num ficheiro, concatenados com outros dados e ser extraídos. Para executar o *crawler* deve ser premido o botão "Start". Esta execução pode ser interrompida e ser retomada mais tarde. Os resultados obtidos são apresentados no espaço inferior da interface, podendo ser apresentados em formato de gráfico se for selecionada a opção "Graph".

Na figura 4 é apresentado o gráfico gerado para a página http://blog.scrapinghub.com.

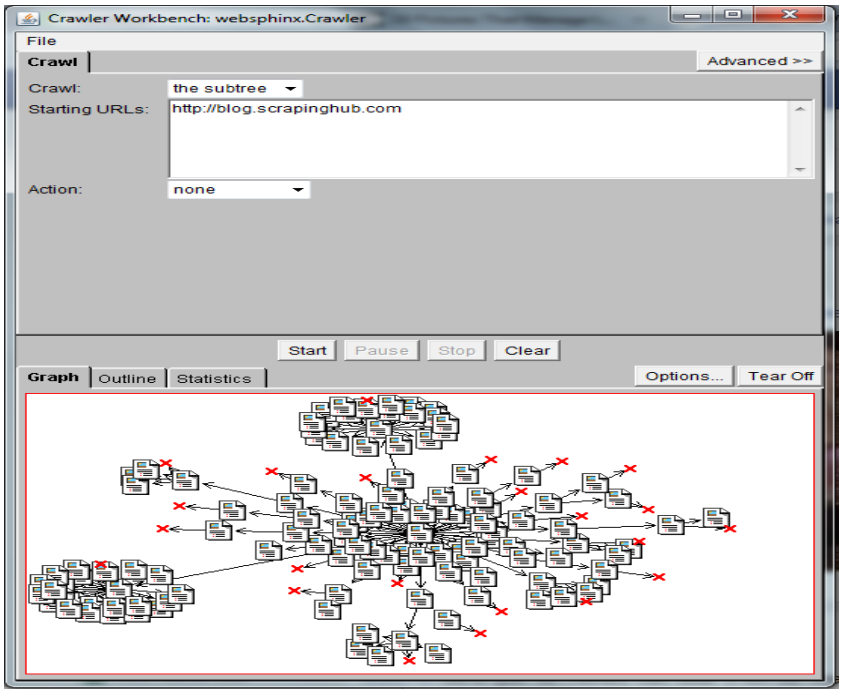

Figura 4 - Gráfico de resultados Interface WebSphinx

Ao ser selecionada a opção “Outline”, é apresentada uma listagem de links pesquisados. Na figura 5 é apresentada a listagem de lins extraídos da página http://blog.scrapinghub.com.

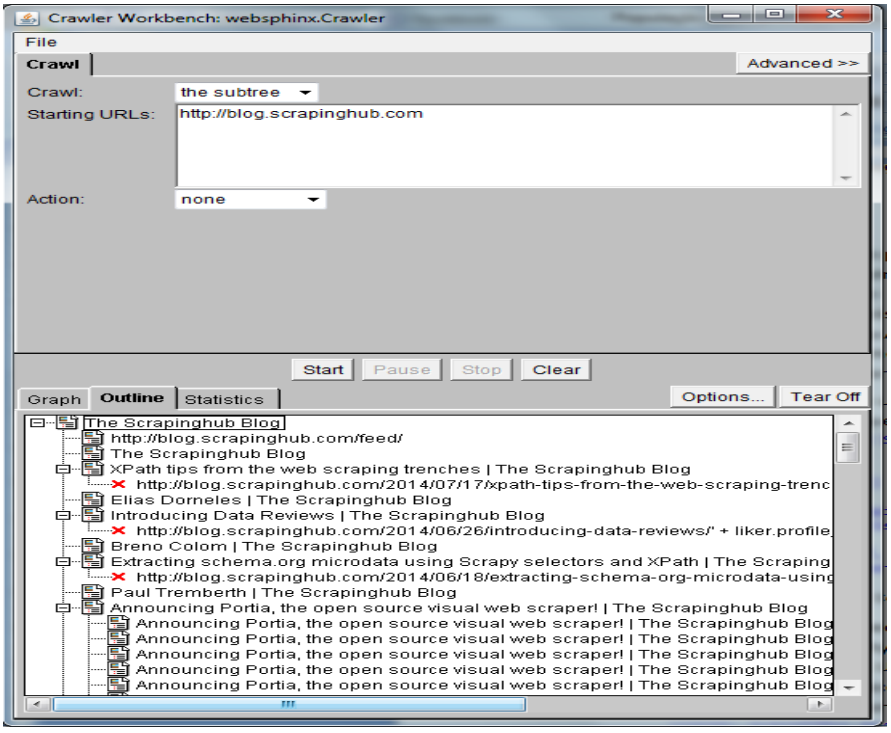

Figura 5 - Listagem de Links

Na seleção da opção “Statistics” são apresentadas as estatísticas da pesquisa, tais como o número total de páginas visitadas, o número de páginas visitadas por segundo, a memória utilizada e o tempo de execução.

Na figura 6 são apresentadas as estatísticas da extração realizada à página http://blog.scrapinghub.com.

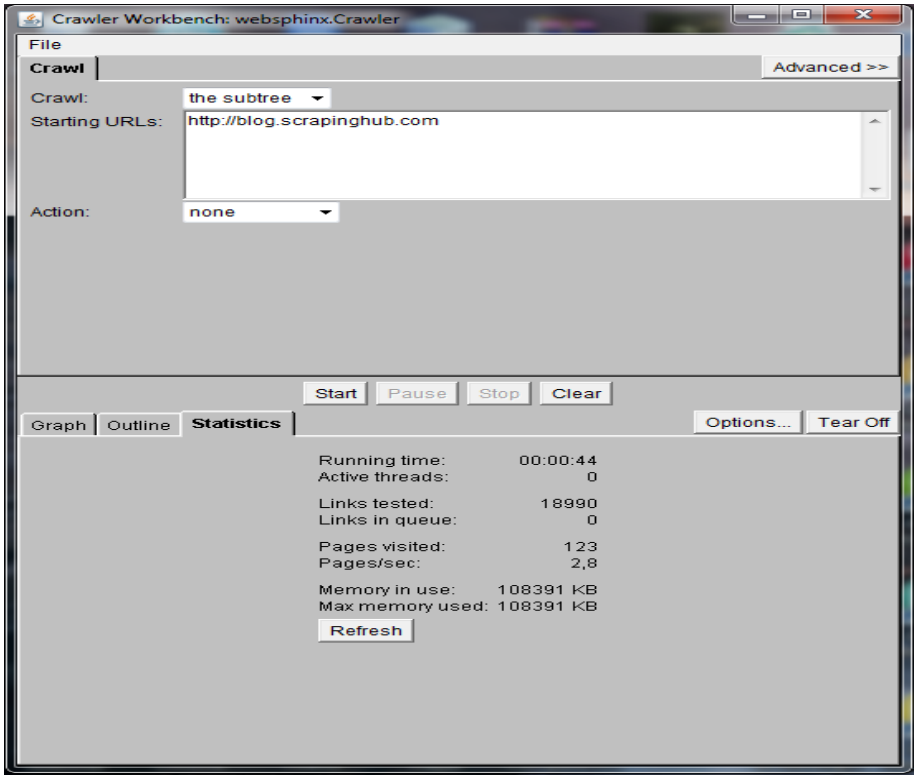

Figura 6 - Estatísticas

2.3.2 Scrapy

Scrapy [40] é uma aplicação para pesquisar em páginas web e extrair dados estruturados que podem ser utilizadas noutras aplicações úteis, como Data Minning, processamento de informação ou de arquivo histórico. Apesar de Scrapy ser originalmente projetado para captura de ecrã (mais precisamente, Web Srapping), também pode ser usado para extrair dados usando APIs (como a Amazon Associates Web Services). É controlada através de linha de comandos [40]. A ferramenta recorre à linguagem Pyhton para criar Spiders. Spiders são classes criadas pelo utilizador para realizar extrações de informação.

De seguida é apresentado um exemplo de criação de um Spider recorrendo à ferramenta Scrapy [41], que realiza uma extração à página 'http://blog.scrapinghub.com'.

```
from scrapy import Spider, Item, Field
class Post(Item):
    title = Field()
class BlogSpider(Spider):
```

```
name, start_urls = 'blogspider', ['http://blog.scrapinghub.com']
def parse(self, response):
    return [Post(title=e.extract()) for e in response.css("h2 a::text")]
```

Código 5 - Exemplo de utilização do Scrapy

Para criar um Spider, é necessário criar um ficheiro com a definição da classe. Como a ferramenta baseia-se em Phyton, a codificação dos Spiders é realizada nesta linguagem, tendo o ficheiro a extensão '.py'. Uma vez que a interface desta ferramenta é linha de comandos, para executar o Spider é necessário inserir o comando referente à sua execução. No exemplo de código 6 é apresentado o comando para executar um spider.

```
scrapy runspider myspider.py
```

Código 6 - Comando de Execução de um Spider

Esta ferramenta enquadra-se nas técnicas *Tree-based.*

2.3.3 IRobotSoft

IRobotSoft [42] é um software para realizar web scraping. Com esta ferramenta é possível criar robots inteligentes para realizar várias ações em páginas web. Com IRobotSoft pode ser simulado o clicar em links, o envio de formulários, a criação de horários, obter dados de páginas e tabelas, e até mesmo ligar a bases de dados. Podem ser criados robôs para adicionar amigos para as Redes sociais como o twitter e o facebook.

Esta ferramenta enquadra-se nas técnicas *Tree-based.*

Na imagem 7 é apresentada a interface da aplicação iRobotSoft.

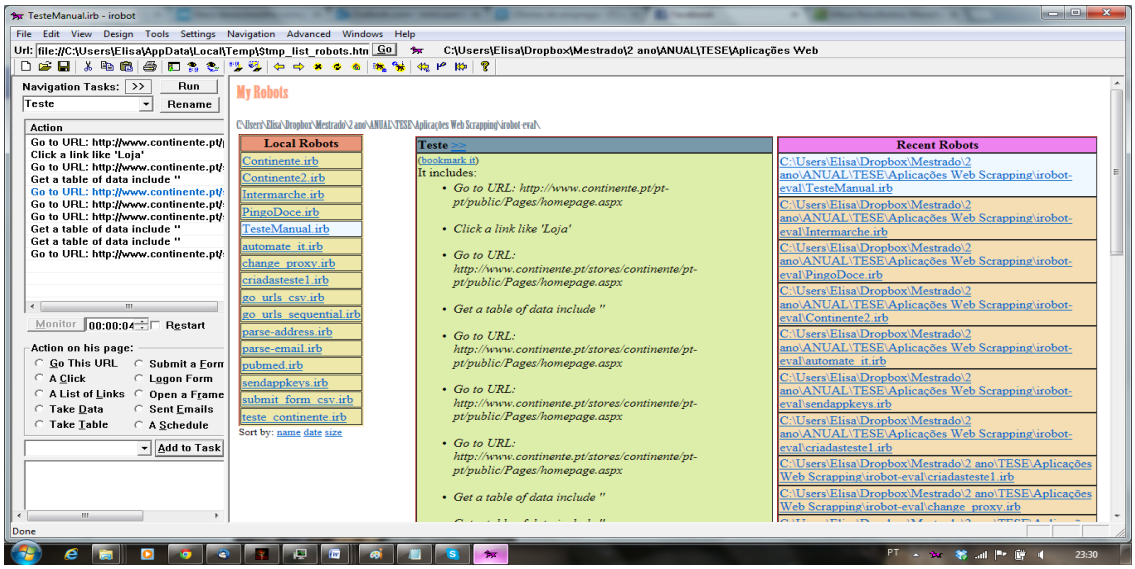

Figura 7 - Interface iRobotSoft

Do lado esquerdo da interface, no início é apresentada a listagem de ações definidas para o

robot que se encontre selecionado, bem como as opções de executar o robot e mudar o nome que foi atribuído a este mesmo robot, representada esta função na figura 8.

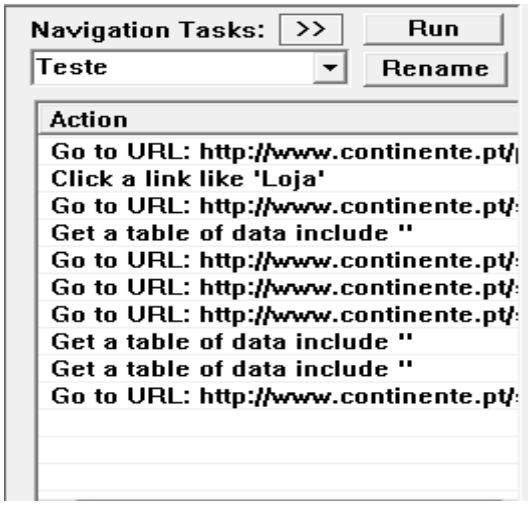

Figura 8 - Lista de ações do robot

Ao ser selecionada a opção de executar o robot, é apresentada ao utilizador a sua execução.

Na segunda parte é possível selecionar as ações a serem adicionadas na listagem de ações do robot, representadas na figura 9.

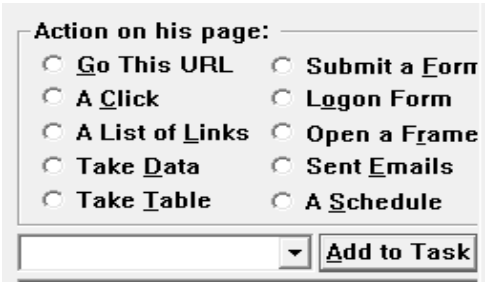

Figura 9 - Ações disponíveis no iRobot

A escolha de uma ação e a sua inserção na lista de ações do robot necessita de ser configurada. Como exemplo, no caso da opção 'Go This URL', esta ação necessita da inserção de um URL para qual será definida a navegação. Depois de configurada a ação, para adicionar a ação, deve ser selecionada a opção Add to Task, sendo esta inserida na listagem de açoes do robot.

2.3.4 Scraper

Scraper [43] é uma extensão do Google Chrome e trata-se de uma ferramenta de web scraping, para a captura de dados de páginas web e colocar a informação obtida em folhas de cálculo do Google. Utiliza expressões definidas em XPath para definir as regras de pesquisa e extração de dados.

Esta ferramenta enquadra-se nas técnicas *Tree-based.*

Na figura 10 é apresentada a interface da extensão Scraper.

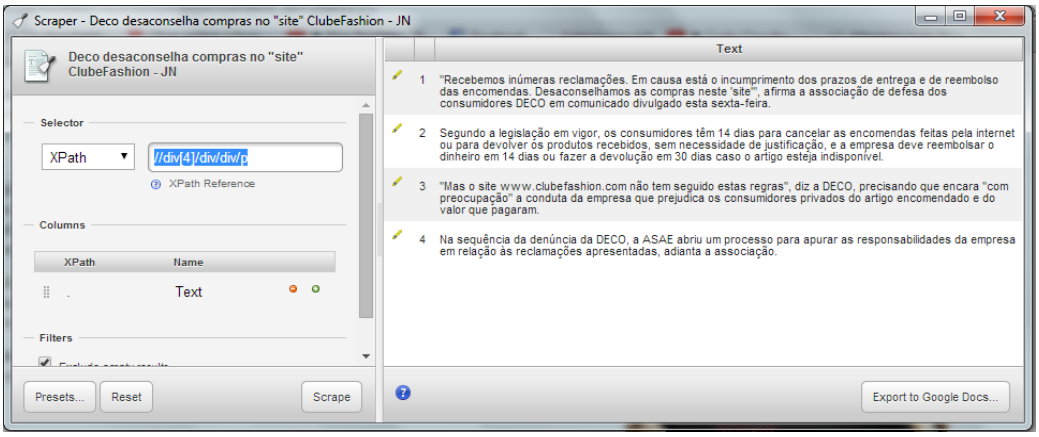

Figura 10 - Interface Scraper

Esta ferramenta permite a definição das regras de extração, e apresentar os resultados da extração levada a cabo por essas regras. Na figura 11 é apresentado o espaço de definição das regras.

Figura 11 - Definição de regras de extração

Os dados resultantes do processo de extração são apresentados na interface da ferramenta. Na figura 12 são apresentados os dados extraídos.

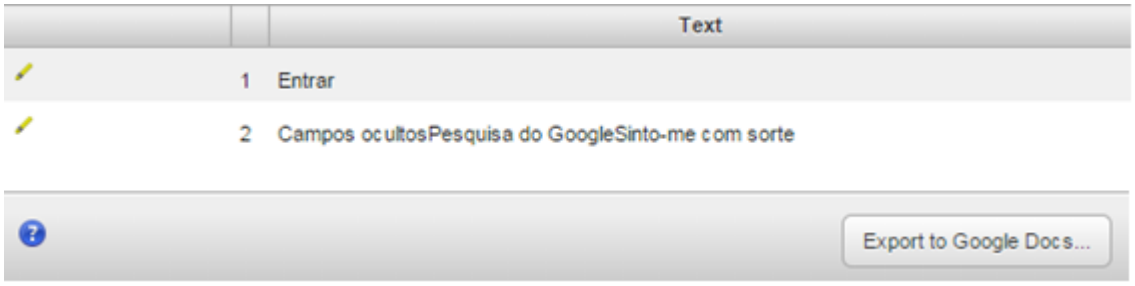

Figura 12 - Dados extraídos

A ferramenta apresenta a possibilidade de exportar os dados apresentados para um ficheiro do Google Docs.

2.3.5 Mozenda

Mozenda [44] é uma aplicação de extração de dados a partir da web. Esta permite aos utilizadores configurarem agentes que serão responsáveis por extrair a informação da web, bem como armazená-la. A aplicação permite guardar a informação como CSV, TSV e XML e permite a visualização dos dados em Excel, entre outros programas. Além das funcionalidades descritas, a aplicação permite também o envio da informação por e-mail e ftp. A aplicação possui duas formas de interação:

- Mozenda Agent Builder permite a criação de agentes para a realização de tarefas definidas pelo utilizador, permitindo a definição de todas estas tarefas. Permite também dar ordem de execução destes.
- Mozenda Web Console, acessível através do browser permite fazer a gestão dos Agentes existentes, eliminá-los, executá-los e consultar os seus resultados.

Na figura 13 é apresentada a interface do Mozenda Agent Builder.

Figura 13 - Interface do Mozenda Agent Builder

A interface do Mozenda Agent Builder permite criar agentes e definir as ações a executar, tal como clicar em items, capturar texto ou imagens e criar listas. Permite também modificar o comportamento destas ações. A interface permite criar estas ações manipulando a página web alvo de extração.

Na figura 14 é apresentada a página inicial do Mozenda Web Console quando se seleciona a opção “Agents” no menu do lado esquerdo, sendo apresentada a listagem dos Agentes criados e agendar a sua execução.

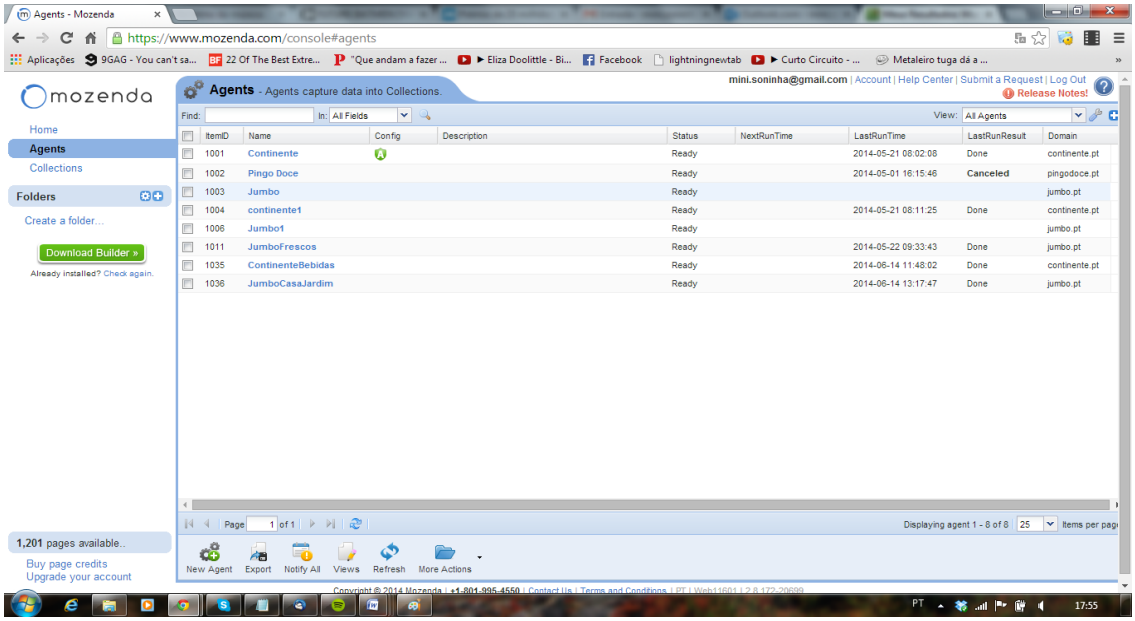

Figura 14 - Página de listagem dos Agentes

A Figura 15 apresenta um exemplo da interface para visualizar os resultados obtidos.

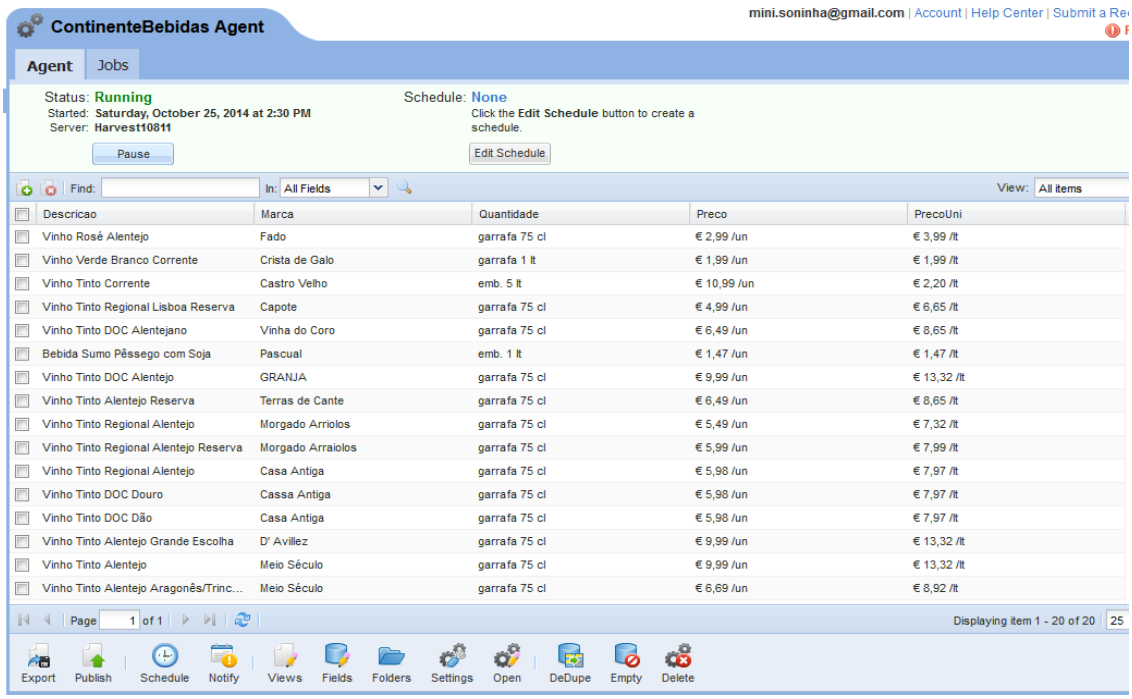

Figura 15 - Resulatados obtido por agente

2.3.6 Sistematização das Ferramentas

Para melhor perceber as características das várias ferramentas encontradas e estudadas, foi feita uma sistematização de algumas das suas características. A tabela 1 apresenta a

sistematização realizada.

Tabela 1 - Sistematização de Ferramentas

Ferramenta	Exportação Dados	UI	Envio de Formulários	Linha de Comandos	Agentes	Expressões Regulares	Agendamento	Técnica utilizada
WebSPHINX	TXT	Sim	Não	Não	Não	Sim	Não	Tree-based
Scrapy	JSON, JSON lines, CSV, XML	Não	Não	Sim	Não	Sim	Não	Tree-based
iRobotSoft	SGBD, CSV	Sim	Sim	Não	Sim	Não	Sim	Tree-based
Scraper	Google Spreadsheet	Sim	Não	Não	Não	Sim	Não	Tree-based
Mozenda	CSV, TSV, XML, Excel	Sim	Não	Não	Sim	Não	Sim	Tree-based

2.4 Ontologias

Ontologia é um ramo da filosofia que lida com a natureza do ser [45]. Por outro lado, nas ciências da Computação, Ontologia é um artefacto para capturar conhecimento relativo a um determinado domínio de conhecimento, sendo desenvolvidas como instrumento de representação de conhecimento. Trata-se portanto dum artefacto constituído por vocabulário que descreve o domínio, define os seus conceitos e relações entre conceitos. Define também restrições que capturam conhecimento adicional sobre o domínio, sendo uma extensão ao vocabulário [46][47].

A Ontologia é composta por duas partes distintas: TBox e ABox. TBox inclui a conceptualização da Ontologia, isto é, todos os conceitos definidos e as suas respetivas propriedades. ABox inclui todas as instâncias relacionadas com os conceitos da TBox. O conjunto constituído pela TBox e pela respetiva ABox também se denomina base de conhecimento (knowledge base, KB).

As ontologias são de interesse pela potencialidade que têm em organizar e representar informação, pois possuem a capacidade de abranger o conteúdo dum domínio e consensualizar a sua interpretação. O conhecimento representado na Ontologia é aceite pela comunidade pela qual será utilizada, pois trata-se de um conjunto de conceitos partilhados pelos elementos dessa mesma comunidade. As Ontologias são portanto uma conceptualização e representação de conhecimento e vocabulário partilhado pelos vários elementos, tendo como base um tema de conhecimento [48].

A capacidade de organização e sistematização de conceitos permite organizar o conteúdo de várias fontes de dados num determinado domínio, e pode melhorar e automatizar os processos de tratamento de informação.

A apropriação do termo “ontologia” da filosofia deve-se ao facto das ontologias servirem como meio de organização de coisas passíveis de representação simbólica (representação formal), e, a partir da representação formal, possibilitar raciocínio dedutivo através de regras

de inferência [48].

As ontologias podem ser classificadas quanto ao seu conteúdo: de domínio, de tarefas, de aplicação, genéricas e de representação [48].

1. Ontologias de Domínio - Descrevem conceitos e vocabulários relacionados a domínios particulares, tais como medicina ou computação, por exemplo. Este é o tipo de ontologia mais comum, geralmente construída para representar um "micro-mundo".

2. Ontologias de Tarefas - Descrevem tarefas ou atividades genéricas, que podem contribuir na resolução de problemas, independentes do domínio, por exemplo, processos de vendas ou diagnóstico. A sua principal motivação é facilitar a integração dos conhecimentos de tarefa e domínio numa abordagem mais uniforme e consistente, tendo por base o uso de ontologias.

3. Ontologias de Aplicação - Descrevem conceitos que dependem tanto de um domínio particular quanto de uma tarefa específica. Devem ser especializações dos termos das ontologias de domínio e de tarefa correspondentes. Estes conceitos normalmente correspondem a regras aplicadas a entidades de domínio enquanto executam determinada tarefa.

4. Ontologias de Representação - Explicam os conceitos que fundamentam os formalismos de representação de conhecimento, procurando tornar claros os compromissos ontológicos embutidos nestes formalismos. Um exemplo desta categoria é a ontologia de *frames*, utilizada em Ontolíngua.

5. Ontologias Genéricas - São consideradas ontologias "gerais". Descrevem conceitos mais amplos, como elementos da natureza, espaço, tempo, coisas, estados, eventos, processos ou ações, independente de um problema específico ou de um domínio em particular. Pesquisas com foco em ontologias genéricas procuram construir teorias básicas do mundo, de caráter bastante abstrato, aplicáveis a qualquer domínio (conhecimento de senso comum). [49]

Os componentes básicos de uma ontologia são os seguintes:

- Classes – organização e definição dos conceitos de um domínio, organizadas numa taxonomia. As classes são organizadas de forma semelhante ao modelo Orientado a Objetos e as classes-filho herdam as características das classes-pai, de forma similar ao modelo orientado a objetos;

- Propriedades - representam as interações entre os conceitos de um domínio. Existem dois tipos de propriedades:
    - Object Properties – trata-se de uma propriedade que representa a interação entre Individuals (Classes);
    - Data Properties - trata-se de uma propriedade que representa a interação entre um Individual (classe) e um Literal (Instância);
- Axiomas - usados para restringir a interpretação e o uso dos termos envolvidos;
- Instâncias - utilizadas para representar objetos específicos (os próprios dados) [48].

2.4.1 Ferramentas

2.4.1.1 Protége

É um ambiente interativo para construção de ontologias, com uma interface gráfica para manipulação de ontologias [50]. Permite a inserção de novos recursos. Foi desenvolvido pela Universidade de Stanford (http://www.stanford.edu/). A aplicação é composta por um editor de ontologia e uma biblioteca de plugins. Permite importar/exportar para diversas linguagens, entre elas OIL e XML [49][47]. Na figura 16 é apresentada a interface do Protégé.

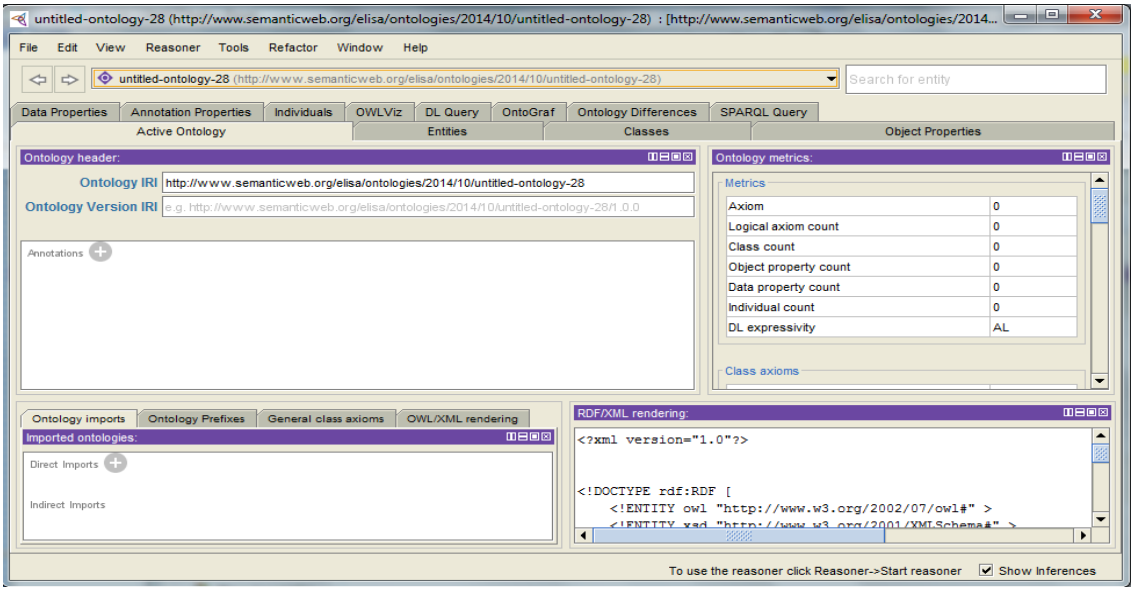

Figura 16 - Interface Protégé

2.4.1.2 OntoEdit

Trata-se de ambiente gráfico para edição de ontologias que permite codificar e alterar ontologias. Foi desenvolvido pela AIFB (Institut für Angewandte Informatik und Formale Beschreibungsverfahren) na Universidade de Karlsruhe. Possui uma arquitetura baseada em

plugins e atualmente está disponível em duas versões: OntoEdit Free e OntoEdit Professional. Permite importar/exportar para Flogic, XML, RDF(S)e DAML+OIL [49]. Na figura 17 é apresentada a interface da ferramenta OntoEdit [51].

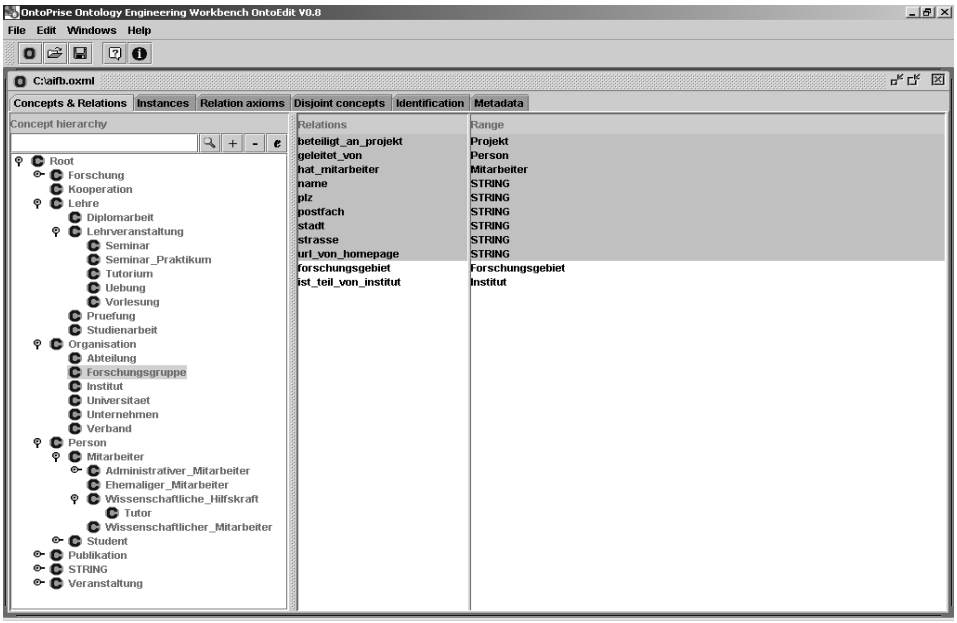

Figura 17 - Interface OntoEdit

2.4.1.3 WebODE

Aplicação desenvolvida no laboratório de inteligência artificial da Universidade Politécnica de Madrid. Trata-se de uma aplicação Web onde as ontologias são armazenadas em bases de dados relacionais. Possui serviços de documentação, avaliação e fusão de ontologias e permite importar/exportar para linguagens como XML, RDF(S) , OIL, DAML + OIL , Flogic e Prolog [49][52]. Na figura 18 é apresentada a interface da aplicação WebODE [52].

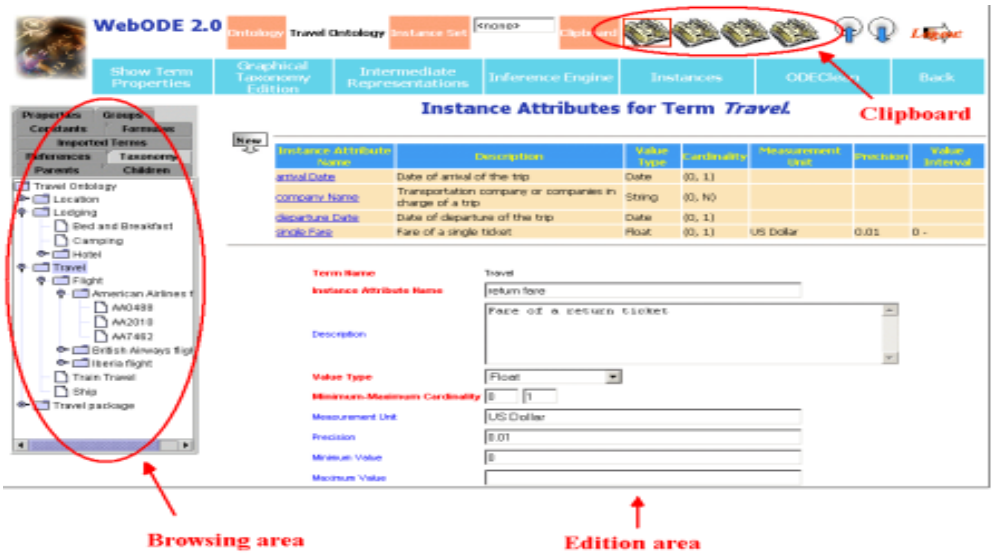

Figura 18 - Interface WebODE

2.5 Mapeamento

Quando se recorre a várias ferramentas e tecnologias distintas, os outputs gerados são, na maior parte das situações, completamente distintos. Quando é necessário relacionar estes outputs, torna-se muitas vezes necessário recorrer a um mapeamento entre eles.

Mapeamento de dados é o processo pelo qual são criadas relações entre duas fontes de dados distintas [53], com o objetivo de criar relações de semelhança entre os conceitos das duas fontes, de forma a obter dados uniformizados e com a possibilidade de efetuar transformações de dados entre os dois modelos. Mapeamento dos dados é usado para mediar a relação entre uma fonte de dados inicial e um destino, no qual os dados são utilizados [53].

Mapeamento de dados pode ser usado com diversos intuitos, entre eles [53]:

- Identificação das relações de dados;
- A transformação de dados ou mediação de dados entre uma fonte de dados origem e uma fonte de dados de destino.

Existem dois tipos de Mapeamento:

- Mapeamento de Conceitos – ocorre quando é criada uma relação de semelhança entre dois conceitos, e.g. entre duas tabelas distintas de duas bases de dados distintas. Entende-se neste caso por relação de semelhança, a definição de semelhança ou igualdade entre dois conceitos (aqui representados por tabelas) diferentes, de forma a identificar conceitos que tenham o mesmo significado.
- Mapeamento de instâncias – ocorre quando é criada uma relação de semelhança entre duas instâncias de Conceitos diferentes. Um exemplo deste tipo de mapeamento ocorre quando é criada uma relação de semelhança entre duas instâncias (linhas ou relações) de tabelas diferentes e de bases de dados diferentes, de forma a identificar instâncias semelhantes.

2.5.1 Ferramentas

2.5.1.1 MAFRA Toolkit (MApping FRAmework Toolkit)

Framework que permite criar relações semânticas entre duas ontologias, a origem e a dedestino, e aplicar essas relações para traduzir instâncias da ontologia fonte em instâncias da ontologia alvo.

Este processo de mapeamento é composto por várias fases conforme representadas na Figura 19 [54].

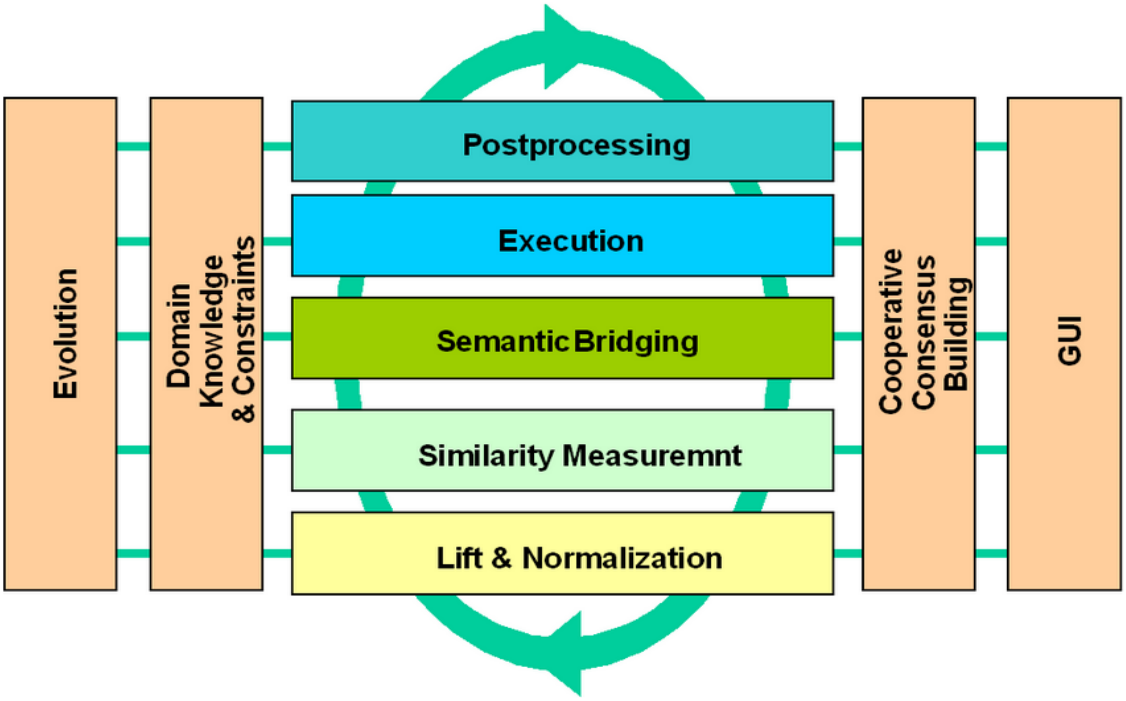

Figura 19 - Descrição conceptual MAFRA

Na figura 20 é apresentada a interface da Framework MAFRA [54].

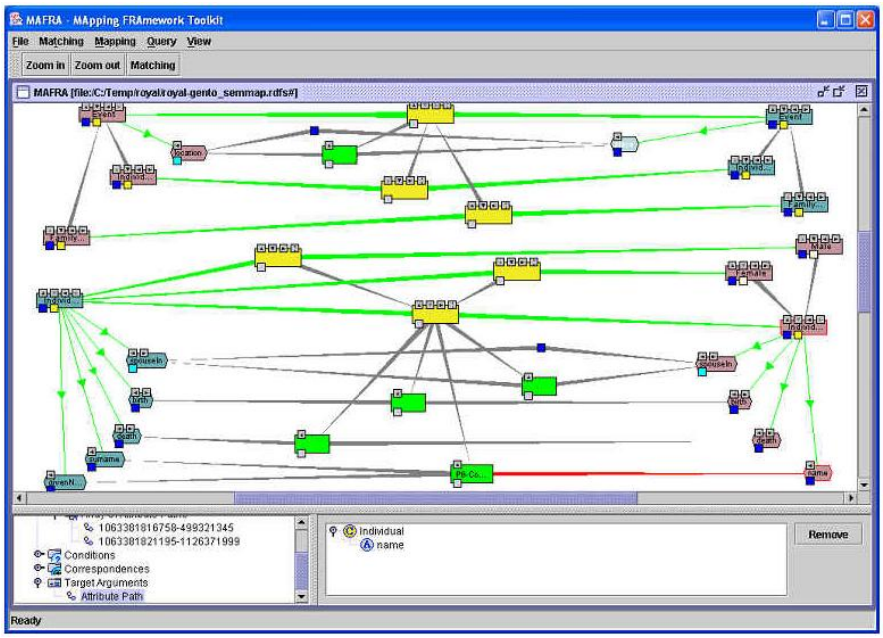

Figura 20- Interface MAFRA

2.5.1.2 MapForce

Trata-se de uma ferramenta para realizar mapeamento e transformação de dados, permitindo realizar estes processos entre dados provenientes de XML, bases de dados, EDI,

XBRL, flat file, Excel, JSON, e Web services. Esta ferramenta possui uma interface que permite realizar mapeamentos de forma visual [55].

Na figura 21 é apresentada a interface da ferramenta MapForce.

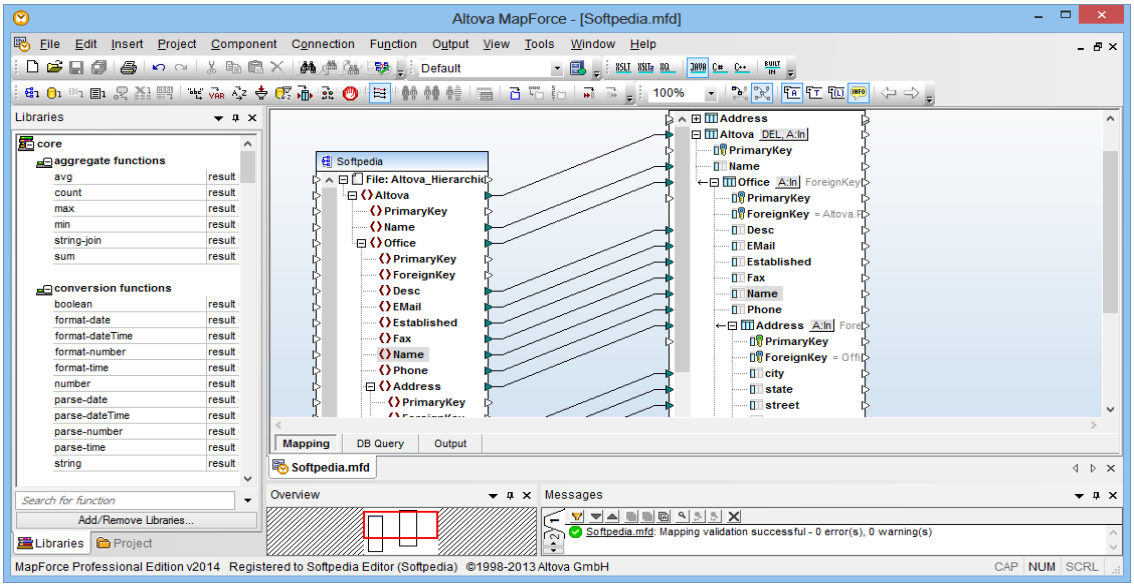

Figura 21 - Interface MoapForce

2.6 Repositórios

Repositórios são coleções de informação digital, que podem ser construídas de diferentes formas e com diferentes propósitos [56]. Estes repositórios de software são locais de armazenamento de informação [57]. Um dos tipos de repositórios atualmente mais comuns são os Sistemas de Gestão de Base de Dados Relacionais (SGBD Relacionais).

As Triplestore (RDF storage) são repositórios construídos com o propósito de armazenar triplos e permitir efetuar consultas [58]. Triplo é uma forma de representação de informação na forma sujeito-predicado-objeto, representando sempre relações de recursos entre si. Apresenta muitas semelhanças com as bases de dados relacionais, pois permite armazenar dados e consultá-los recorrendo a uma linguagem de consulta, apresentando uma grande diferença destas no tipo de dados armazenados, uma vez que as Triplestores estão otimizadas para armazenar triplos, podendo importar e exportar os triplos usando Resource Description Framework (RDF), XML e outros formatos. Existem diversas Triplestores com capacidades de armazenamento diferentes, podendo algumas delas armazenar biliões de triplos. De seguida são apresentados alguns exemplos de Triplestores:

- Jena [59] – trata-se de uma Framework *open source* para Java que possui uma API para extrair dados e criar grafos RDF, e permite realizar queries em SPARQL para consultar os dados dos grafos [60]. Esta Framework suporta a linguagem OWL e a

serialização de grafos RDF para bases de dados relacionais e RDF/XML. Possui uma Triple Store nativa, TDB, e possui um terminal onde é possível realizar queries SPARQL, acessível através de HTTP [59]. Permite ainda ligação a motores de raciocínio Descroption Logics, que raciocinam sobre a base de conhecimento considerando as restrições defindas.

- Ontotext OWLIM [61] – trata-se de um repositório semântico, desenvolvido em Java, e tem como standard para representação de dados a linguagem RDF. Suporta linguagens de consulta como o SPARQL e linguagens como OWL e RDFS. Permite importar e exportar dados em XML, N-Triplos e em Turtle. Possui três edições com algumas especificações distintas: OWLIM-Lite, OWLIM-SE e OWLIM-Enterprise [62].

- SparkleDB [63] – trata-se de uma plataforma de gestão de bases de conhecimento RDF, desenvolvida em C++. A plataforma é uma Triplestore utilizável em grande parte das linguagens de programação e sistemas operativos, e permite realizar queries SPARQL [64].

Na tabela 2 é apresentada uma sistematização das características das TripleStores apresentadas.

Tabela 2 - Sistematização de Triplestores

TripleStore	Linguagem	Query Language	Suporte OWL	Armazenamento
Jena	Java	SPARQL	Sim	Tuple store
Ontotext OWLIM	Java	SPARQL	Sim	3rd party
SparkleDB	C++	SPARQL	Sim	Triplestore

3 Proposta

O problema a ser estudado neste trabalho envolve a utilização da técnica de Web Scraping recorrendo a ferramentas que aplicam esta técnica, obter dados estruturados segundo uma Ontologia e utilizar estes mesmos dados para enriquecer e popular a Ontologia, podendo estes mesmos dados serem usados para outros fins.

3.1 Arquitetura conceptual

Ao ser analisado o problema, foi necessário definir uma série de processos e ações a realizar segundo a interpretação que foi feita do problema. Deste modo, foi desenvolvida uma arquitetura de sistema, representada no diagrama seguinte.

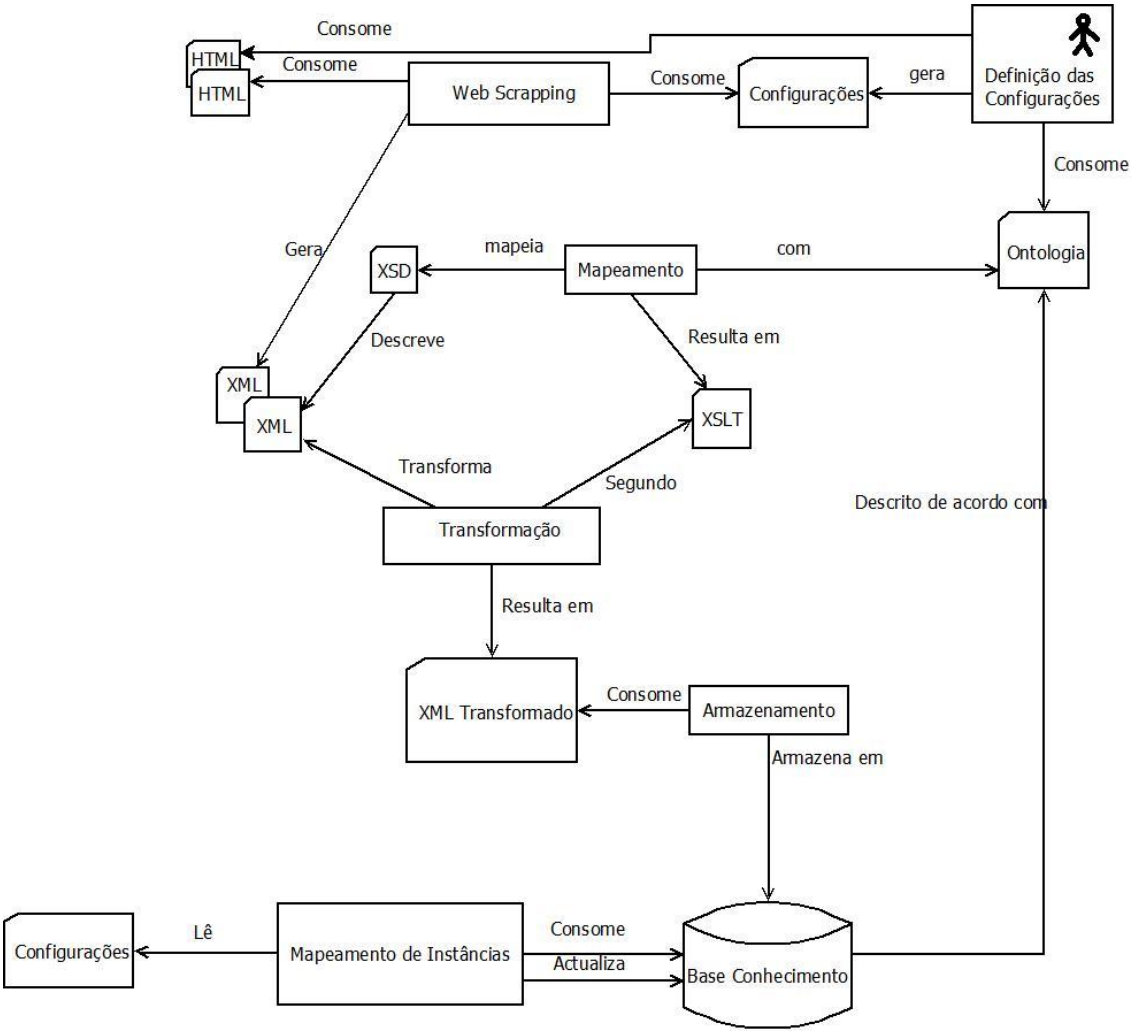

Figura 22 - Diagrama da Arquitetura de Sistema

O processo central, e que serve de ponto de partida para este trabalho é o processo de Web Scraping. Este processo é o responsável por aplicar a técnica de extração de informação. Mas para a aplicação deste processo é necessário definir dois aspetos muito importantes:

- A informação a extrair, definida pelas páginas web a serem consumidas por este processo
- A forma como a informação deverá ser extraída, definida pelas configurações definidas para o processo de extração.

Torna-se então necessário definir, em primeiro lugar, quais as páginas web que vão ser consumidas e da qual será extraída a informação. Estas páginas web são estruturadas em HTML e serão consumidas pelo processo de Web Scraping no processamento e recolha de informação. Por outro lado a escolha do domínio de conhecimento vai influenciar quais as páginas a serem consideradas para a extração.

Torna-se necessário verificar as condições em que o processo de Web Scraping ocorre e quais os fatores que o influenciam, isto é, quais as configurações que devem ser definidas para que a ferramenta possa saber qual a informação a ser extraída no decorrer do processo. Foi então definido o processo de Definição das Configurações, a ser executado pelo utilizador.

Este processo tem como objectivo a definição dos limites que a ferramenta deve ter em conta aquando da extração de informação, gerando este processo um ficheiro de configurações que será tido em conta no processo de Web Scraping. Tendo como exemplo a página web de um Hipermercado, se o objetivo é obter informação de um determinado produto, a Definição das Configurações vai ter que expressar este objetivo nas configurações, de forma a que a ferramenta saiba especificamente quais os objetivos que lhe são propostos e a forma como deve proceder para os atingir.

A análise e extração da informação em cada página, para além de depender das configurações, depende também da estrutura de cada página visitada, uma vez que cada uma delas terá uma estrutura HTML diferente, e consequentemente necessitarão de uma análise diferente, como os próprios dados poderão estar descritos de formas distintas, necessitando o processo de extração ser adaptado a cada uma destas páginas Web. Estas configurações definem em particular os dados a ler na página HTML, algo que é tipicamente definido usando XPath. A geração deste ficheiro de configurações por parte deste processo está portanto dependente da Ontologia, onde estão definidos os conceitos, propriedades e relações do domínio que interessa recolher.

O ficheiro de configurações vai então ser consumido pelo processo de Web Scraping, que vai então proceder à análise das páginas HTML definidas como input e realizar a extração da informação pretendida, respeitando as limitações definidas pelo ficheiro de configurações.

O processo de extração vai retornar os dados extraídos, no formato de XML, gerando um ficheiro por cada página processada. O conteúdo destes ficheiros gerados será influenciado pela informação existente nas páginas, bem como das configurações definidas para o processo de extração, nomeadamente o equema de dados (XSD), i.e. documento que descreve a estrutura dos dados obtidos, os elementos que a compõem, bem como as suas

propriedades.

Apesar do cuidado que deve ser colocado na definição da informação a extrair e do esquema usado para a armazenar nos ficheiros XML, é previsivel que existam algumas diferenças entre o esquema e a Ontologia. Além do mais pode ser necessário interrelacionar dados extraídos de páginas diferentes, e que dão origem a instâncias de conceitos diferentes mas interligados. Por exemplo:

- Um tipo de página contém informação acerca de pessoas, seus endereços e o número do código postal
- Outro tipo de página contém informação acerca do nome das localidades dos códigos postais

Havendo então diferenças entre XSD e Ontologia, é necessário criar relações de equivalência entre este esquema e a ontologia que define o domínio. Para isso serve o processo de mapeamento de conceitos. Este processo tem como objetivo a criação de relações entre os dados obtidos e as definições contidas no ficheiro da Ontologia, tendo este processo como dados de entrada o ficheiro XSD que descreve os outputs da extração e a Ontologia, sendo este processo realizado por uma ferramenta de mapeamento. Durante este processo é analisado o esquema do XSD e o esquema da Ontologia e são criadas relações entre cada um dos conceitos descritos no XSD com os conceitos equivalentes descritos na Ontologia. Deste processo resulta um ficheiro XSLT (eXtensible Stylesheet Language for Transformation), onde são representadas as relações e a forma como as instâncias do XSD são transformadas em instâncias da ontologia.

Este é o processo de Transformação, que recebe os ficheiros XML gerados no processo de Web Scraping e o ficheiro XSLT gerado no processo de Mapeamento, e realiza a transformação dos ficheiros XML segundo o XSLT. A realização deste processo produz um ficheiro XML transformado, diferente dos anteriores, pois este processo adapta os dados de forma a estes refletirem os mapeamentos realizados no processo de Mapeamento.

Agora que se dispõe de dados extraídos na forma pretendida (i.e. segundo a ontologia), é necessário armazená-los, o que é realizado através do processo de Armazenamento. Este processo recebe como input o XML transformado, gerado no processo de Transformação, e processa todos os dados nele existentes. Este processamento dá origem a uma Base de Conhecimento, onde são registados e representados os dados extraídos. Esta Base de Conhecimento, para estar definida de acordo com o domínio e a definição dada ao problema, deve ser definida de acordo com a Ontologia criada, de forma a estar coerente com a representação do conhecimento relativo ao domínio, descrito na Ontologia.

Há ainda necessidade de realizadar um cruzamento dos dados obtidos de forma a comparar dados possivelmente semelhantes, mas obtidos em páginas web distintas. Para realizar esta comparação e definição de dados semelhantes, foi definido o pelo utilizador, de modo a definir as condições do mapeamento, sendo criado manualmente um ficheiro de configurações, que será um dos inputs deste processo. Para aceder aos dados dos objetos,

este processo recorre aos dados constantes na Base de Conhecimento como input, e, recorrendo a software, é realizado um mapeamento entre objetos correspondentes. Realizado este mapeamento, os resultados obtidos são inseridos na Base de Conhecimento, atualizando os dados armazenados.processo Mapeamento de Instâncias. Este processo tem como objectivo mapear objectos semelhantes, para comparar os dados relativos a cada um dos objetos entre si. Este processo necessita de ser configurado

3.2 Implementação

Esta secção descreve o processo de implementação dum protótipo da arquitetura descrita na secção anterior.

Foram realizadas várias experiências, numa fase inicial, com várias das ferramentas referenciadas no Estado da Arte. Foram selecionadas três dessas ferramentas com as quais foram realizados testes de adequação. As ferramentas selecionadas inicialmente foram:

- Mozenda;
- WebSphinx;
- iRobot.

Após alguns testes, apenas a ferramenta Mozenda foi considerada para a implementação de protótipo e para as experiências seguintes, pois verificou-se que a ferramenta WebSphinx era extremamente limitada no que diz respeito ao resultado gerado e a ferramenta iRobot era muito pouco intuitiva na sua utilização.

Decidiu-se por isso a utilização da ferramenta Mozenda para o processo de web scraping.

O processo base e o grande passo da implementação é o processo de Web Scraping. Este processo porém, depende do processo de Definição das Configurações, sendo este realizado de forma manual pelo utilizador.

O processo de Definição de Configurações recebe como input a Ontologia, de forma a definir as configurações de acordo com esta, e gera como output o ficheiro de configurações. A Ontologia é criada recorrendo ao software Protégé [50], um software de criação e edição de Ontologias por parte do utilizador, estando o processo de desenvolvimento da ontologia fora do âmbito deste trabalho.

O processo de configuração pressupõe o seguinte: uma Ontologia já definida, de forma a poder criar as configurações relacionadas com o domínio, sendo também necessário conhecer a forma de funcionamento da ferramenta, dado que este influencia o processo de configurações.

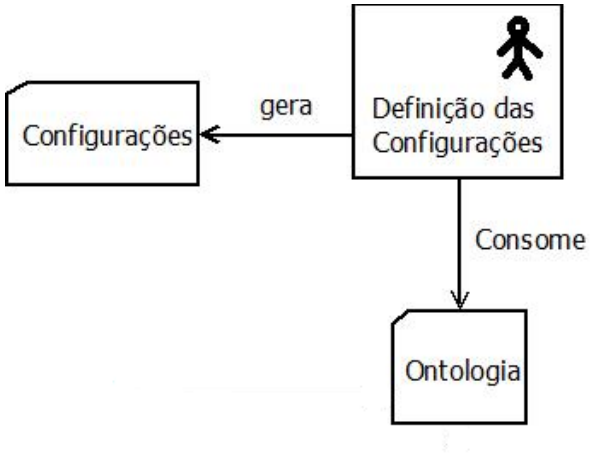

Figura 23 - Processo Definição de Configurações

O ficheiro gerado vai conter então as configurações que definem a informação que vai ser extraída, sendo configurações de domínio:

- <name="Descricao"/>
- <name="Marca"/>
- <name="Quantidade"/>
- <name="Preco"/>
- <name="PrecoUni"/>.

Foram também definidas as configurações técnicas da ferramenta, que definirão o modo de atuação da ferramenta para conseguir os dados pretendidos.

Para implementar o processo de Web Scraping, foi utilizado o software Mozenda [44], e recebe como input o ficheiro de configurações gerado pelo processo de Definição de Configurações, descrito no passo anterior, e recebe também páginas HTML. Estas páginas HTML são analisadas segundo as configurações definidas no ficheiro de configurações, já descrito anteriormente.

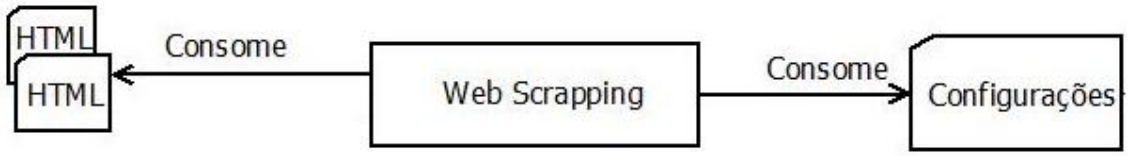

Figura 24 - Input Processo Web Scraping

A ferramenta Mozenda permite exportar os dados obtidos para ficheiros XML, sendo um formato bastante adequado para utilização futura, dado que XML pode ser processado virtualmente por qualquer computador. O resultado do processo de Web Scraping são portanto ficheiros XML (com uma estrutura representado no esquema XSD), com os dados recolhidos das páginas HTML.

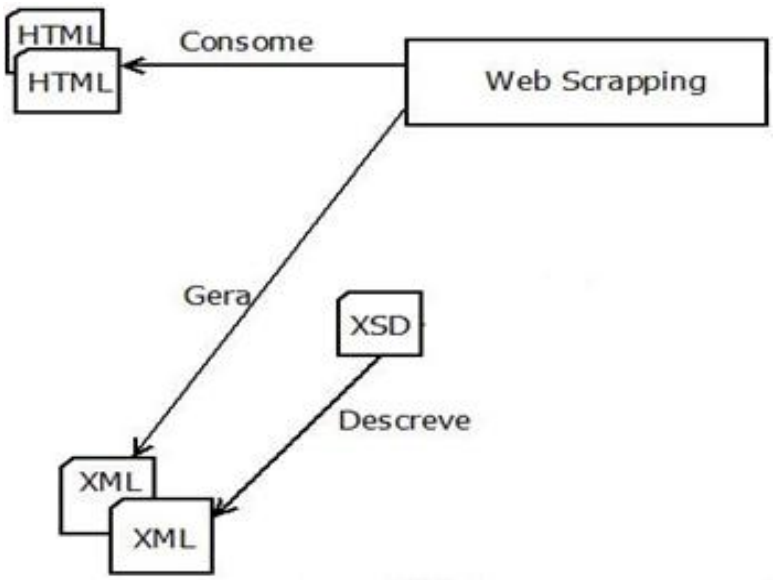

Figura 25 - Resultado Processo Web Scraping

Os ficheiros XML gerados pelo Mozenda foram descritos por um ficheiro XSD, sendo este o primeiro input do processo de Mapeamento. O segundo é a ontologia de domínio de conhecimento. Este processo foi implementado recorrendo ao software MapForce [55].

Este mapeamento gera relações entre os conceitos definidos no XSD e os conceitos de domínio descritos na Ontologia. Este processo pressupõe que o ficheiro XSD esteja bem formado e defina corretamente a estrutura do XML.

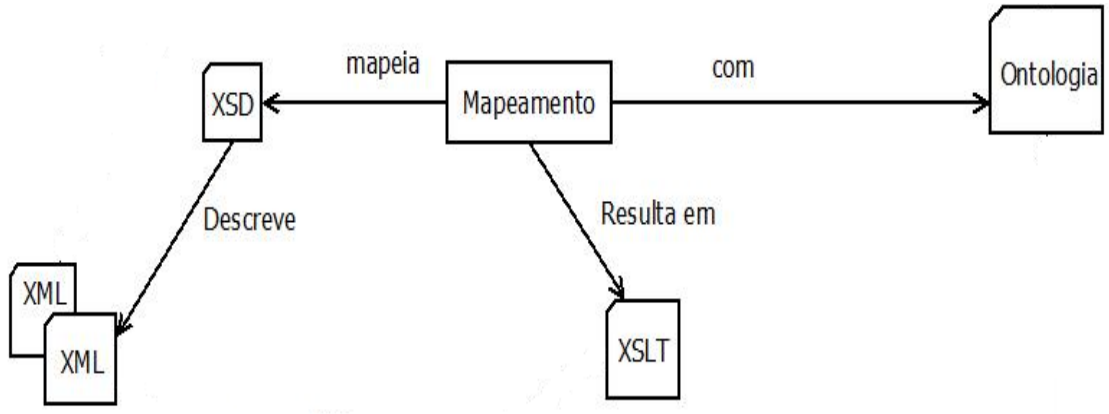

Figura 26 - Processo de Mapeamento

O resultado do processo de mapeamento é um ficheiro XSLT. Este ficheiro será utilizado para transformar os ficheiros XML resultantes do processo de Web Scraping, para que a sua estrutura seja adaptada à definição do domínio, sendo os ficheiros XML e XSLT os inputs do processo de Transformação.

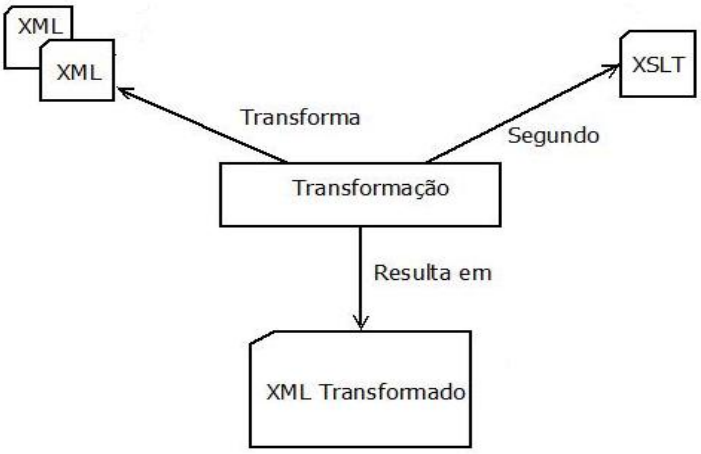

Figura 27 - Processo de Transformação

Este processo vai transformar os ficheiros XML de acordo com o esquema do XSLT, gerando um ficheiro XML transformado como output, com a informação já estruturada segundo o domínio definido pela Ontologia, cujos dados refletem já as relações criadas no processo de mapeamento.

Este XML transformado será então consumido pelo processo de Armazenamento. Este processo é responsável por armazenar os dados contidos no XML transformado para a base de conhecimento, que foi implementada na Framework Jena Fuseki [59], uma Triplestore implementada em Java. Este processo pressupõe que o ficheiro XML inserido é o ficheiro transformado resultante do processo anterior e não os ficheiros originais resultantes do Mozenda, uma vez que é necessário neste processo, que os dados estejam já descritos de acordo com a Ontologia.

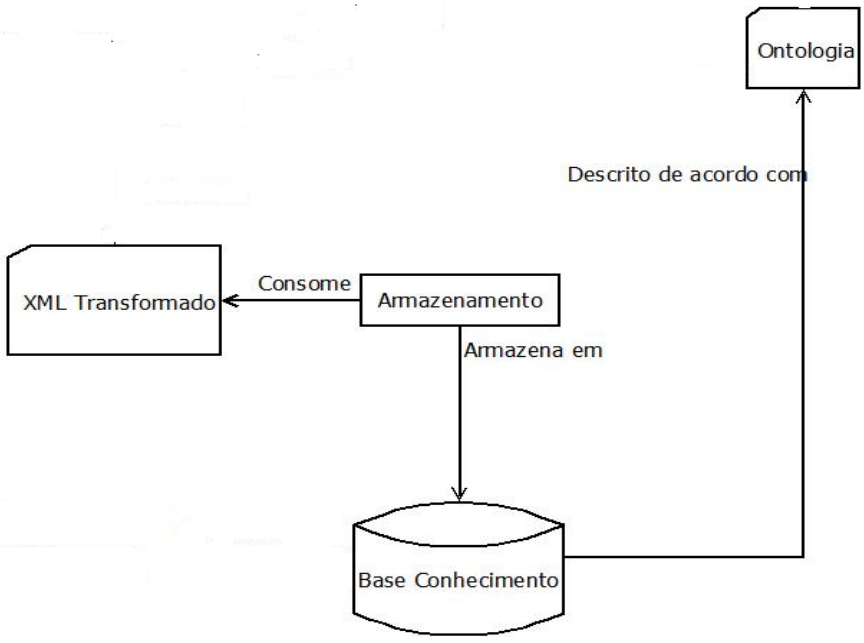

Figura 28 - Processo de Armazenamento

A necessidade do ficheiro de input deste processo ter de ser o ficheiro transformado baseia-se na definição da Base de conhecimento. Esta encontra-se descrita segundo a Ontologia, para que reflita a descrição que foi definida para o domínio, necessitando que os dados estejam também descritos desta forma.

O resultado do processo de Armazenamento é o preenchimento da Base de Conhecimento com os dados do XML. A estrutura da Base de Conhecimento é definida de acordo com a Ontologia, tal como referido.

A Base de Conhecimento serve de input ao processo de Mapeamento de Instâncias, sendo o último processo a realizar. Este processo também consome um ficheiro de configurações, criado manualmente, que guia o processo. O processo tem como output a Base de Conhecimento actualizada. Este processo pretende criar relações entre instâncias da Base de Conhecimento, pressupondo que podem existir várias instâncias que se referem ao mesmo objeto, dado que os dados provêm de fontes distintas e que possuem dados semelhantes. Este processo não foi ainda implementado, sendo um dos pontos a trabalhar no futuro.

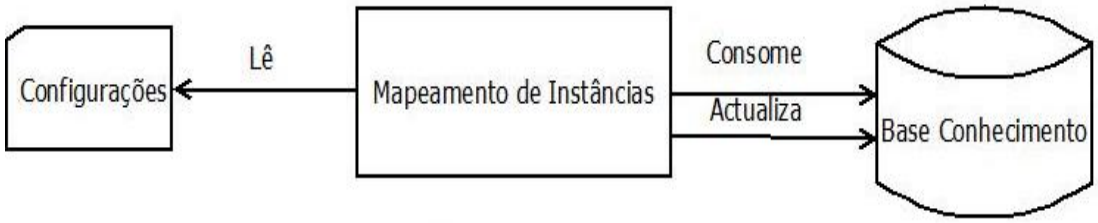

Figura 29 – Processo de Object Mapping

4 Experiências

As experiências realizadas incidiram sobre o domínio de bebidas e mercearia de super-mercados em Portugal. Foram selecionados páginas Web das seguintes cadeias de Super e Hiper Mercados:

- Continente (http://www.continente.pt/pt-pt/public/Pages/homepage.aspx) [65];
- Jumbo (http://www.jumbo.pt/Frontoffice/ContentPages/JumboNetWelcome.aspx) [66].

4.1 Ontologia

A ontologia define a forma como as ferramentas de Web Scraping analisam e extraem os dados, uma vez que a ontologia define a interpretação do domínio. Esta ontologia influencia também a forma como os dados são persistidos, dado que o armazenamento dos dados extraídos tem que estar de acordo com a representação da ontologia. Na figura 30 é apresentado o diagrama que descreve a interpretação do domínio de conhecimento.

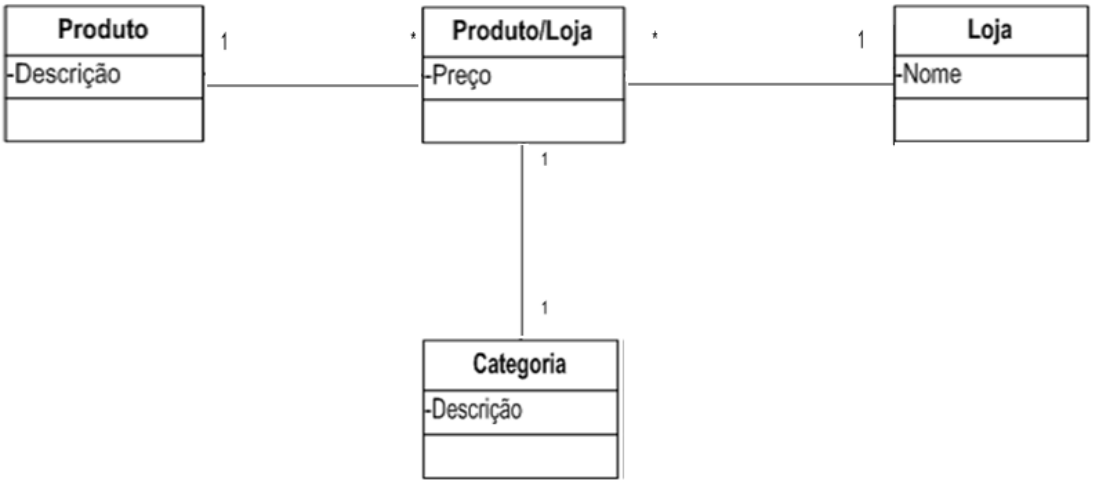

Figura 30 - Interpretação do problema

Na abordagem seguida, a interpretação feita aos dados e a possibilidade de existirem diferenças entre as várias lojas, levaram a criar uma Ontologia que tivesse em consideração esta interpretação. Com esta interpretação foi definido que haveria um Produto, uma Loja e uma Categoria. Foi definido também um Produto/Loja que possui um atributo preço, e que representa um determinado produto referente a uma Loja específica. Este Produto/Loja terá um produto, está à venda numa Loja e pertence a uma Categoria. Esta Ontologia tem em consideração as possíveis diferenças entre preços praticados em Lojas diferentes,

considerando o preço um campo dependente da Loja.

No exemplo de código 7 é apresentada parte da definição da Ontologia criada.

```
<?xml version="1.0"?>
<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.semanticweb.org/elisa/ontologies/2014/6/untitled-
ontology-23"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"

ontologyIRI="http://www.semanticweb.org/elisa/ontologies/2014/6/untitled-
ontology-23">
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Declaration>
        <Class IRI="#Categoria"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Loja"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Produto"/>
    </Declaration>
    <Declaration>
        <Class IRI="#ProdutoLoja"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#temCategoria"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#temLoja"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#temProduto"/>
    </Declaration>
    <Declaration>
        <DataProperty IRI="#temPreco"/>
    </Declaration>
    <SubClassOf>
        <Class IRI="#ProdutoLoja"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#temCategoria"/>
            <Class IRI="#Categoria"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ProdutoLoja"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#temLoja"/>
```

```
            <Class IRI="#Loja"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ProdutoLoja"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#temProduto"/>
            <Class IRI="#Produto"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
</Ontology>
```

Código 7 - Definição da Ontologia

O Código total da Ontologia encontra-se descrito no Anexo 1.

A ontologia foi desenvolvida com recurso à aplicação Protégé. Apesar do desenvolvimento da ontologia estar fora do âmbito deste trabalho, as figuras 26, 27 e 28 apresentam representações parciais da ontologia criada na interface da aplicação Protégé.

Na figura 31 são apresentadas as Classes criadas na Ontologia.

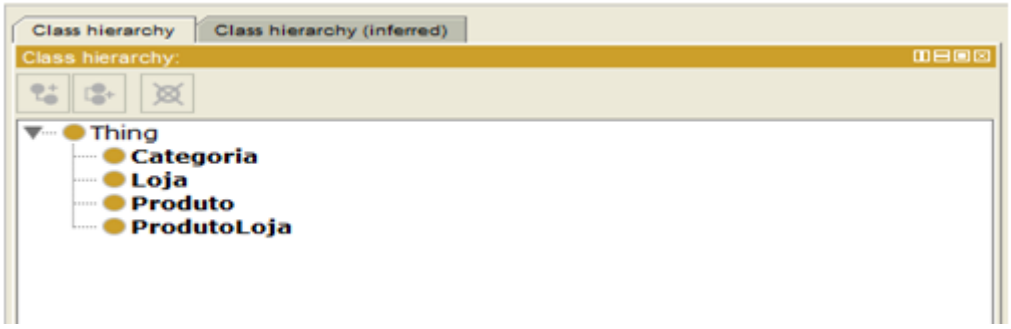

Figura 31 - Ontologia Classes

Na figura 32 são apresentadas as Data Properties criadas na Ontologia.

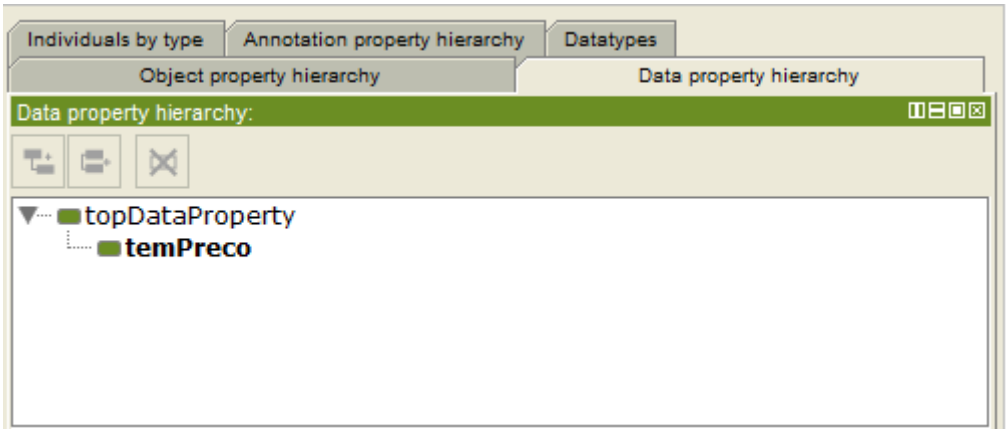

Figura 32 - Ontologia Data Properties

Pela figura 33 é possível verificar as Object Properties criadas na Ontologia.

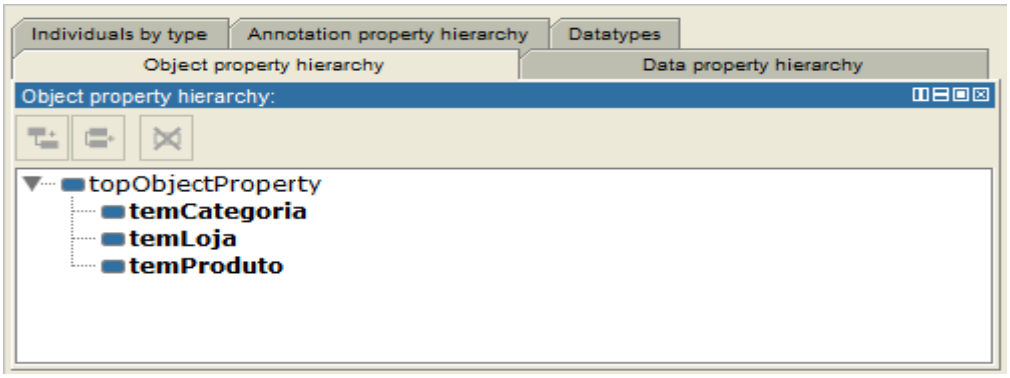

Figura 33 - Ontologia - Object Properties

4.2 Web scraping

Para aceder à informação presente nas várias páginas dos sites escolhidos, foram criados vários agentes no Mozenda, para aceder aos dados dos produtos existentes, sendo um agente para cada categoria. Eram recolhidos os seguintes campos:

- Nome do Produto
- Quantidade de Produto
- Marca
- Preço
- Preço por unidade/quilo

Na figura seguinte é apresentado o resultado destas experiências para um dos Agentes criados, em particular numa página web sobre produtos Frescos no site do Jumbo.

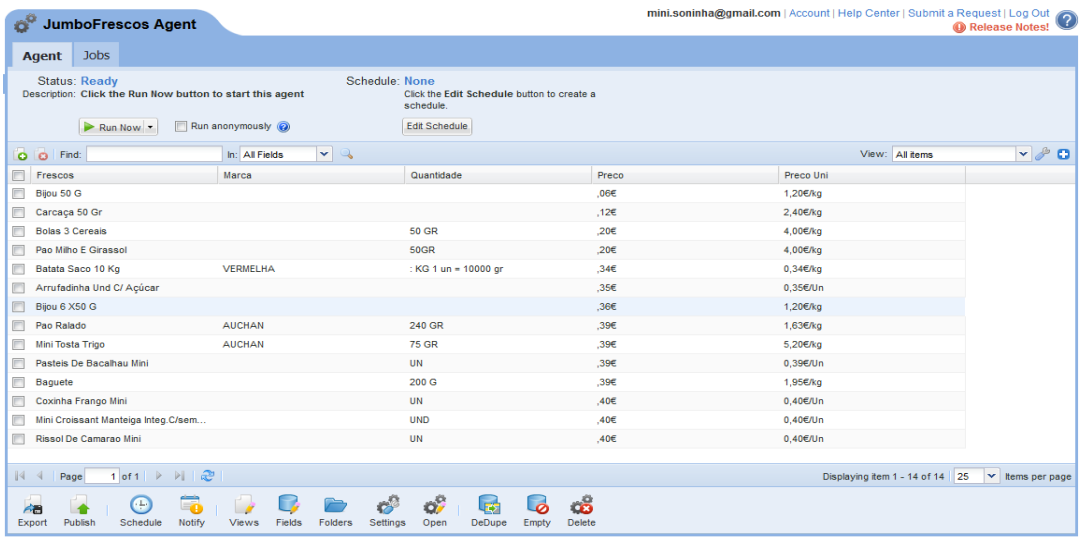

Figura 34 - Resultados Agente

Pela observação da imagem é possível verificar as ações que é possível realizar com os agentes, desde exportação de dados, modificação e agendamento.

A ferramenta Mozenda permite fazer exportação de dados para XML, sendo exportada toda esta informação para um ficheiro. Foi inserido posteriormente, em cada um dos ficheiros criados (um por categoria), o campo “Categoria” e o campo “Loja”. Estes campos foram inseridos manualmente.

Este processo foi repetido para todas as categorias existentes, para a página do Continente e do Jumbo.

Para ficar com apenas um ficheiro XML com todos os resultados obtidos, foi necessário juntar num só ficheiro todos os ficheiros individuais obtidos através do Mozenda.

Os dois extratos de código XML 8 e 9 representam excertos dos dados recolhidos pelo Mozenda, sendo um deles referente a dados extraídos da página do Jumbo e outro da página do Continente.

```
<Allitems>
 <Item>
        <Descricao>Alimento Seco para Cão</Descricao>
        <Marca>Continente</Marca>
        <Quantidade>emb. 20 kg</Quantidade>
        <Preco> 27,99 /un</Preco>
        <PrecoUni> 1,40 /kg</PrecoUni>
        <Categoria>Animais</Categoria>
        <Loja>Continente</Loja>
 </Item>
 <Item>
        <Descricao>Tapete Educação para Cão</Descricao>
        <Marca>Continente</Marca>
        <Quantidade>12 un</Quantidade>
        <Preco> 6,99 /un</Preco>
        <PrecoUni> 6,99 /un</PrecoUni>
        <Categoria>Animais</Categoria>
        <Loja>Continente</Loja>
 </Item>
<Item>
        <Descricao>Snack para Gato Temptations Vaca</Descricao>
        <Marca>Whiskas</Marca>
        <Quantidade>emb. 60 gr</Quantidade>
        <Preco> 2,39 /un</Preco>
        <PrecoUni> 47,80 /kg</PrecoUni>
        <Categoria>Animais</Categoria>
        <Loja>Continente</Loja>
 </Item>
 <Item>
        <Descricao>Snack para Gato Temptations Salmão</Descricao>
        <Marca>Whiskas</Marca>
        <Quantidade>emb. 60 gr</Quantidade>
        <Preco> 2,39 /un</Preco>
        <PrecoUni> 39,83 /kg</PrecoUni>
        <Categoria>Animais</Categoria>
        <Loja>Continente</Loja>
 </Item>
```

```
</Allitems>
```

Código 8 - XML Loja Continente

```
<Allitems>
 <Item>
        <Descricao>Esponja De Banho Bébé Suave</Descricao>
        <Marca>AUCHAN</Marca>
        <Quantidade>1 UN</Quantidade>
        <Preco>,77</Preco>
        <PrecoUni>0,77/Un</PrecoUni>
        <Categoria>Bebé</Categoria>
        <Loja>Jumbo</Loja>
 </Item>
 <Item>
        <Descricao>Bolacha Sanduiche Creamy Kiss Baunilha</Descricao>
        <Marca>DAN CAKE</Marca>
        <Quantidade>180 G</Quantidade>
        <Preco>,79</Preco>
        <PrecoUni>4,39/kg</PrecoUni>
        <Categoria>Bebé</Categoria>
        <Loja>Jumbo</Loja>
 </Item>
 <Item>
        <Descricao>Bolacha Creme Junior</Descricao>
        <Marca>GULLON</Marca>
        <Quantidade>170 G</Quantidade>
        <Preco>,79</Preco>
        <PrecoUni>4,65/kg</PrecoUni>
        <Categoria>Bebé</Categoria>
        <Loja>Jumbo</Loja>
 </Item>
 <Item>
        <Descricao>Bolacha Sanduiche Morango</Descricao>
        <Marca>CREAMY KISS</Marca>
        <Quantidade>180 GR</Quantidade>
        <Preco>,79</Preco>
        <PrecoUni>4,39/kg</PrecoUni>
        <Categoria>Bebé</Categoria>
        <Loja>Jumbo</Loja>
 </Item>
</Allitems>
```

Código 9 - XML Loja Jumbo

4.3 Mapeamento de conceitos

Foi necessário realizar mapeamentos entre a Ontologia criada e os esquemas dos ficheiros XML resultantes do web scraping realizado pela ferramenta Mozenda, uma vez que se tornou necessário criar um relacionamento entre estes resultados e a definição da Ontologia, de forma a relacionar os conceitos semelhantes entre o XML e o domínio. Para tal foi utilizada a aplicação MapForce.

Uma vez que a definição do modelo através de uma Ontologia é definida em OWL [67], uma

linguagem para definição e instanciação de Ontologias [67], e a exportação dos dados obtidos através do Mozenda é realizada para ficheiros XML, torna-se necessário realizar um mapeamento dos dados presentes nos ficheiros XML, de forma a criar relações de proximidade entre esses dados e a definição da Ontologia, de forma a relaciona-los com o modelo do problema, isto é, relacionar a definição dos produtos obtida a partir das páginas web dos supermercados com a descrição do problema feita pela Ontologia.

Foi necessário realizar o mapeamento dos resultados obtidos pelo Mozenda, de acordo com a ontologia, usando um XML com o esquema da ontologia para mapear os resultados.

De seguida é apresentado o XSD usado para fazer o mapeamento do XML resultante do Mozenda.

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Allitems">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Item" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Descricao"/>
              <xs:element type="xs:string" name="Marca"/>
              <xs:element type="xs:string" name="Quantidade"/>
              <xs:element type="xs:string" name="Preco"/>
              <xs:element type="xs:string" name="PrecoUni"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Código 10 – XML Base para Mapeamento da Ontologia

Foi analisada a ontologia criada, sendo realizado um mapeamento entre esta e o XSD descritivo dos dados. O mapeamento segundo a Ontologia resultou no esquema que é apresentado na figura 35.

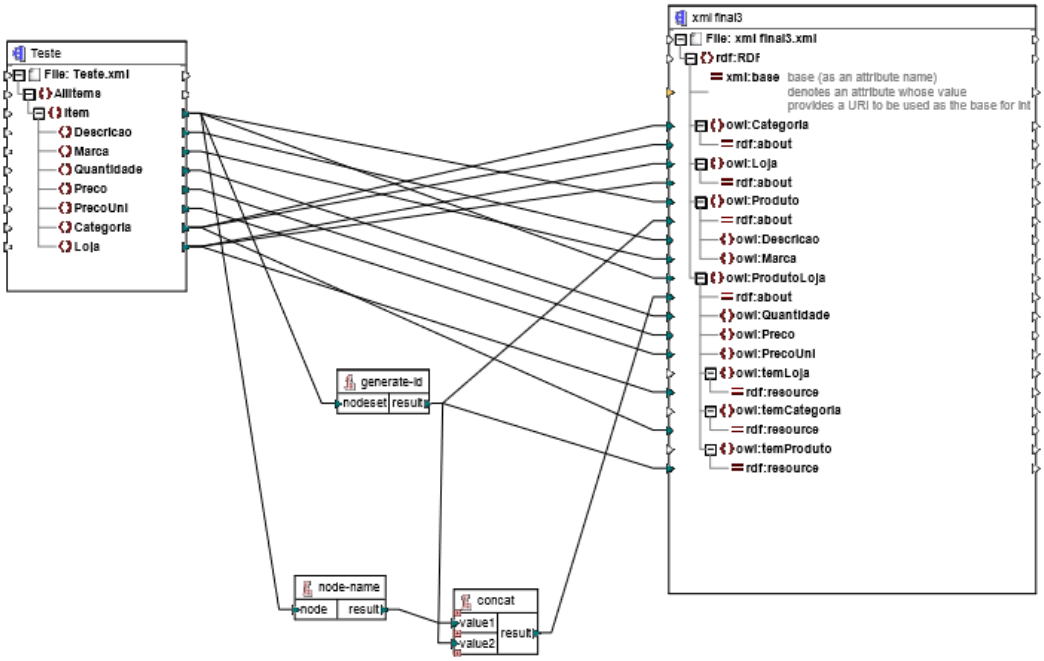

Figura 35 – Mapeamento da Ontologia

O XSLT criado com este mapeamento encontra-se definido abaixo.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/xpath-functions" exclude-result-
prefixes="xs fn">
 <xsl:output method="xml" encoding="UTF-8" byte-order-mark="no"
indent="yes"/>
 <xsl:template match="/">
       <xsl:variable name="var1_Allitems" as="node()?" select="Allitems"/>
       <RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#">
              <xsl:attribute name="xsi:schemaLocation"
namespace="http://www.w3.org/2001/XMLSchema-instance"
select="'http://www.w3.org/1999/02/22-rdf-syntax-ns#
C:/Users/Elisa/Desktop/BACKUP~2/TESTES~1/TESTEF~1/XMLFIN~1.XSD'"/>
              <xsl:for-each select="$var1_Allitems/*:Item[fn:namespace-
uri() eq '']">
                     <owl:Categoria>
                           <xsl:attribute name="about"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="fn:string(*:Categoria[fn:namespace-uri() eq ''])"/>
                     </owl:Categoria>
              </xsl:for-each>
<xsl:for-each select="$var1_Allitems/*:Item[fn:namespace-uri() eq '']">
                     <xsl:variable name="var2_resultof_generate_id"
as="xs:string" select="generate-id(.)"/>
                     <owl:ProdutoLoja>
<owl:temCategoria>
```

```
                                    <xsl:attribute name="resource"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="fn:string(*:Categoria[fn:namespace-uri() eq ''])"/>
                                </owl:temCategoria>
                                <owl:temProduto>
                                    <xsl:attribute name="resource"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="$var2_resultof_generate_id"/>
                                </owl:temProduto>
                        </owl:ProdutoLoja>
                </xsl:for-each>
        </RDF>
  </xsl:template>
</xsl:stylesheet>
```

Código 11 - XSLT Mapeamento da Ontologia

O resultado da execução da transformação é um ficheiro XML transformado segundo o ficheiro XSLT, que é o ficheiro que serve de fonte ao processo de armazenamento.

Abaixo é apresentado o XML gerado com a transformação do XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/1999/02/22-rdf-syntax-ns#
C:/Users/Elisa/Desktop/BACKUP~2/TESTES~1/TESTEF~1/XMLFIN~1.XSD">
  <owl:Categoria xmlns:ns0="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns0:about="Animais"/>
  <owl:Categoria xmlns:ns1="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns1:about="Animais"/>
  <owl:Loja xmlns:ns2="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns2:about="Continente"/>
  <owl:Loja xmlns:ns3="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns3:about="Jumbo"/>
  <owl:Produto xmlns:ns4="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns4:about="idelem4x3">
        <owl:Descricao>Alimento Seco para Cão</owl:Descricao>
        <owl:Marca>Continente</owl:Marca>
  </owl:Produto>
  <owl:Produto xmlns:ns5="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns5:about="idelem4x27">
        <owl:Descricao>Alimento Seco para Cão</owl:Descricao>
        <owl:Marca>Continente</owl:Marca>
  </owl:Produto>
  <owl:ProdutoLoja xmlns:ns6="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns6:about="Itemidelem4x3">
        <owl:Quantidade>emb. 20 kg</owl:Quantidade>
        <owl:Preco> 27,99 /un</owl:Preco>
        <owl:PrecoUni> 1,40 /kg</owl:PrecoUni>
        <owl:temLoja xmlns:ns7="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" ns7:resource="Continente"/>
        <owl:temCategoria xmlns:ns8="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" ns8:resource="Animais"/>
<owl:temProduto xmlns:ns9="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns9:resource="idelem4x3"/>
  </owl:ProdutoLoja>
```

```
 <owl:ProdutoLoja xmlns:ns10="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns10:about="Itemidelem4x27">
        <owl:Quantidade>emb. 20 kg</owl:Quantidade>
        <owl:Preco> 27,99 /un</owl:Preco>
        <owl:PrecoUni> 1,40 /kg</owl:PrecoUni>
        <owl:temLoja xmlns:ns11="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" ns11:resource="Jumbo"/>
        <owl:temCategoria xmlns:ns12="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" ns12:resource="Animais"/>
        <owl:temProduto xmlns:ns13="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" ns13:resource="idelem4x27"/>
 </owl:ProdutoLoja>
</RDF>
```

Código 12 - XML transformado

Foi realizada a inserção do XML transformado na Triple Store, sendo apresentado de seguida o resultado de uma query realizada à base de conhecimento.

a	b	c
<http://example/upload-base/425>	<http://www.w3.org/2002/07/owl#PrecoUni>	"2,95/kg"
<http://example/upload-base/427>	<http://www.w3.org/2002/07/owl#Marca>	"SALUTEM"
<http://example/upload-base/228>	<http://www.w3.org/2002/07/owl#PrecoUni>	" 112,50 /lt"
<http://example/upload-base/211>	<http://www.w3.org/2002/07/owl#PrecoUni>	"€ 2,38 /dúzia"
<http://example/upload-base/253>	<http://www.w3.org/2002/07/owl#Marca>	"AUCHAN"
<http://example/upload-base/414>	<http://www.w3.org/2002/07/owl#Marca>	"SALEXPOR"
<http://example/upload-base/12>	<http://www.w3.org/2002/07/owl#Preco>	" 2,39 /un"
<http://example/upload-base/363>	<http://www.w3.org/2002/07/owl#Descricao>	"Natas Uht"
<http://example/upload-base/6>	<http://www.w3.org/2002/07/owl#temCategoria>	<http://example/upload-base/Animais>
<http://example/upload-base/74>	<http://www.w3.org/2002/07/owl#Marca>	"Uso"
<http://example/upload-base/235>	<http://www.w3.org/2002/07/owl#Marca>	"Elancyl"
<http://example/upload-base/344>	<http://www.w3.org/2002/07/owl#PrecoUni>	"0,52/Un"
<http://example/upload-base/54>	<http://www.w3.org/2002/07/owl#Descricao>	"Vinho Tinto Alentejo Grande Escolha"
<http://example/upload-base/252>	<http://www.w3.org/2002/07/owl#Marca>	"AUCHAN"
<http://example/upload-base/2>	<http://www.w3.org/2002/07/owl#Preco>	" 6,99 /un"
<http://example/upload-base/13>	<http://www.w3.org/2002/07/owl#Preco>	" 2,39 /un"
<http://example/upload-base/7>	<http://www.w3.org/2002/07/owl#temCategoria>	<http://example/upload-base/Animais>
<http://example/upload-base/105>	<http://www.w3.org/2002/07/owl#Descricao>	"Polvo 1 a 2 Kg Congelado"
<http://example/upload-base/75>	<http://www.w3.org/2002/07/owl#Marca>	"Uso"
<http://example/upload-base/234>	<http://www.w3.org/2002/07/owl#Marca>	"Elancyl"
<http://example/upload-base/107>	<http://www.w3.org/2002/07/owl#PrecoUni>	"€ 8,33 /kg"
<http://example/upload-base/87>	<http://www.w3.org/2002/07/owl#PrecoUni>	" 17,58 /kg"
<http://example/upload-base/168>	<http://www.w3.org/2002/07/owl#Descricao>	"Pioneses Coloridos"
<http://example/upload-base/114>	<http://www.w3.org/2002/07/owl#Preco>	"€ 2,00 /un"
<http://example/upload-base/251>	<http://www.w3.org/2002/07/owl#Marca>	"AUCHAN"
<http://example/upload-base/8>	<http://www.w3.org/2002/07/owl#temCategoria>	<http://example/upload-base/Animais>
<http://example/upload-base/72>	<http://www.w3.org/2002/07/owl#Marca>	"Uso"
<http://example/upload-base/16>	<http://www.w3.org/2002/07/owl#Quantidade>	"emb. 60 gr"
<http://example/upload-base/115>	<http://www.w3.org/2002/07/owl#Quantidade>	"emb. 300 gr"
<http://example/upload-base/354>	<http://www.w3.org/2002/07/owl#Descricao>	"Leite C/chocolate Emb."
<http://example/upload-base/113>	<http://www.w3.org/2002/07/owl#Preco>	"€ 2,00 /un"
<http://example/upload-base/432>	<http://www.w3.org/2002/07/owl#Descricao>	"Pastilha Elastica Go Up S/aç Freshmint"
<http://example/upload-base/250>	<http://www.w3.org/2002/07/owl#Marca>	"AUCHAN"
<http://example/upload-base/145>	<http://www.w3.org/2002/07/owl#Quantidade>	"emb. 2 x 115 gr"
<http://example/upload-base/9>	<http://www.w3.org/2002/07/owl#temCategoria>	<http://example/upload-base/Animais>
<http://example/upload-base/236>	<http://www.w3.org/2002/07/owl#Marca>	"Elancyl"
<http://example/upload-base/73>	<http://www.w3.org/2002/07/owl#Marca>	"Uso"

Figura 36 – Query à Triple Store criada

Os ficheiros XSLT e XML são representados na íntegra nos Anexos 2 e 3 respectivamente.

5 Conclusões e Trabalho Futuro

O rápido crescimento da informação presente na Web, aliado às ferramentas existentes que permitem fazer uma recolha desta de muitas e distintas fontes, permite aceder a informação útil na tomada de decisões por parte do utilizador.

Os objetivos deste trabalho foram a aplicação da técnica de Web Scraping para a recolha de informação da web, manipular, transformar e estruturar todos estes dados tendo como modelo uma Ontologia, sendo esta Ontologia povoada e enriquecida com a informação estruturada recolhida automaticamente.

No âmbito desta Dissertação foi desenvolvida uma arquitetura de web scraping configurável por domínio, e o desenvolvimento dum prótipo que a implementa recorrendo em particular a:

- Mozenda como ferramentas de Web Scraping;
- MapForce como ferramentas de manipulação de XML, XSD e XSLT;
- Jena Fuseki e TDB como repositório semântico da informação recolhida.

As experiências realizadas permitiram recolher informação relativa a várias categorias de produtos e produtos de mercearia comercializados on-line a partir das páginas Web de várias cadeias de híper-mercados.

A informação não estruturada e dispersa das páginas web passa assim a estar disponível estruturadamente num único repositório semântico, o que permite pesquisar toda informação formalmente e estruturadamente (e.g. via SPARQL).

A utilização desta ferramenta permitiu cumprir o objetivo de extração de informação, concluindo que o uso da técnica de Web Scraping pode tornar-se muito útil para a recolha de informação na Web, de forma a ser utilizada para o âmbito que o utilizador necessitar.

A aplicação das funcionalidades da ferramenta MapForce permitiu a ligação dos dados obtidos com o modelo da Ontologia, de forma a poder manipular corretamente os dados e estrutura-los de acordo com a Ontologia.

5.1 Limitações

Durante a implementação de todo o trabalho necessário para esta dissertação foram encontradas algumas limitações ao nível das ferramentas utilizadas.

As ferramentas utilizadas durante as experiências eram todas elas gratuitas, pelo que apresentavam algumas limitações ao nível das interfaces, sendo por vezes muito pouco intuitivas. Uma dessas é a ferramenta Scrapy que apenas apresenta uma interface de Linha de comandos pouco intuitiva para um utilizador pouco experiente. A escassa e deficiente

documentação que algumas das ferramentas apresentavam nada contribuía para entender o funcionamento das ferramentas. O facto de serem gratuitas limitava também as funcionalidades disponibilizadas, chegando na ferramenta Mozenda a limitar a quantidade de informação a que se conseguia aceder em cada página e até a limitar o número de páginas a serem pesquisadas, bem como o número de agentes de extração que eram permitidos criar. Estas situações levaram a que a informação obtida fosse muito menor que aquela que seria esperada, dada a quantidade de produtos disponíveis.

A solução apresenta algumas limitações ao nível da visualização dos dados, pois esta não inclui uma interface de visualização. O facto de apenas se ter aplicado uma ferramenta de Web Scraping, não permite realizar uma comparação de resultados entre ferramentas neste caso específico.

Foi também concluído durante a implementação que a utilização desta técnica pode pôr em causa alguns requisitos legais, tal como violar os termos de uso de algumas páginas web.

5.2 Trabalho Futuro

O trabalho realizado nesta dissertação focou-se na recolha de informação web não estruturada e a seu armazenamento de acordo com o domínio de conhecimento capturado numa Ontologia. Mas surgem muitos outros pontos que podem ser implementados partindo destes resultados.

A solução encontrada para este trabalho, apesar de ter cumprido os objetivos, não permite que a informação seja consultada pelo utilizador comum que não seja capaz de utilizar SPARQL. O desenvolvimento duma aplicação que forneça uma interface de consulta dos dados recolhidos é uma das melhorias a implementar, relativamente à solução apresentada.

Dado que o principal objetivo do trabalho era aplicar a técnica de Web Scraping para obter dados estruturados, não foi contemplado neste trabalho qual o fim a dar aos dados após a realização deste processo. Uma possibilidade encontrada foi, encontrar uma solução para realizar uma comparação de produtos e preços entre si, de forma a encontrar diferenças entre lojas distintas, sendo para isso necessário implementar o processo de Mapeamento de Instâncias, referido na proposta de arquitetura.

Uma vez que neste trabalho apenas foram contemplados os hipermercados Continente e Jumbo, dada a acessibilidade da informação, um outro ponto a ser trabalhado no futuro é a estender as experiências a outros sites, cuja informação se encontra restrita a utilizadores identificados através de login.

6 Referências

[1] A. Josi, L. A. Abdillah, e Suryayusra, «Penerapan teknik web scraping pada mesin pencari artikel ilmiah», *ArXiv14105777 Cs*, Out. 2014.

[2] «HTML5». [Online]. Disponível em: http://www.w3.org/TR/html5/. [Acedido: 16-Dez-2014].

[3] «Web Scraping», *About*. [Online]. Disponível em: http://onlinebusiness.about.com/od/searchengines/g/Web-Scraping.htm. [Acedido: 05-Out-2014].

[4] S. K. Arora, J. Youtie, P. Shapira, L. Gao, e T. Ma, «Entry strategies in an emerging technology: a pilot web-based study of graphene firms», *Scientometrics*, vol. 95, n. 3, pp. 1189–1207, Jun. 2013.

[5] X. Niu, H. Wang, G. Wu, G. Qi, e Y. Yu, «Evaluating the Stability and Credibility of Ontology Matching Methods», em *The Semantic Web: Research and Applications*, G. Antoniou, M. Grobelnik, E. Simperl, B. Parsia, D. Plexousakis, P. D. Leenheer, e J. Pan, Eds. Springer Berlin Heidelberg, 2011, pp. 275–289.

[6] N. Silva, «An Approach to Ontology Mapping Negotiation», em *In Proceedings of the Workshop on Integrating Ontologies*, 2005.

[7] J. B.-G. José Ignacio Fernández-Villamor, «A Semantic Scraping Model for Web Resources - Applying Linked Data to Web Page Screen Scraping.», pp. 451–456, 2011.

[8] «HTML/Specifications - W3C Wiki». [Online]. Disponível em: http://www.w3.org/wiki/HTML/Specifications. [Acedido: 16-Dez-2014].

[9] «Extensible Markup Language (XML)». [Online]. Disponível em: http://www.w3.org/XML/. [Acedido: 16-Dez-2014].

[10] Y. Lin, B. Li, e B. Liang, «Differentiated Data Persistence with Priority Random Linear Codes», em *27th International Conference on Distributed Computing Systems, 2007. ICDCS '07*, 2007, pp. 47–47.

[11] Y. Lin, B. Liang, e B. Li, «Priority Random Linear Codes in Distributed Storage Systems», *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, n. 11, pp. 1653–1667, 2009.

[12] «SEO e Dados Estruturados | MATERA Systems». .

[13] «RDFa in XHTML: Syntax and Processing». [Online]. Disponível em: http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014/. [Acedido: 16-Dez-2014].

[14] «About microformats - Webmaster Tools Help». [Online]. Disponível em: https://support.google.com/webmasters/answer/146897?hl=en. [Acedido: 25-Out-2014].

[15] L. A. Sabucedo e L. A. Rifon, «A microformat based approach for crawling and locating services in the eGovernment domain», em *24th International Symposium on Computer and Information Sciences, 2009. ISCIS 2009*, 2009, pp. 111–116.
[16] «XML Path Language (XPath)». [Online]. Disponível em: http://www.w3.org/TR/xpath/. [Acedido: 09-Out-2014].
[17] «What is HTML (Hypertext Markup Language)? - Definition from WhatIs.com». [Online]. Disponível em: http://searchsoa.techtarget.com/definition/HTML. [Acedido: 05-Out-2014].
[18] «HTML (HyperText Markup Language)», *Mozilla Developer Network*. [Online]. Disponível em: https://developer.mozilla.org/en-US/docs/Web/HTML. [Acedido: 06-Out-2014].
[19] E. Ferrara, P. De Meo, G. Fiumara, e R. Baumgartner, «Web Data Extraction, Applications and Techniques: A Survey», *ArXiv12070246 Cs*, Jul. 2012.
[20] «XHTML™ Modularization 1.1 - Second Edition». [Online]. Disponível em: http://www.w3.org/TR/xhtml-modularization/. [Acedido: 16-Dez-2014].
[21] «What is the Document Object Model?». [Online]. Disponível em: http://www.w3.org/TR/DOM-Level-3-Core/introduction.html. [Acedido: 07-Out-2014].
[22] Y. Zhai e B. Liu, «Structured Data Extraction from the Web Based on Partial Tree Alignment», *IEEE Trans. Knowl. Data Eng.*, vol. 18, n. 12, pp. 1614–1628, Dez. 2006.
[23] Y. Zhai e B. Liu, «Web Data Extraction Based on Partial Tree Alignment», em *Proceedings of the 14th International Conference on World Wide Web*, New York, NY, USA, 2005, pp. 76–85.
[24] «Trabalhando com DOM em Javascript». [Online]. Disponível em: http://www.devmedia.com.br/trabalhando-com-dom-em-javascript/29039. [Acedido: 30-Nov-2014].
[25] «WysiWyg Web Wrapper Factory (W4F)». [Online]. Disponível em: http://db.cis.upenn.edu/DL/WWW8/. [Acedido: 15-Out-2014].
[26] A. Sahuguet e F. Azavant, «Building Light-Weight Wrappers for Legacy Web Data-Sources Using W4F», em *Proceedings of the 25th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 1999, pp. 738–741.
[27] G. Gottlob e C. Koch, «Logic-based Web Information Extraction», *SIGMOD Rec*, vol. 33, n. 2, pp. 87–94, Jun. 2004.
[28] G. Gottlob e C. Koch, «Monadic Datalog and the Expressive Power of Languages for Web Information Extraction», *J ACM*, vol. 51, n. 1, pp. 74–113, Jan. 2004.
[29] R. Baumgartner, S. Flesca, e G. Gottlob, «Visual Web Information Extraction with Lixto», em *Proceedings of the 27th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 2001, pp. 119–128.
[30] R. Baumgartner, S. Flesca, e G. Gottlob, «The Elog Web Extraction Language», em *Proceedings of the Artificial Intelligence on Logic for Programming*, London, UK, UK, 2001, pp. 548–560.
[31] N. Kushmerick, «Wrapper Induction: Efficiency and Expressiveness», *Artif. Intell.*, vol. 118, p. 2000, 2000.
[32] S. Soderland, «Learning Information Extraction Rules for Semi-Structured and Free Text», *Mach Learn*, vol. 34, n. 1–3, pp. 233–272, Fev. 1999.
[33] M. E. Califf, R. J. Mooney, e D. Cohn, «Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction», 2003, pp. 328–334.
[34] M. E. Califf e R. J. Mooney, «Relational Learning of Pattern-match Rules for Information Extraction», em *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, Menlo Park, CA, USA, 1999, pp. 328–334.

[35] X. Phan, S. Horiguchi, e T. Ho, «Automated Data Extraction from the Web with Conditional Models», *Int J Bus Intell Data Min*, vol. 1, n. 2, pp. 194–209, Dez. 2005.
[36] M.-W. Chang, L. Ratinov, e D. Roth, «Structured Learning with Constrained Conditional Models», *Mach Learn*, vol. 88, n. 3, pp. 399–431, Set. 2012.
[37] J. Turmo, A. Ageno, e N. Català, «Adaptive Information Extraction», *ACM Comput Surv*, vol. 38, n. 2, Jul. 2006.
[38] «WebSPHINX: A Personal, Customizable Web Crawler». [Online]. Disponível em: http://www.cs.cmu.edu/~rcm/websphinx/. [Acedido: 30-Set-2014].
[39] «Jakarta Regexp - Apache Attic». [Online]. Disponível em: http://attic.apache.org/projects/jakarta-regexp.html. [Acedido: 10-Dez-2014].
[40] «Scrapy | An open source web scraping framework for Python». [Online]. Disponível em: http://scrapy.org/. [Acedido: 30-Set-2014].
[41] «Command line tool — Scrapy 0.24.4 documentation». [Online]. Disponível em: http://doc.scrapy.org/en/latest/topics/commands.html. [Acedido: 08-Out-2014].
[42] «IRobotSoft -- Visual Web Scraping and Web Automation Tool for FREE». [Online]. Disponível em: http://www.irobotsoft.com/. [Acedido: 30-Set-2014].
[43] «Get started with screenscraping using Google Chrome's Scraper extension», *dataist*. .
[44] «Data Extraction, Web Screen Scraping Tool, Mozenda Scraper». [Online]. Disponível em: http://www.mozenda.com/. [Acedido: 30-Set-2014].
[45] E. R. Martins, «GEOGRAFIA E ONTOLOGIA: O FUNDAMENTO GEOGRÁFICO DO SER», *GEOUSP Espaço E Tempo*, vol. 0, n. 21, pp. 33–51, Jul. 2011.
[46] D. J. Canestraro Josimara, «Sistema de Apoio à Decisão Baseado em Ontologias para Unidades de Dor Torácica», *Rev. Eletrônica Sist. Informação*, 2006.
[47] N. F. Noy e D. L. Mcguinness, «Ontology Development 101: A Guide to Creating Your First Ontology», 2001.
[48] «DataGramaZero - Revista de Ciência da Informação - Artigo 04». [Online]. Disponível em: http://www.dgz.org.br/out09/Art_05.htm. [Acedido: 30-Set-2014].
[49] E. Morais e A. P. Ambrósio, «Ontologias: conceitos,usos,tipos,metodologias,ferramentas e linguagens».
[50] «protégé». [Online]. Disponível em: http://protege.stanford.edu/. [Acedido: 30-Set-2014].
[51] A. Maedche, E. Maedche, S. Staab, N. Stojanovic, Y. Sure, e R. Studer, «SEmantic portAL - The SEAL approach», em *Spinning the Semantic Web*, 2001, pp. 317–359.
[52] «WebODE». [Online]. Disponível em: http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/old-technologies/60-webode. [Acedido: 25-Nov-2014].
[53] «What is Data Mapping?», *wiseGEEK*. [Online]. Disponível em: http://www.wisegeek.com/what-is-data-mapping.htm. [Acedido: 30-Set-2014].
[54] «MAFRA Toolkit». [Online]. Disponível em: http://mafra-toolkit.sourceforge.net/. [Acedido: 25-Nov-2014].
[55] «Altova MapForce – Graphical Data Mapping, Conversion, and Integration Tool». [Online]. Disponível em: http://www.altova.com/mapforce.html. [Acedido: 30-Set-2014].
[56] A. B. Martins, E. Rodrigues, e M. Nunes, «Repositórios de informação e ambientes de aprendizagem»: [Online]. Disponível em: http://www.rbe.min-edu.pt/news/newsletter3/repositorios.pdf. [Acedido: 30-Set-2014].
[57] «Repositório», *Wikipédia, a enciclopédia livre*. 22-Ago-2014.
[58] K. Rohloff, M. Dean, I. Emmons, D. Ryder, e J. Sumner, «An Evaluation of Triple-Store Technologies for Large Data Stores», em *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, R. Meersman, Z. Tari, e P. Herrero, Eds. Springer Berlin Heidelberg, 2007, pp. 1105–1114.
[59] «Apache Jena - Home». [Online]. Disponível em: https://jena.apache.org/. [Acedido: 21-Out-2014].

[60] «Jena (framework)», *Wikipedia, the free encyclopedia*. 07-Out-2014.
[61] «Ontotext GraphDB - Ontotext». [Online]. Disponível em: http://www.ontotext.com/products/ontotext-graphdb-owlim/. [Acedido: 21-Out-2014].
[62] «Home - OWLIM54 - Ontotext Wiki». [Online]. Disponível em: http://owlim.ontotext.com/display/OWLIMv54/Home. [Acedido: 21-Out-2014].
[63] Admin, «SparkleDB», 13-Out-2014. [Online]. Disponível em: https://www.sparkledb.net/. [Acedido: 21-Out-2014].
[64] «SparkleDB», *Wikipedia, the free encyclopedia*. 18-Out-2014.
[65] «Continente - o seu hipermercado online.» [Online]. Disponível em: http://www.continente.pt/pt-pt/public/Pages/homepage.aspx. [Acedido: 11-Jan-2014].
[66] «.: JUMBO - Online :.» [Online]. Disponível em: http://www.jumbo.pt/Frontoffice/ContentPages/JumboNetWelcome.aspx. [Acedido: 11-Jan-2014].
[67] «OWL - Semantic Web Standards». [Online]. Disponível em: http://www.w3.org/2001/sw/wiki/OWL. [Acedido: 14-Out-2014].

7 Anexos

7.1 Anexo 1 – Definição da Ontologia

```
<?xml version="1.0"?>
<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.semanticweb.org/elisa/ontologies/2014/6/untitled-
ontology-23"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"

ontologyIRI="http://www.semanticweb.org/elisa/ontologies/2014/6/untitled-
ontology-23">
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Declaration>
        <Class IRI="#Categoria"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Loja"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Produto"/>
    </Declaration>
    <Declaration>
        <Class IRI="#ProdutoLoja"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#temCategoria"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#temLoja"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#temProduto"/>
    </Declaration>
    <Declaration>
        <DataProperty IRI="#temPreco"/>
    </Declaration>
    <SubClassOf>
        <Class IRI="#ProdutoLoja"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#temCategoria"/>
            <Class IRI="#Categoria"/>
```

```
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ProdutoLoja"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#temLoja"/>
            <Class IRI="#Loja"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ProdutoLoja"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#temProduto"/>
            <Class IRI="#Produto"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
</Ontology>
```

7.2 Anexo 2 – XSLT do Mapeamento da Ontologia

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/xpath-functions" exclude-result-
prefixes="xs fn">
 <xsl:output method="xml" encoding="UTF-8" byte-order-mark="no"
indent="yes"/>
 <xsl:template match="/">
       <xsl:variable name="var1_Allitems" as="node()?" select="Allitems"/>
       <RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#">
              <xsl:attribute name="xsi:schemaLocation"
namespace="http://www.w3.org/2001/XMLSchema-instance"
select="'http://www.w3.org/1999/02/22-rdf-syntax-ns#
C:/Users/Elisa/Desktop/BACKUP~2/TESTES~1/TESTEF~1/XMLFIN~1.XSD'"/>
              <xsl:for-each select="$var1_Allitems/*:Item[fn:namespace-
uri() eq '']">
                     <owl:Categoria>
                           <xsl:attribute name="about"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="fn:string(*:Categoria[fn:namespace-uri() eq ''])"/>
                     </owl:Categoria>
              </xsl:for-each>
              <xsl:for-each select="$var1_Allitems/*:Item[fn:namespace-
uri() eq '']">
                     <owl:Loja>
                           <xsl:attribute name="about"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="fn:string(*:Loja[fn:namespace-uri() eq ''])"/>
                     </owl:Loja>
              </xsl:for-each>
              <xsl:for-each select="$var1_Allitems/*:Item[fn:namespace-
uri() eq '']">
                     <owl:Produto>
                           <xsl:attribute name="about"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#" select="generate-
id(.)"/>
                           <owl:Descricao>
                                 <xsl:sequence
select="fn:string(*:Descricao[fn:namespace-uri() eq ''])"/>
                           </owl:Descricao>
                           <owl:Marca>
                                 <xsl:sequence
select="fn:string(*:Marca[fn:namespace-uri() eq ''])"/>
                           </owl:Marca>
                     </owl:Produto>
              </xsl:for-each>
              <xsl:for-each select="$var1_Allitems/*:Item[fn:namespace-
uri() eq '']">
                     <xsl:variable name="var2_resultof_generate_id"
as="xs:string" select="generate-id(.)"/>
                     <owl:ProdutoLoja>
                           <xsl:attribute name="about"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="fn:concat(fn:node-name(.), $var2_resultof_generate_id)"/>
                           <owl:Quantidade>
```

```
                                    <xsl:sequence
select="fn:string(*:Quantidade[fn:namespace-uri() eq ''])"/>
                                </owl:Quantidade>
                                <owl:Preco>
                                    <xsl:sequence
select="fn:string(*:Preco[fn:namespace-uri() eq ''])"/>
                                </owl:Preco>
                                <owl:PrecoUni>
                                    <xsl:sequence
select="fn:string(*:PrecoUni[fn:namespace-uri() eq ''])"/>
                                </owl:PrecoUni>
                                <owl:temLoja>
                                    <xsl:attribute name="resource"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="fn:string(*:Loja[fn:namespace-uri() eq ''])"/>
                                </owl:temLoja>
                                <owl:temCategoria>
                                    <xsl:attribute name="resource"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="fn:string(*:Categoria[fn:namespace-uri() eq ''])"/>
                                </owl:temCategoria>
                                <owl:temProduto>
                                    <xsl:attribute name="resource"
namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
select="$var2_resultof_generate_id"/>
                                </owl:temProduto>
                            </owl:ProdutoLoja>
                        </xsl:for-each>
                </RDF>
        </xsl:template>
</xsl:stylesheet>
```

7.3 Anexo 3 - XML Final do Mapeamento da Ontologia

```
<?xml version="1.0" encoding="UTF-8"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <owl:Categoria xmlns:ns0="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns0:about="Animais"/>
 <owl:Categoria xmlns:ns1="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns1:about="Animais"/>
 <owl:Loja xmlns:ns2="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns2:about="Continente"/>
 <owl:Loja xmlns:ns3="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns3:about="Jumbo"/>
 <owl:Produto xmlns:ns4="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns4:about="idelem4x3">
        <owl:Descricao>Alimento Seco para Cão</owl:Descricao>
        <owl:Marca>Continente</owl:Marca>
 </owl:Produto>
 <owl:Produto xmlns:ns5="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns5:about="idelem4x27">
        <owl:Descricao>Alimento Seco para Cão</owl:Descricao>
        <owl:Marca>Continente</owl:Marca>
 </owl:Produto>
 <owl:ProdutoLoja xmlns:ns6="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns6:about="Itemidelem4x3">
        <owl:Quantidade>emb. 20 kg</owl:Quantidade>
        <owl:Preco> 27,99 /un</owl:Preco>
        <owl:PrecoUni> 1,40 /kg</owl:PrecoUni>
        <owl:temLoja xmlns:ns7="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" ns7:resource="Continente"/>
        <owl:temCategoria xmlns:ns8="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" ns8:resource="Animais"/>
        <owl:temProduto xmlns:ns9="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" ns9:resource="idelem4x3"/>
 </owl:ProdutoLoja>
 <owl:ProdutoLoja xmlns:ns10="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
ns10:about="Itemidelem4x27">
        <owl:Quantidade>emb. 20 kg</owl:Quantidade>
        <owl:Preco> 27,99 /un</owl:Preco>
        <owl:PrecoUni> 1,40 /kg</owl:PrecoUni>
        <owl:temLoja xmlns:ns11="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" ns11:resource="Jumbo"/>
        <owl:temCategoria xmlns:ns12="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" ns12:resource="Animais"/>
        <owl:temProduto xmlns:ns13="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" ns13:resource="idelem4x27"/>
 </owl:ProdutoLoja>
</RDF>
```