



Dissertação

Mestrado em Engenharia Informática – Computação Móvel

***Integração semântica para identificação de recursos
em ambientes distribuídos heterogéneos***

Miguel Sousa Pragosa

Leiria, Setembro de 2012



Dissertação

Mestrado em Engenharia Informática – Computação Móvel

***Integração semântica para identificação de recursos
em ambientes distribuídos heterogéneos***

Miguel Sousa Pragosa

Dissertação de Mestrado realizada sob a orientação do Doutor Vitor Manuel Basto Fernandes,
Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, Setembro de 2012

Agradecimentos

Deixo o meu principal agradecimento Professor Doutor Vitor Manuel Basto Fernandes, orientador desta dissertação. Acima de tudo agradeço a sua inteira disponibilidade para acompanhar este projeto, mas também o espírito crítico que demonstrou, e ainda o interesse que manifestou pelo tema, o qual se revelou fundamental à evolução da dissertação.

A todos os colegas que me acompanharam durante este mestrado.

À minha família, e em especial aos meus pais, Carlos Pragosa e Judite Pragosa que, com muito carinho, sempre me incentivaram nesta caminhada, apoiando-me incondicionalmente mesmo nos momentos de maior ausência da minha parte.

Aos meus avós maternos e paternos, pela força que sempre me deram para conseguir atingir mais um exigente objetivo ao qual me propus.

À Tatiana, pela amizade e paciência que sempre demonstrou, mas também pela tão preciosa motivação que foi capaz de incutir em mim durante este último ano lectivo.

Resumo

Este documento apresenta o desenvolvimento de uma aplicação informática capaz de tornar possível a integração semântica entre sistemas heterogêneos e que não partilhem de qualquer modelo previamente definido para a identificação de recursos (ou produtos).

O projeto desenvolvido baseia-se na integração de duas bases de dados distintas, tornando possível a elaboração de um registo central e uniforme dos dados tratados. Este poderá ser utilizado para o desenvolvimento de vários tipos de aplicações enquadradas na área dos sistemas de análise de comportamentos de consumo alimentar segundo uma perspetiva de saúde pública.

O caso de uso demonstrado foca em particular a integração de sistemas de venda alimentar (bares, cantinas, etc.).

Para atingirmos os objetivos aos quais nos propomos, partimos para uma abordagem baseada em várias etapas, sendo que cada uma delas nos permitirá focar num subconjunto de elementos.

Em primeiro lugar aplicaremos técnicas de tratamento léxico e gramatical, as quais nos permitirão determinar relações entre os produtos em causa. Para além disto serão ainda aplicadas técnicas de integração baseadas em ontologias, as quais nos permitem inferir relações mais complexas entre os elementos de ambos os *datasets*.

Por fim será ainda apresentada a fase de calibração do mecanismo implementado, o qual se baseia em técnicas de otimização multiobjetivo.

Palavras-chave: Web semântica, Integração de sistemas, Partilha de informação

Abstract

This document presents the development of a software application which allows the semantic integration between heterogeneous systems without any preshared model for the resources' identification.

This project basis on the integration of two different databases, allowing us to create a central repository with all the information processed. This data may enable the scientific community to develop several applications on the public health care analysis field.

We will show a practical use case that focus on the integration of food-vending system databases.

In order to achieve our targets we decided to adopt a step-by-step approach.

First of all we will apply some lexical and grammatical techniques, which will help us deducing some relations between the items being compared. After that we will also apply some ontology-based integration techniques. Those techniques will let us infer more complex relations between both datasets.

Finally we will present the calibration phase, where we will tune the comparison engine in order to achieve the best possible results for the considered datasets.

Key-Words: Semantic web, System integration, Data sharing

Índice de Figuras

FIGURA 1. DIAGRAMA DE ALTO NÍVEL DA SOLUÇÃO PROPOSTA	27
FIGURA 2. DIAGRAMA DE CLASSES: PACKAGE “ENTITIES” – PARTE 1	29
FIGURA 3. DIAGRAMA DE CLASSES: PACKAGE “ENTITIES” – PARTE 2	30
FIGURA 4. DIAGRAMA DE CLASSES: PACKAGE “ENTITIES” – PARTE 3	31
FIGURA 5. DIAGRAMA DE CLASSES: PACKAGE “PARSERS” – PARTE 1	32
FIGURA 6. DIAGRAMA DE CLASSES: PACKAGE “PARSERS” – PARTE 2	33
FIGURA 7. DIAGRAMA DE CLASSES: PACKAGE “PROCESS” – PARTE 1	34
FIGURA 8. DIAGRAMA DE CLASSES: PACKAGE “PROCESS” – PARTE 2	35
FIGURA 9. DIAGRAMA DE CLASSES: PACKAGE “PROVIDERS” – PARTE 1	36
FIGURA 10. DIAGRAMA DE CLASSES: PACKAGE “PROVIDERS” – PARTE 2	37
FIGURA 11. SINTAXE DE UM ELEMENTO DE UM FICHEIRO DE REGRAS.....	41
FIGURA 12. EXEMPLO DA DIVISÃO DE UM TERMO EM TOKENS.....	43
FIGURA 13. SINTAXE DE UM ELEMENTO DE UM DICIONÁRIO	44
FIGURA 14. SINTAXE DA TAG DE ABERTURA DE UM ELEMENTO “RULESET” (UTILIZADO NA DEFINIÇÃO DE DICIONÁRIOS)	46
FIGURA 15. EXEMPLO DA CLASSIFICAÇÃO GRAMATICAL DE UM TERMO	47
FIGURA 16. EXEMPLO DA CLASSIFICAÇÃO GRAMATICAL ATRAVÉS DE REGRAS RELACIONAIS.....	48
FIGURA 17. SINTAXE A RESPEITAR PARA A DEFINIÇÃO DE PARÂMETROS DINÂMICOS	49
FIGURA 18. SINTAXE DE UM ELEMENTO DO DICIONÁRIO DE GRAMÁTICA RELACIONAL.....	50
FIGURA 19. CAMADAS DA ARQUITETURA DE SISTEMAS BASEADOS EM WEB SEMÂNTICA[16]	51
FIGURA 20. SINTAXE DE UM ELEMENTO DE UM DICIONÁRIO (COM FOCO NA INFORMAÇÃO DE ONTOLOGIA)	53
FIGURA 21. ORGANIZAÇÃO DE CLASSES DA ONTOLOGIA GERADA (REPRESENTAÇÃO GRÁFICA PELA APLICAÇÃO PROTÉGÉ)	54
FIGURA 22. ORGANIZAÇÃO DE CLASSES DA ONTOLOGIA GERADA (REPRESENTAÇÃO GRÁFICA PELA APLICAÇÃO PROTÉGÉ)	55
FIGURA 23. ALGORITMO DE PROCESSAMENTO SEMÂNTICO: VISÃO GERAL.....	57
FIGURA 24. ALGORITMO DE PROCESSAMENTO SEMÂNTICO: COMPARAÇÃO ENTRE TOKENS	58
FIGURA 25. ALGORITMO DE PROCESSAMENTO SEMÂNTICO: COMPARAÇÃO SIMPLES ENTRE TOKENS	59
FIGURA 26. ALGORITMO DE PROCESSAMENTO SEMÂNTICO: COMPARAÇÃO DE TOKENS COM BASE NA ONTOLOGIA	60

FIGURA 27. FÓRMULA DE CÁLCULO PARA A OBTENÇÃO DO RESULTADO ABSOLUTO DE UMA COMPARAÇÃO	62
FIGURA 28. DIAGRAMA DE SEQUÊNCIA: COMUNICAÇÃO ENTRE MOTOR DE OTIMIZAÇÃO E APLICAÇÃO DESENVOLVIDA	70
FIGURA 29. RESULTADOS DA EXECUÇÃO DO MOTOR DE OTIMIZAÇÃO MULTIOBJETIVO: VISTA GERAL	71
FIGURA 30. RESULTADOS DA EXECUÇÃO DO MOTOR DE OTIMIZAÇÃO MULTIOBJETIVO: VISTA DETALHADA SOBRE OS 10 MELHORES RESULTADOS OBTIDOS.....	72

Índice de Tabelas

TABELA I. EXEMPLOS DA UTILIZAÇÃO DE PARÂMETROS DINÂMICOS PARA GRAMÁTICA RELACIONAL.....	50
TABELA II. EXEMPLO DA DECOMPOSIÇÃO DE PALAVRAS NOS SEUS RADICAIS	59
TABELA III. EXEMPLO PRÁTICO SOBRE A AVALIAÇÃO DE RESULTADOS DA APLICAÇÃO	68
TABELA IV. VALORES OBTIDOS NO MELHORES RESULTADOS DE OTIMIZAÇÃO	72

Lista de Siglas

API	Application Programming Interface
BD	Base de dados
eBNF	Extended Backus-Naur Form
GPL	GNU Public License
JDBC	Java Database Connectivity
OWL	Web Ontology Language
POS	Point of Sale
RDF	Resource Description Framework
SGBD	Sistema de gestão de bases de dados
SQL	Structured Query Language
UML	Unified Modeling Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSD	XML Schema Definition

Índice

DEDICATÓRIA.....	I
AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABELAS	IX
LISTA DE SIGLAS	XI
ÍNDICE	XIII
INTRODUÇÃO.....	15
1.1 CASO DE ESTUDO DEMONSTRADO	15
1.2 MOTIVAÇÃO E OBJETIVOS DO PROJETO.....	16
1.3 METODOLOGIA DE INVESTIGAÇÃO	17
1.4 ESTRUTURA DO DOCUMENTO	17
REVISÃO DA LITERATURA	19
2.1 MOTORES DE BUSCA	19
2.2 APLICAÇÕES EMPRESARIAIS	20
2.3 REPOSITÓRIOS DE INFORMAÇÃO.....	20
2.4 BASES DE DADOS ALIMENTARES.....	20
CONSIDERAÇÕES INICIAIS.....	23
3.1 LINGUAGEM DE PROGRAMAÇÃO ADOTADA.....	23
3.1.1 FRAMEWORKS DE DESENVOLVIMENTO.....	23
3.2 SISTEMA DE GESTÃO DE BASES DE DADOS ADOTADO	24
3.3 TERMINOLOGIA ADOTADA.....	25
ESTRUTURA DA SOLUÇÃO	27
4.1 DIAGRAMAS DE CLASSES	28
ESTUDO E ADAPTAÇÃO DOS RECURSOS DISPONÍVEIS	39
5.1 CARACTERIZAÇÃO DOS DATASETS	39
5.2 ANÁLISE DOS DATASETS	39
5.2.1 REMOÇÃO E CORREÇÃO MANUAL DE TERMOS.....	40
5.2.2 CORREÇÃO E NORMALIZAÇÃO AUTOMÁTICA DE TERMOS	41

EXTRAÇÃO DE ATRIBUTOS	43
6.1 ANÁLISE LEXICAL	43
6.1.1 IMPLEMENTAÇÃO	44
6.2 ANÁLISE GRAMATICAL.....	46
6.2.1 IMPLEMENTAÇÃO	48
6.2.2 SINTAXE DINÂMICA PARA REGRAS DE GRAMÁTICA RELACIONAL.....	49
6.3 CONSTRUÇÃO DE ONTOLOGIAS	50
6.3.1 TECNOLOGIAS UTILIZADAS	51
6.3.2 IMPLEMENTAÇÃO	52
MOTOR DE PROCESSAMENTO SEMÂNTICO.....	57
7.1 ALGORITMO DE PROCESSAMENTO	57
7.2 MECANISMO DE PONTUAÇÃO	61
7.2.1 IMPLEMENTAÇÃO	62
A) EXEMPLO 1.....	62
A) EXEMPLO 2.....	63
EXPERIMENTAÇÃO	67
8.1 OTIMIZAÇÃO DE RESULTADOS.....	68
CONCLUSÕES.....	73
9.1 CUMPRIMENTO DOS OBJETIVOS DEFINIDOS	73
9.2 PRINCIPAIS LIMITAÇÕES	74
9.3 CONTRIBUIÇÃO	74
9.4 TRABALHO FUTURO.....	75
BIBLIOGRAFIA	77
ANEXOS	81

Introdução

Com o expandir das tecnologias da informação a (quase) todas as áreas de atividade humana, torna-se muitas vezes fundamental a integração de sistemas heterogêneos não preparados ou desenhados para interoperar entre si.

Adicionalmente, é também importante notar que a adoção de normas para troca de informação entre sistemas heterogêneos, ao nível da identificação de recursos, modelos e sintaxes de dados ou processos de negócio, é um cenário bastante improvável. Posto isto, apenas cenários altamente controlados em cadeias de valor especializadas, revelam alinhamentos dos sistemas de informação das entidades em interação.

Deste modo, a aplicação de técnicas de integração semântica de sistemas torna-se muitas vezes imprescindível para a integração de processos de negócio compostos por elementos não desenhados para comunicar entre si.

1.1 Caso de estudo demonstrado

Por forma a demonstrar o conceito apresentado, este documento evidenciará a integração de bases de dados (BD) de sistemas de venda alimentar – *Points of Sale* (POS).

São inúmeros os fabricantes de equipamentos e *software* do tipo POS (v.g. WinRest, WinTouch, etc.), no entanto a partilha e adoção de normas únicas de identificação dos produtos por parte destes é extremamente improvável. Assim, teremos como principal objetivo deste trabalho descobrir a correta correspondência entre um conjunto de produtos conhecidos por um sistema de POS, e um catálogo de produtos alojados num repositório central remoto.

Pretende-se em concreto implementar um mecanismo que permita, a partir das descrições textuais em Português de produtos existentes em ambas as bases de dados, estabelecer a correspondência entre os produtos semelhantes existentes nessas mesmas bases de dados.

Em termos práticos, este processo focar-se-á na disponibilização de um registo uniforme de dados de consumo alimentar recolhidos a partir de múltiplos POS, tornando assim viável a implementação de aplicações destinadas à análise dos perfis de consumo de comunidades de interesse (podendo tomar como base o tipo de alimentos consumidos, os valores químicos e nutricionais dos mesmos, etc.) [1].

1.2 Motivação e objetivos do projeto

São muitas as áreas de negócio que carecem de mecanismos de integração de sistemas tal como aquele que é apresentado ao longo deste documento. Deste modo, este é um projeto com uma aplicação prática bastante alargada.

Por forma a potenciar a adoção do projeto desenvolvido por parte destas áreas, este documento focará com especial atenção os três tópicos seguidamente enunciados:

- **Adaptabilidade:** Dependendo da área de negócio a que se aplique, os *datasets* manipulados pela aplicação poderão sofrer constantes mutações. Dado isto, o sistema desenvolvido deve estar preparado para expandir a sua abrangência a estes novos elementos. A aplicação deve ainda adaptar-se a *datasets* de variadas dimensões;
- **Manutenção:** Muitas das aplicações desenvolvidas no meio científico-académico, acabam por revelar-se inúteis aquando da sua aplicação prática, ou seja, fora de um ambiente fechado. Deste modo, será dada grande importância à forma como a manutenção do sistema é feita, tentando tornar este processo o mais simples possível, por forma a poder ser executado por pessoas sem qualquer formação técnica especializada;
- **Precisão:** A aplicação do sistema desenvolvido a um cenário prático não fechado, jamais será possível se a mesma não garantir um grau de confiabilidade mínimo aceitável. É assim natural que um dos tópicos que mais esforços mova durante o desenvolvimento desta aplicação seja a implementação de um mecanismo de *matching* com elevado grau de confiança na pesquisa de produtos semelhantes em bases de dados diferentes.

Importa também destacar que com este projeto pretende-se contribuir para uma evolução

nas áreas de investigação da integração de sistemas.

1.3 Metodologia de investigação

Dado que o problema que se pretende resolver remete diretamente para as tecnologias de tratamento de texto não anotado e tecnologias de modelação semântica, foi seguida uma abordagem de investigação típica desta área de conhecimento.

Ou seja, foi realizada inicialmente a recolha de dados de exemplo, representativos para o problema da integração de sistemas na área dos sistemas de venda de produtos alimentares. Seguiu-se a fase de limpeza e normalização dos dados (remoção de erros ortográficos, correção de problemas de formatação e *encoding*, etc.) e a modelação lexical do domínio do problema (especificação do vocabulário e suas derivações). De seguida procedeu-se à definição das estruturas gramaticais aplicáveis à descrição de produtos, à modelação da base de conhecimento (ontologia) representativa do domínio alimentar, e por fim à especificação, implementação e calibração dos mecanismos de *matching*.

1.4 Estrutura do documento

No capítulo dois é feita uma abordagem aos trabalhos que contribuíram direta ou indiretamente para a realização deste projeto.

O capítulo três é dedicado à apresentação das ferramentas usadas no trabalho e a algumas considerações iniciais necessárias para o enquadramento do trabalho desenvolvido.

O quarto capítulo servirá para expor a estrutura da aplicação. Isto será feito com recurso a diversos tipos de diagramas, representando os principais componentes da aplicação.

No quinto capítulo serão focados diversos aspetos em torno dos dados utilizados como base para a integração de sistemas. Estes serão analisados, caracterizados e adaptados às características e às necessidades do sistema proposto.

De seguida será exposto o funcionamento do mecanismo de extração de atributos. Este tem à sua responsabilidade a análise lexical e gramatical dos *datasets* e ainda a construção de ontologias adaptadas a estes.

O sétimo capítulo focar-se-á no mecanismo de processamento semântico, fazendo também referência ao motor de pontuação implementado. Após isto, será ainda abordada a fase de

experimentação, a qual incluirá a descrição do algoritmo de otimização de pesos implementado.

Por fim serão apresentadas as conclusões tendo em conta vários aspetos – o cumprimento dos objetivos propostos, as principais limitações da solução proposta, a contribuição da mesma em termos científicos e tecnológicos, e os tópicos abertos a desenvolvimentos futuros.

Revisão da literatura

O presente capítulo surge como resultado da pesquisa bibliográfica produzida. Pretende-se assim enquadrar os esforços e trabalhos de investigação desenvolvidos até à data na implementação de produtos cujos objetivos se assemelham aos do projeto apresentado.

Dadas as suas potencialidades, durante os últimos anos temos vindo a assistir a um aumento do número de ferramentas que utilizam técnicas de integração ou análise semântica. Estes conceitos representam parte do que se designa por Web 3.0 ou Web Semântica [2].

2.1 Motores de busca

Algumas das aplicações que mais têm vindo a usufruir da utilização de técnicas de integração semântica são sem dúvida os motores de busca. Através desta evolução espera-se que dentro de algum tempo os motores de busca sejam capazes de interpretar, não apenas palavras-chave conhecidas, mas também perguntas complexas em formato “*human readable*”.

Exemplos como o Knigine¹, o Hakia Search Engine² ou o DuckDuckGo³ tiram já partido das vantagens da integração semântica, garantindo assim um conjunto de resultados mais abrangente e mais inteligente para as suas pesquisas.

Apesar de não existir ainda qualquer informação oficial que o confirme, tudo indica para que o motor de busca Google se prepare também para integrar algumas destas funcionalidades. A fundamentar esta afirmação encontra-se o facto de a empresa americana ter adquirido, em Julho de 2010, a empresa Metaweb [3] – uma *startup* focada no desenvolvimento de uma base de dados semântica livre e verdadeiramente abrangente, à qual foi dado o nome de Freebase. Esta base de dados é de momento mantida pela americana Google®, mantendo-se livre e dispondo já de mais de 22 milhões de elementos.

¹ <http://kngine.com/>

² <http://www.hakia.com/>

³ <http://duckduckgo.com/>

2.2 Aplicações empresariais

Ao nível empresarial, surgem também já algumas implementações, destacando-se entre estas o conjunto de produtos oferecidos pela empresa TopQuadrant⁴. No mercado desde 2001, a empresa é atualmente uma das mais bem-sucedidas na área do desenvolvimento de aplicações empresariais com recurso a ferramentas semânticas. As aplicações por si disponibilizadas visam possibilitar uma melhor organização dos dados produzidos pela organização, permitindo assim que se economize tempo e dinheiro.

Também a gigante americana Oracle® disponibiliza a sua *suite* de aplicações semânticas [4]. Através da versão 11g da Oracle Database, é também disponibilizado um conjunto de tecnologias semânticas, as quais permitem atribuir um conjunto de significados à informação existente, estabelecer relações entre os vários elementos existentes na base de dados, ou executar ações de *data mining* com maior facilidade.

Esta *suite* de funcionalidades implementa ainda um conjunto de standards, permitindo desta forma a sua utilização por parte de aplicações terceiras.

2.3 Repositórios de informação

Também outros repositórios de informação online, tais como as plataformas Wiki poderão beneficiar em muito das características da Web semântica. Estas permitirão, por exemplo, um mapeamento mais inteligente entre os diversos artigos que compõem o repositório ou mesmo entre os seus elementos [5].

A extensão Semantic MediaWiki⁵ para a plataforma MediaWiki⁶ (na qual se baseia, por exemplo, a conhecida Wikipedia), é um exemplo bastante bem sucedido, sendo já implementado por dezenas de portais Wiki em todo o mundo.

2.4 Bases de dados alimentares

A empresa Google® toma, desde Janeiro de 2011, partido de ferramentas semânticas para a disponibilização de uma ferramenta de pesquisa de receitas – a Google Recipe View [6].

Atualmente disponível apenas nos Estados Unidos da América e no Japão, esta ferramenta

⁴ <http://topquadrant.com/>

⁵ <http://www.semantic-mediawiki.org/>

⁶ <http://www.wikipedia.org/>

online permite ao utilizador obter uma pesquisa bastante mais refinada, não tendo de se cingir à pesquisa de conjuntos textuais.

De entre as funcionalidades disponíveis, destaca-se o facto de ser possível pesquisar receitas que contenham um determinado ingrediente ou que pelo contrário, não devam conter um ingrediente não desejado. Este motor de busca permite ainda refinar a pesquisa, quer em termos do seu tempo de confecção, quer em termos da carga calórica do mesmo.

Uma alternativa a esta plataforma é o portal Yummly⁷. Este motor de busca vai ainda mais longe, permitindo ao utilizador definir o sabor que pretende obter (mais ou menos salgado, doce, picante, etc.), a origem cultural do alimento (cozinha indiana, mexicana, italiana, etc.), ou mesmo filtrar alimentos aos quais seja alérgico (identificando alimentos que contenham um determinado componente, com por exemplo, glúten ou sulfitos).

Por último, mas não menos importante, importa ainda fazer referência ao projeto Wordnet. Iniciado na Universidade de Princeton, nos Estados Unidos da América, este projeto visa disponibilizar uma base de dados léxico-conceitual da língua inglesa.

Oferecendo um conceito bastante semelhante ao das ontologias, o Wordnet permite estabelecer uma rede relacional entre os elementos de uma língua. De entre as relações que poderão ser estabelecidas contam-se as relações de equivalência, de hiperonímia, de similaridade, entre outras [7].

Tal como aconteceu já em muitos outros países, também a língua portuguesa acabou por ganhar a sua própria Wordnet. A aplicação Wordnet.PT dispõe atualmente de um total de mais de 19000 expressões repartidas por diversos domínios semânticos⁸. Esta implementação é da responsabilidade do Grupo de Computação do Conhecimento Léxico-Gramatical do Centro de Linguística da Universidade de Lisboa.

⁷ <http://www.yummly.com/>

⁸ <http://www.clul.ul.pt/clg/wordnetpt/>

Considerações iniciais

O início de um projeto de investigação implica obrigatoriamente a tomada de algumas decisões desde logo fundamentais à forma como se desenrolará todo o trabalho.

Tratando-se de um projeto que envolve fundamentalmente a programação de sistemas informáticos, as considerações apresentadas durante este capítulo prender-se-ão com questões de ordem técnica.

3.1 Linguagem de programação adotada

A primeira e também uma das mais importantes decisões tomadas foi sem dúvida a escolha da plataforma de programação a utilizar, na medida em que esta influencia diretamente todo o trabalho desenvolvido daí em diante.

Foram vários os parâmetros que pesaram na decisão final, devendo esta ser uma linguagem de alto nível, com a qual os membros da equipa de desenvolvimento estivessem já familiarizados, e que fosse acompanhada por uma vasta comunidade de programadores. Dados estes fatores, a nossa escolha recaiu sobre a plataforma Java, já que esta é também uma plataforma não-proprietária e livre de direitos de utilização.

3.1.1 Frameworks de desenvolvimento

Por forma a agilizar alguns processos da aplicação, foram utilizadas sobre a plataforma Java algumas *frameworks* úteis aos objetivos do desenvolvimento.

A primeira destas *frameworks*, Hibernate⁹, permite um maior grau de abstração do código perante o motor de base de dados utilizado para o armazenamento dos dados a integrar. Com esta *framework*, não só o acesso aos dados por parte do programador se torna mais simples, como também os parâmetros de acesso à base de dados se tornam configuráveis

⁹ <http://www.hibernate.org/>

através de ficheiros de texto independentes do código desenvolvido. O primeiro dos fatores acima apresentados permite libertar o programador de tarefas como a criação de *queries* do tipo Structured Query Language (SQL) [8], ou o tratamento manual dos resultados por si devolvidos.

A utilização de um ficheiro de configuração para acesso à BD, permite por sua vez que os parâmetros de acesso à BD sejam modificados sem que seja necessária qualquer alteração no código fonte da aplicação.

Para além disto, foi ainda utilizada a *framework* Log4J¹⁰, a qual oferece ao utilizador o controlo total sobre os *logs* de execução gerados pela aplicação. Tal como na solução anterior, esta configuração é possível através da manipulação de ficheiros de textos simples e independentes do código fonte da aplicação.

Ambas as *frameworks* utilizadas são disponibilizadas através de licenças de código aberto, respetivamente, propriedade das companhias Red Hat, Inc.® e Apache® Software Foundation.

3.2 Sistema de gestão de bases de dados adotado

Tal como já referido, o objetivo deste trabalho consiste na integração de duas fontes de dados distintas. A base de dados relativa ao sistema POS fora disponibilizada em formato de texto, enquanto que a base de dados central se encontrava no formato proprietário Microsoft Access.

Estes fatores, aliados à necessidade de um sistema cuja informação esteja livre de padrões proprietários, levaram à necessidade de centralizar toda a informação de modo a que esta esteja disponível através de um único repositório.

Alguns dos requisitos principais ditam ainda que a plataforma adotada seja suportada pela *framework* Hibernate (e pelo driver Java Database Connectivity - JDBC - por si utilizado), devendo garantir um desempenho razoável sem que sejam necessários grandes investimentos em termos de *hardware* (ao contrário do que acontece por exemplo com as BD *in-memory*). A estes requisitos junta-se ainda a preferência por Sistemas de Gestão de Bases de Dados (SGBD) cuja licença permita a sua livre utilização.

¹⁰ <http://logging.apache.org/log4j>

Dados todos estes fatores, a escolha recaiu sobre o SGBD MySQL [9], propriedade da americana Oracle Corporation® e disponível através de uma licença *open source* do tipo GNU Public License (GPL)¹¹.

Para além das vantagens já apresentadas, o SGBD adotado dispõem de implementações para um vasto leque de sistemas operativos, facilitando assim a disponibilização do serviço.

3.3 Terminologia adotada

Um dos pontos que merece principal foco nesta etapa é sem dúvida a nomenclatura atribuída a cada um dos artefactos integrantes deste projeto.

Ambas as bases de dados fornecidas foram nomeadas consoante o seu conteúdo e a sua finalidade, ou seja, o conjunto de dados referente aos produtos do sistema POS será referida como “BD de POS”, o conjunto de dados referente aos produtos do sistema central será referida como “BD central”, enquanto que, por fim, o conjunto de dados referente às características nutricionais dos produtos da tabela central será referida como “BD de valores”. Ao conjunto destas tabelas será dado o nome de “base de dados de produtos”.

Descendo um pouco no nível de abstração, o nome de cada um dos produtos de ambas as bases de dados será tratado como “termo”. Após a divisão de cada termo em segmentos lógicos de menor dimensão, poderemos obter uma lista daquilo a que chamaremos de “tokens”.

¹¹ <http://www.mysql.com/>

Estrutura da solução

Durante este capítulo será apresentada a solução desenvolvida, focando a esquematização gráfica da mesma. Em traços gerais a aplicação é composta por 6 módulos distintos mas interdependentes:

- Motor de importação de dados (“data import engine”);
- Motor de classificação através de dicionários (“dictionary classification engine”);
- Motor de classificação de gramática relacional (“relational grammar engine”);
- Motor de construção de ontologias (“ontology building engine”);
- Motor de processamento semântico (“processing engine”);
- Motor de otimização multiobjetivo (“optimization engine”).

Todo este fluxo de execução pode ser analisado através da Figura 1.

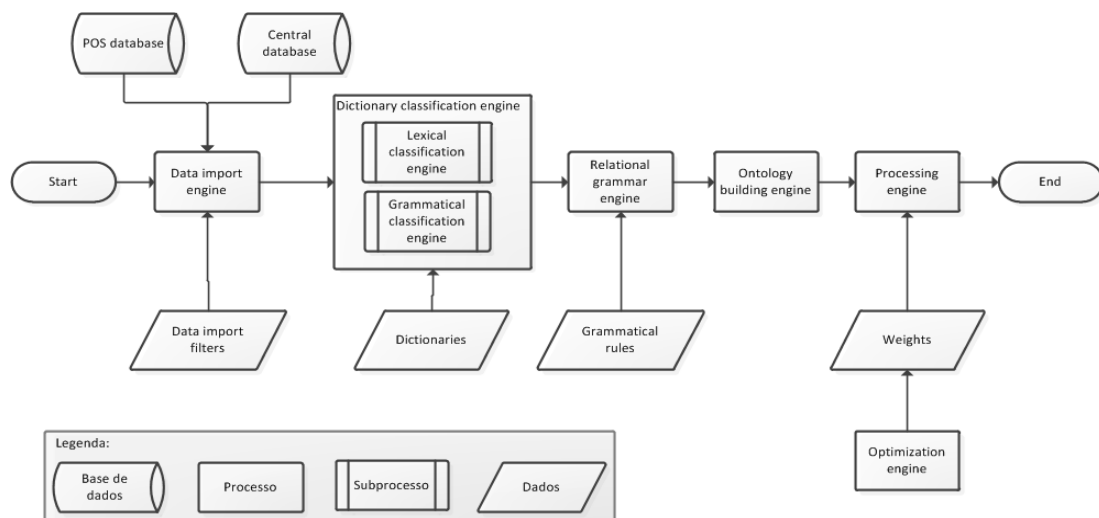


Figura 1. Diagrama de alto nível da solução proposta

No decorrer deste capítulo será exibida em detalhe a estrutura de alguns destes componentes. Para isto será utilizada a notação Unified Modeling Language (UML), um padrão para a modelação de *software* vocacionado para metodologias de desenvolvimento orientadas a objetos [10]. Ao longo deste capítulo focar-nos-emos nos principais diagramas de classes da aplicação. No decorrer do documento serão ainda apresentados os diagramas de atividade dos processos que julgamos serem mais relevantes.

Dada a complexidade da aplicação, esta não será apresentada como um todo, mas sim através da representação dos principais módulos que a constituem. Os restantes diagramas poderão ser consultados em anexo.

4.1 Diagramas de classes

Durante este capítulo serão apresentadas as principais classes do modelo proposto com base no projeto desenvolvido. Esta representação será feita tendo como base os *packages* Java utilizados.

Por forma a garantir uma boa organização do documento, o diagrama de classes global será incluído apenas em suporte digital.

As três primeiras figuras demonstram a estrutura do *package* “Entities”, o qual contém cada uma das entidades necessárias ao funcionamento do projeto (tais como os produtos do POS ou da BD central, ou os resultados produzidos pela aplicação).

As duas figuras que se seguem (*package* “Parsers”) demonstram a estrutura dos parsers utilizados pela aplicação para interpretar dados oriundos de ficheiros externos.

De seguida serão exibidas as classes relacionadas com o motor de processamento semântico (*package* “process”).

Por fim, surgem ainda representados os *providers* utilizados pela aplicação para obter dados a partir de recursos externos. Em alguns dos casos este acesso é feito por intermédio de uma das classes do *package* “Parsers”.

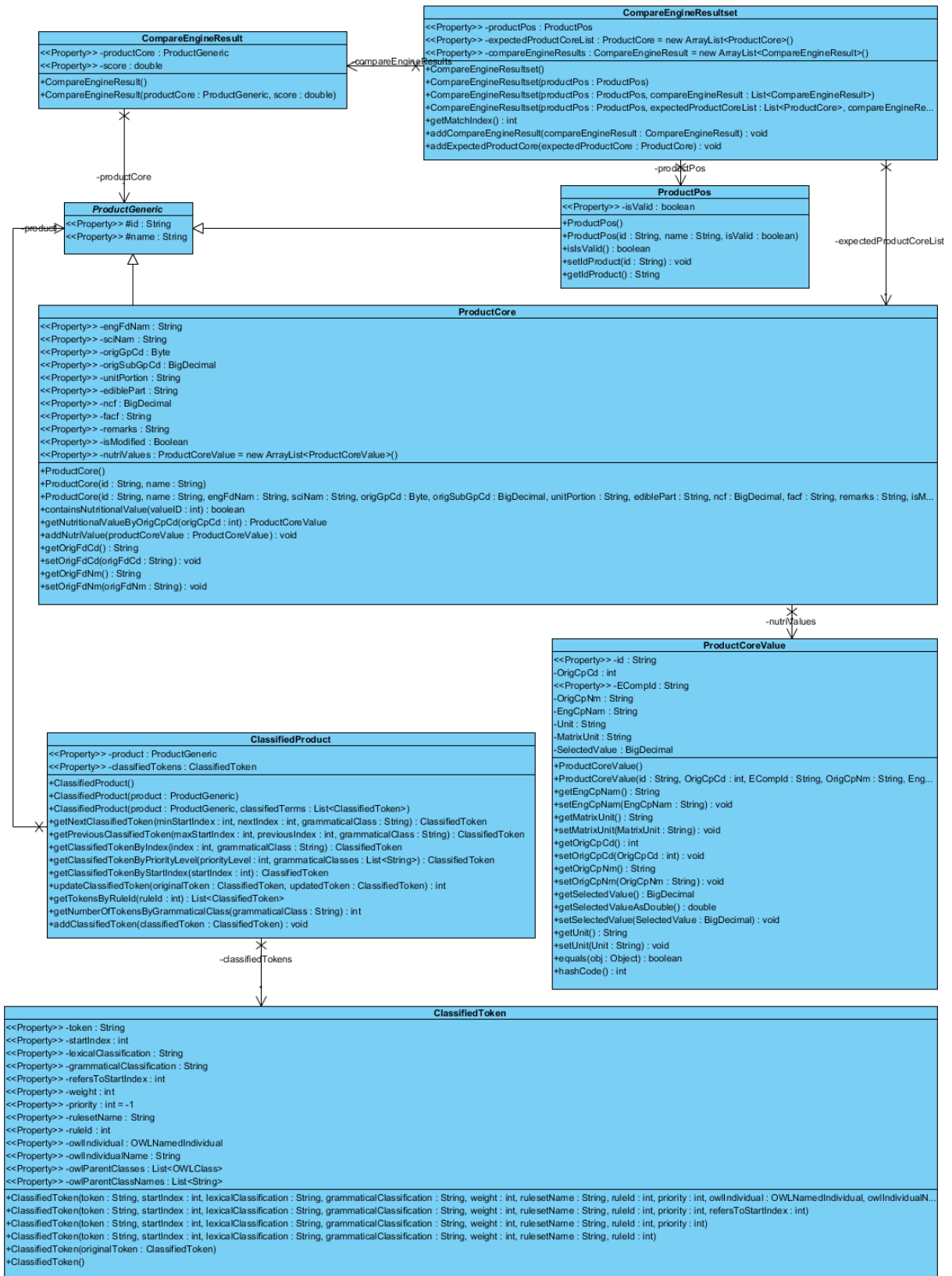


Figura 2. Diagrama de classes: Package “Entities” – Parte 1

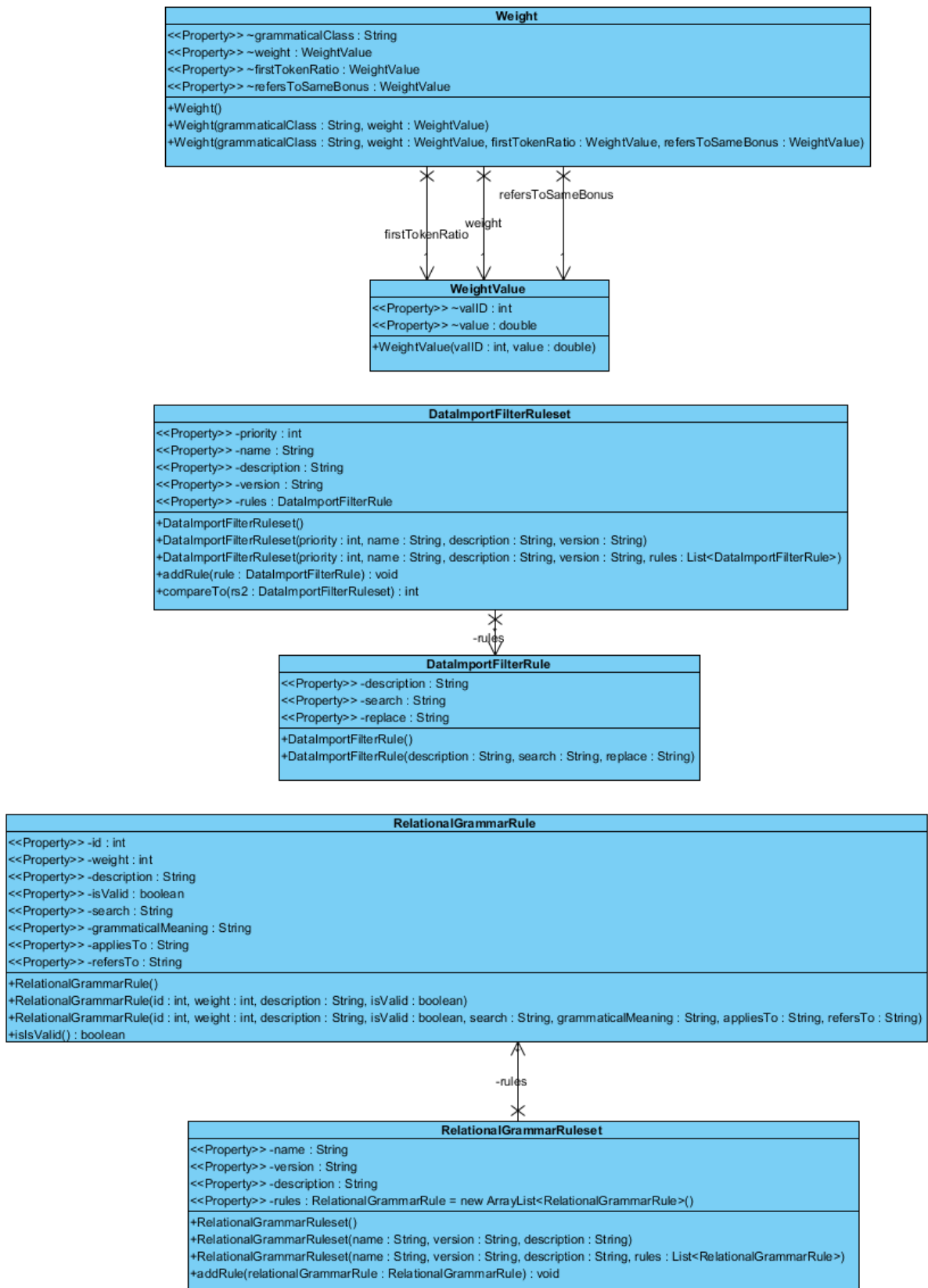


Figura 3. Diagrama de classes: Package “Entities” – Parte 2

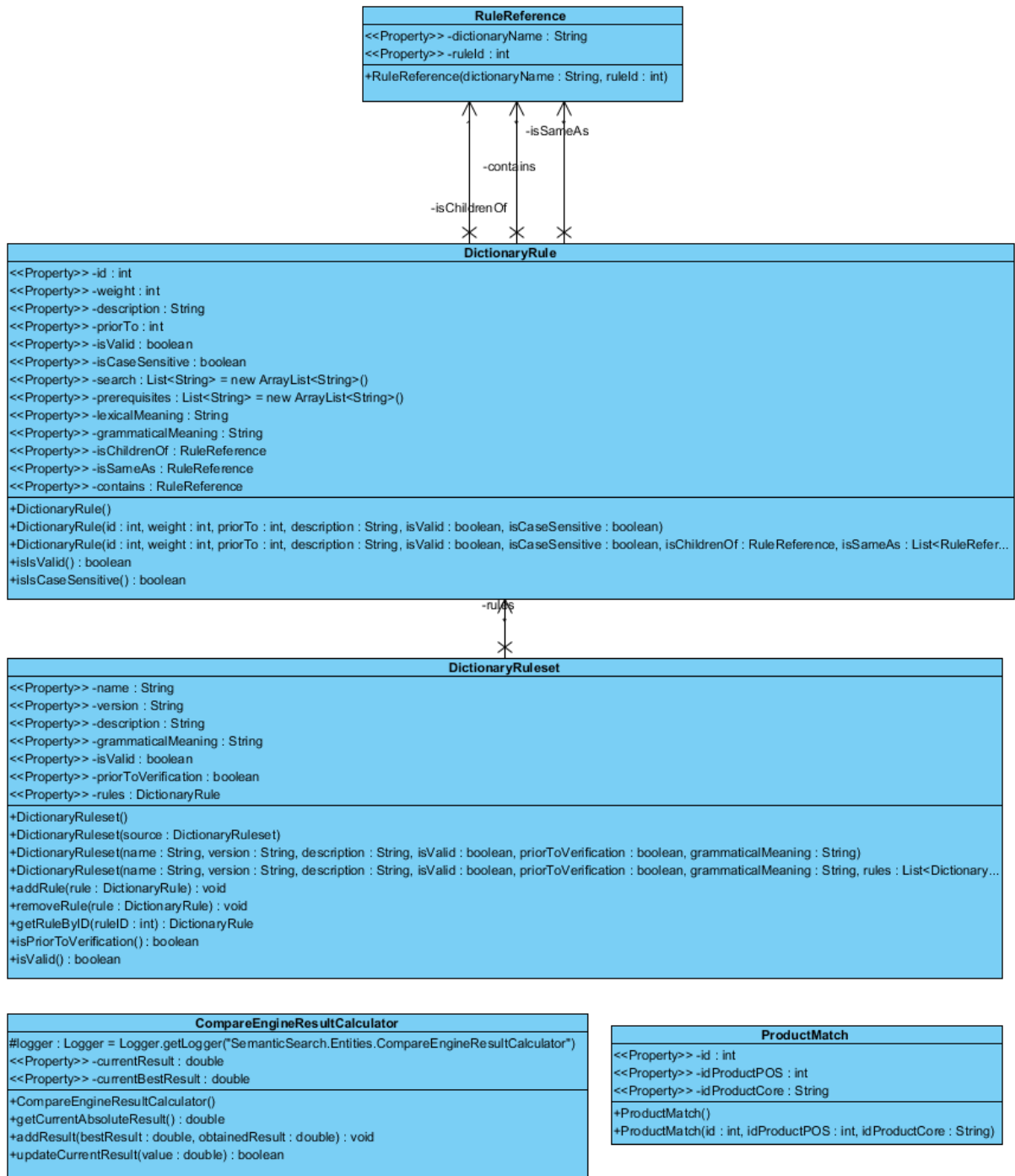


Figura 4. Diagrama de classes: Package “Entities” – Parte 3

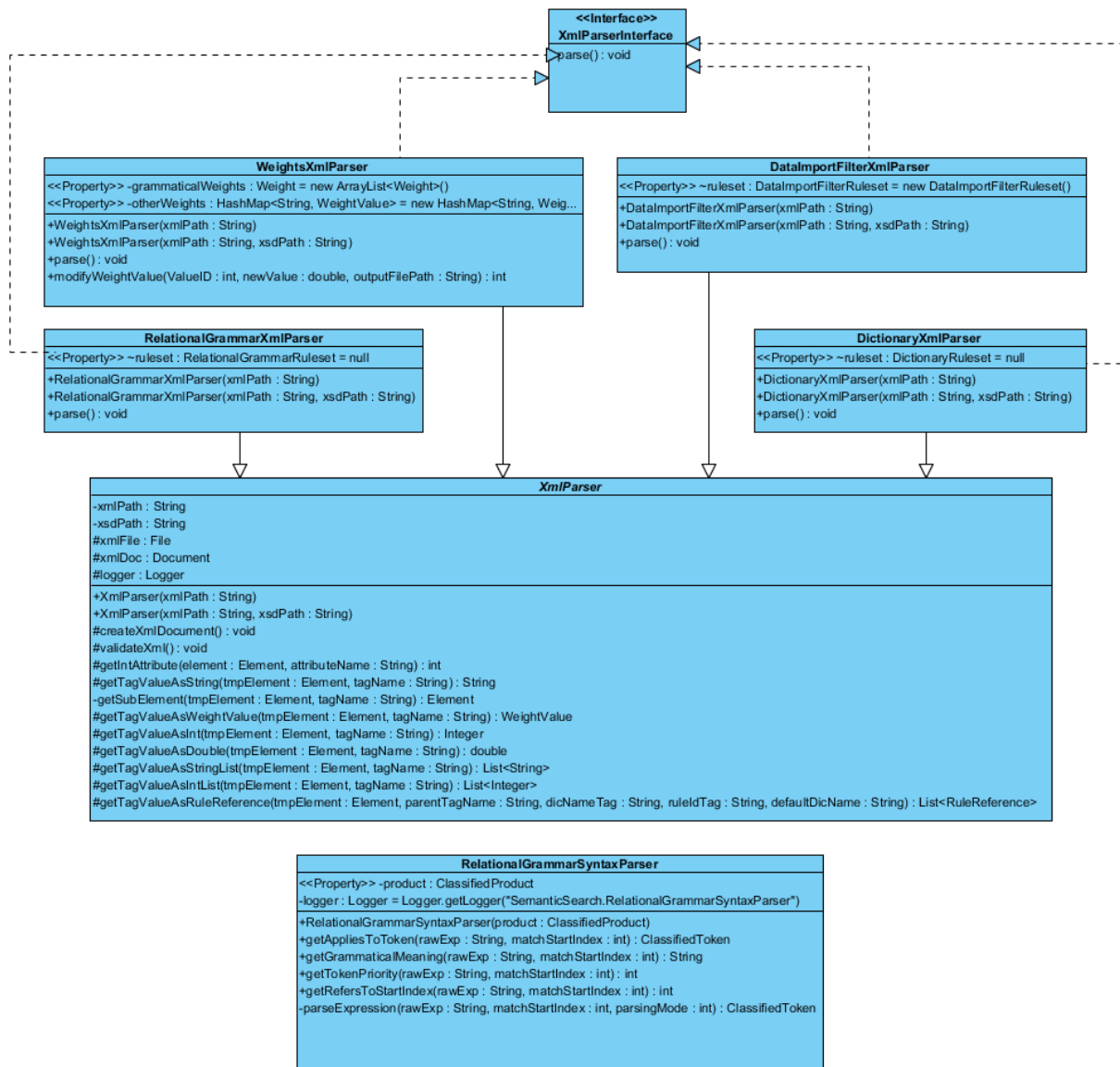


Figura 5. Diagrama de classes: Package “Parsers” – Parte 1

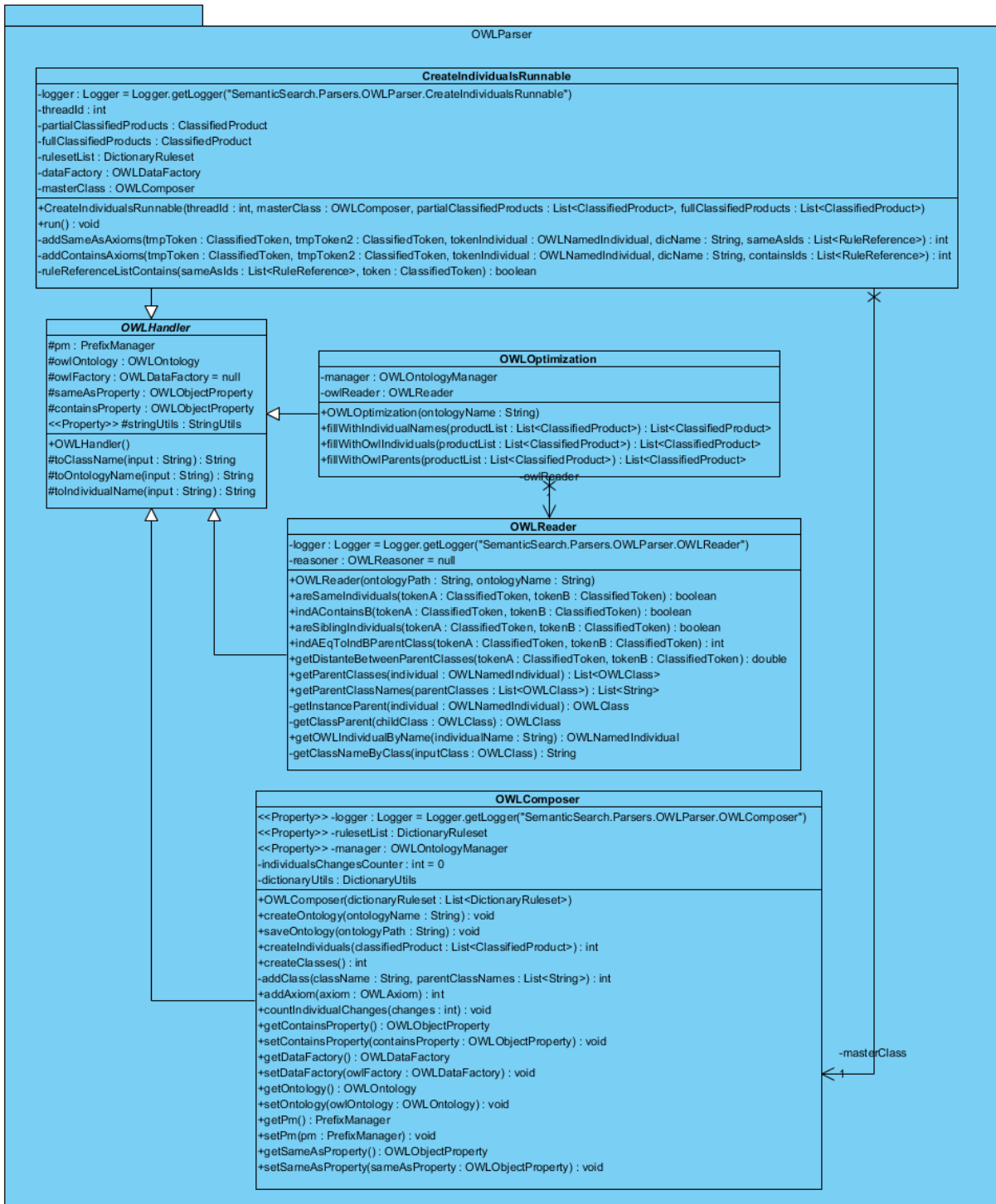


Figura 6. Diagrama de classes: Package "Parsers" – Parte 2

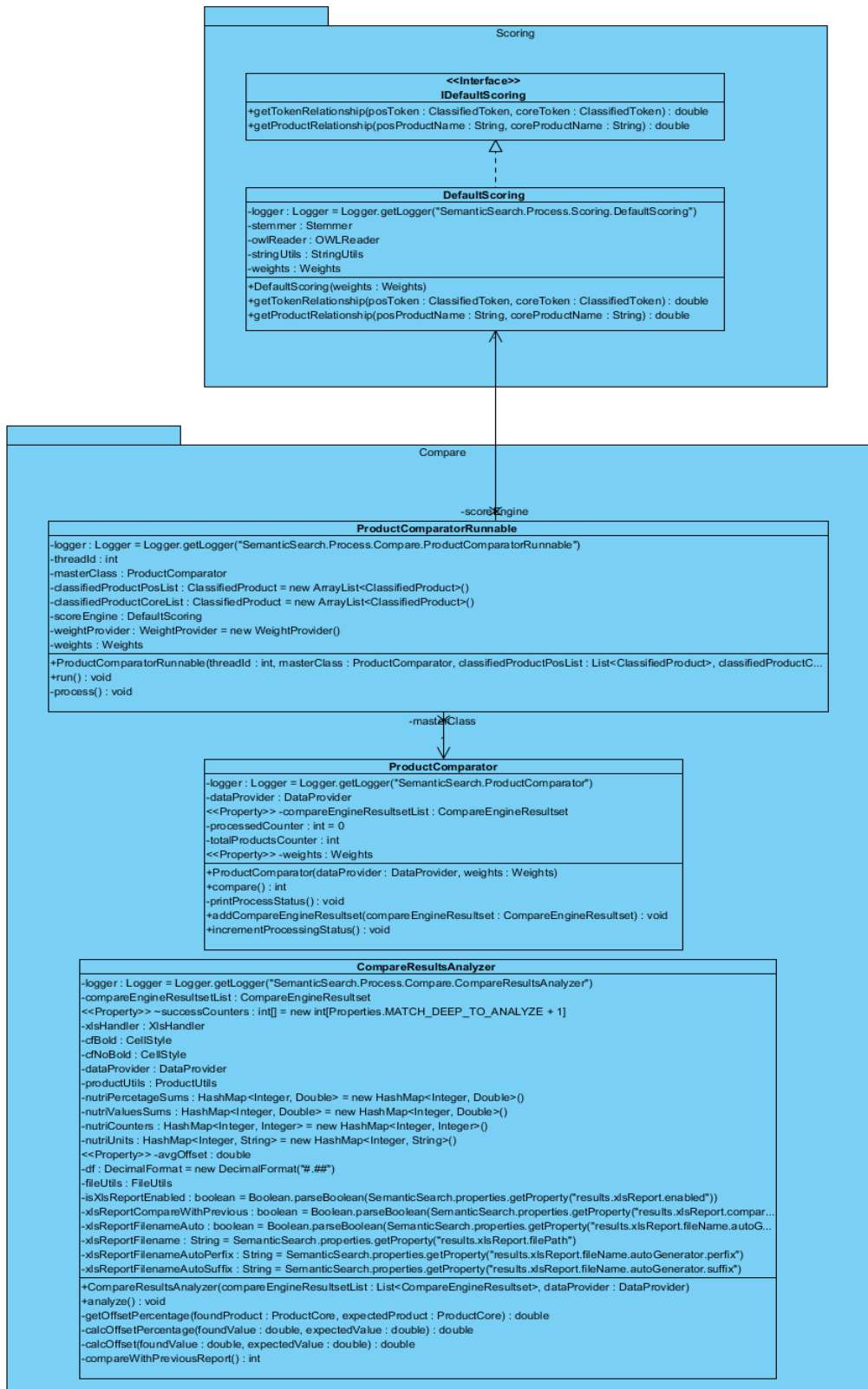


Figura 7. Diagrama de classes: Package "Process" – Parte 1

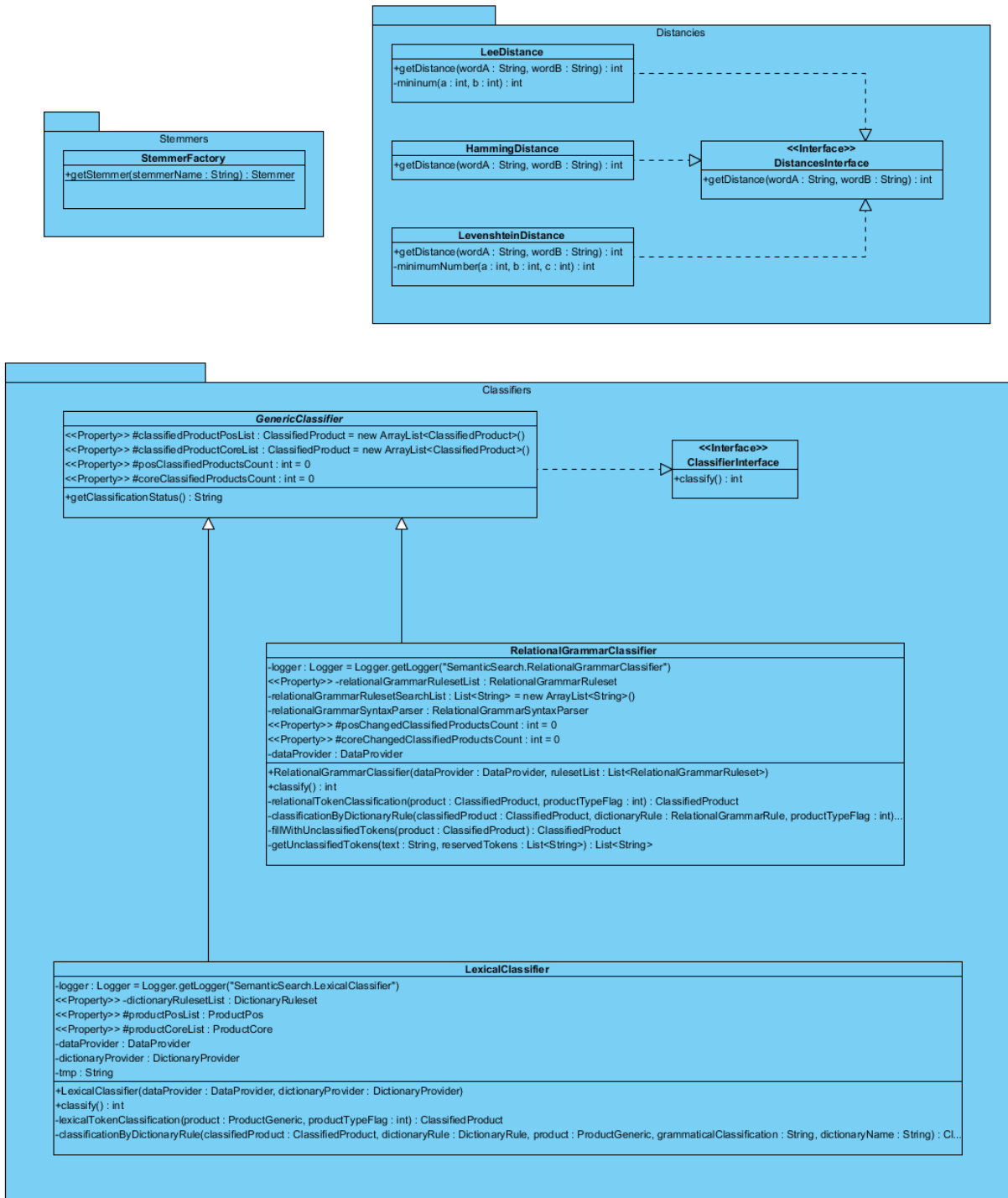


Figura 8. Diagrama de classes: Package "Process" – Parte 2

WeightProvider
<pre> <<Property>> -grammaticalWeights : Weight = new ArrayList<Weight>() <<Property>> -generalWeights : HashMap<String, WeightValue> = new HashMap<String, WeightValue>() -weightsXmlParser : WeightsXmlParser -logger : Logger = Logger.getLogger("SemanticSearch.DictionaryProvider") -fileUtils : FileUtils = new FileUtils() -grammaticalWeightMap : HashMap<String, Weight> +WeightProvider() +loadWeights(WeightFilesPackage : String) : void +getCustomWeightMap(product : ClassifiedProduct) : Map<String, Double> +getWeight(grammaticalClass : String, numberOfTokensInClass : int, priorityPos : int, priorityCore : int, refersToSameToken : boolean, customWeight : double, token... +getWeightByGrammaticalClass(grammaticalClass : String) : Weight +getAllWeightValues() : List<WeightValue> +modifyWeightValue(ValueID : int, newValue : double, outputFilePath : String) : int -normalizeGrammaticalWeights(grammaticalWeights : List<Weight>) : List<Weight> +getGrammaticalWeightsAsMap() : HashMap<String, Weight> +getGeneralWeightByName(name : String) : WeightValue </pre>

RelationalGrammarRulesProvider
<pre> -logger : Logger = Logger.getLogger("SemanticSearch.RelationalGrammarRulesProvider") -fileUtils : FileUtils = new FileUtils() -relationalGrammarXmlParser : RelationalGrammarXmlParser <<Property>> -rulesetList : RelationalGrammarRuleset -RELATIONAL_GRAMMAR_RULESET_PACKAGE : String = SemanticSearch.properties.getProperty("resources.relationalGrammarRulesPath") +RelationalGrammarRulesProvider() +loadDictionaries() : void </pre>

DataImportFilterProvider
<pre> -fileUtils : FileUtils = new FileUtils() -dataImportFilterXmlParser : DataImportFilterXmlParser <<Property>> -rulesetList : DataImportFilterRuleset -DATA_IMPORT_FILTER_RULESET_PACKAGE : String = SemanticSearch.properties.getProperty("resources.dataImportFilterRulesPath") +DataImportFilterProvider() +loadDataImportFilters() : void </pre>

DictionaryProvider
<pre> -logger : Logger = Logger.getLogger("SemanticSearch.DictionaryProvider") -fileUtils : FileUtils = new FileUtils() -dictionaryXmlParser : DictionaryXmlParser <<Property>> -rulesetList : DictionaryRuleset -DICTIONARY_RULESET_PACKAGE : String = SemanticSearch.properties.getProperty("resources.dictionaryRulesPath") +DictionaryProvider() +loadDictionaries() : void +getDictionaryRule(rulesetName : String, ruleID : int) : DictionaryRule </pre>

Figura 9. Diagrama de classes: Package "Providers" – Parte 1

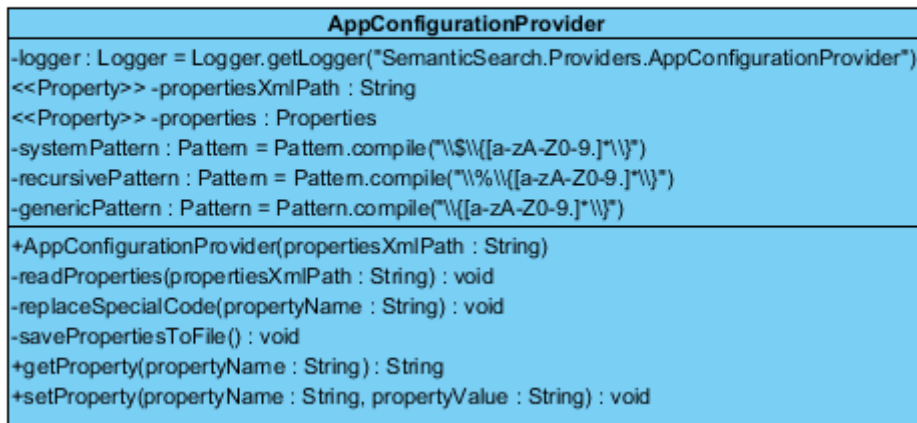


Figura 10. Diagrama de classes: Package "Providers" – Parte 2

Estudo e adaptação dos recursos disponíveis

Dado o foco deste trabalho (integração semântica de sistemas), é de extrema importância realizar uma análise profunda aos *datasets* de ambas as aplicações. Só assim será possível adaptar da melhor forma a solução implementada ao universo de dados existente.

Algumas das diretrizes aqui introduzidas deverão ainda ser tomadas como pressupostos para que seja possível ao sistema atingir graus de acerto/precisão aceitáveis.

5.1 Caracterização dos datasets

Para a elaboração deste trabalho são fornecidas duas fontes de dados distintas, devendo estas servir como base ao processo de integração.

A primeira destas fontes representa a base de dados do ponto de venda (POS), sendo esta originalmente disponibilizada em formato de texto e sem qualquer formatação definida. A informação por si oferecida cinge-se ao nome do produto comercializado, contando com um total de 1457 produtos.

A segunda fonte de dados servirá de repositório central para obtenção dos valores químicos e nutricionais de cada alimento. Esta é fornecida sob o formato proprietário da aplicação Microsoft Access®, disponibilizando cerca de quatro dezenas de atributos para cada um dos produtos especificados (entre os quais, valores energéticos, calóricos ou de gordura). Esta base de dados dispõe de um total de 972 produtos.

5.2 Análise dos datasets

Terminada a caracterização de alto nível aos *datasets* fornecidos, partimos agora para uma análise mais detalhada dos mesmos, servindo esta para definir a normalização que estes devem sofrer antes de serem sujeitos ao processo de integração semântica.

Esta é uma ação de extrema importância ao correto funcionamento da aplicação, pois a

existência de termos erróneos influenciará negativamente os resultados por si devolvidos.

5.2.1 Remoção e correção manual de termos

Após alguma análise aos *datasets* considerados, foi possível elaborar uma lista de 98 produtos fora de contexto (contidos na BD do POS), os quais não se enquadram no contexto da solução, tendo assim sido removidos. A sua presença no sistema deve-se ao facto de este ser utilizado, não apenas para a venda de produtos através dos serviços de bar e refeitório, mas também para a realização de encomendas de produtos necessários ao funcionamento dos mesmos (v.g. detergentes).

Concluída a remoção destes produtos, identificámos ainda um conjunto de 15 itens cujo nome estava truncado (na maioria dos casos a 38 caracteres), sendo que cada um destes foi devidamente completado por forma a fazer sentido junto da aplicação desenvolvida.

Outro dos pontos identificados prende-se com o facto de alguns dos produtos da BD do POS incluírem caracteres especiais deformados. Em suma, foi-nos possível contabilizar 5 termos nestas circunstâncias.

De entre a lista de produtos da BD do POS foi ainda possível identificar 22 itens cujo nome continha um ou mais erros ortográficos. A maioria destes erros estava relacionado com a escrita errónea de nomes de marcas, no entanto existiam também casos de palavras portuguesas escritas de forma incorreta.

Concluímos ainda que as nacionalidades e países de origem dos diversos produtos do POS eram representados de forma variável. Dado isto, decidimos aplicar o padrão ISO 3166-1 alpha-2 [11], o qual define um código único de 2 caracteres para cada um dos países actualmente reconhecidos. Esta normalização foi aplicada a 31 produtos da BD do POS.

Como medida extraordinária, foi ainda removida a indicação de categoria incluída no nome de alguns produtos (tais como “peixe”, “lácteos”, etc.), uma vez que esta poderia influenciar o funcionamento do algoritmo, tratando-se no entanto apenas de informação redundante introduzida na base de dados.

Todos os pontos acima referidos devem ser tomados como diretrizes na inclusão de novos produtos em qualquer uma das bases de dados. Só assim poderão ser atingidos os níveis ideais de acertos. Estes pontos podem ser consultados em detalhe através do anexo A.

5.2.2 Correção e normalização automática de termos

Para além do processo manual já apresentado, foi ainda possível identificar a existência de algumas abreviaturas pouco perceptíveis, assim como de alguma informação inútil à análise dos *datasets*. A normalização destes dados é feita em *runtime*, através da execução de um motor de manipulação de texto desenvolvido para o efeito. Esta ferramenta baseia-se na aplicação de expressões regulares sobre as listas de produtos carregadas de ambas as BDs.

Durante a sua implementação foi mais uma vez tomada em conta a necessidade de adaptabilidade por parte do sistema, sendo que o conjunto de regras por si utilizado é armazenado através de um conjunto de ficheiros do tipo Extensible Markup Language (XML) [12]. Os ficheiros criados são interpretados por ordem alfabética, sendo que as regras de cada um destes *rulesets* são aplicadas sequencialmente.

Tal como já referido, a aplicação é altamente escalável, estando preparada para interpretar qualquer ficheiro XML criado pelo utilizador, desde que este respeite um modelo XML Schema Definition (XSD) elaborado para o efeito.

```
<ns0:rule description="Empty brackets">
  <ns0:search>\(\)\</ns0:search>
  <ns0:replace></ns0:replace>
</ns0:rule>
```

Figura 11. Sintaxe de um elemento de um ficheiro de regras

Tal como pode ser comprovado pela figura Figura 11, a sintaxe destes ficheiros de regras é bastante simples, sendo possível a qualquer utilizador adaptar este processo aos *datasets* utilizados.

Para uma consulta detalhada das regras criadas deve ser consultado o anexo B (o qual contém cada um dos ficheiros XML criados para o efeito, assim como a sintaxe do modelo XSD a respeitar para a criação de novos ficheiros de regras).

Extração de atributos

Um dos passos fundamentais para o desenvolvimento de um mecanismo de integração semântica é a elaboração de um dicionário através do qual se torne possível obter o significado de cada um dos produtos a considerar. Este processo é levado a cabo através da execução dos motores de análise lexical e gramatical.

Ao longo deste documento será apresentado o comportamento definido para cada um destes mecanismos, sendo ainda exibidos alguns exemplos práticos do funcionamento dos mesmos.

Resumidamente, ambos os processos referidos visam acrescentar informação a uma sequência textual, tornando-a assim perceptível à aplicação.

6.1 Análise lexical

O processo de análise lexical consiste, de forma resumida, na conversão de uma sequência de caracteres numa sequência de elementos (palavras ou vocábulos) associados a um significado perceptível. Por outras palavras, trata-se da interpretação de um termo em bruto, transformando-o num conjunto de elementos com significado para a aplicação.

Esta análise toma como referência o valor de um determinado elemento de forma isolada. Por outras palavras, age sobre o elemento ignorando o sentido que este possa ter quando em conjunto com os restantes.

Com este processo surge um novo conceito – o de “*token*” –, sendo que este representa o menor elemento perceptível dentro de cada termo.

Costeletas (carne suína)	assadas (modo de confecção)	com	arroz (cereal)	e	feijão verde (legume)
-----------------------------	--------------------------------	-----	-------------------	---	--------------------------

Figura 12. Exemplo da divisão de um termo em *tokens*

Tal como comprovada a Figura 12 (através do último dos *tokens* identificados), a divisão

de um termo em *tokens* deve implicar a aplicação de filtros lógicos, não devendo limitar-se à divisão de um elemento textual tomando um caracter como referência (vulgo *split*).

À componente de *software* responsável por executar esta tarefa dá-se geralmente o nome de *lexer*, *scanner*, ou *lexical analyzer*. No contexto deste trabalho, chamar-lhe-emos *lexer* ou *lexical classification engine*.

6.1.1 Implementação

Por forma a fornecer ao *lexer* desenvolvido uma base de conhecimento sólida, foi necessário analisar individualmente cada um dos 2181 termos únicos resultantes do processo de análise dos *datasets* de referência.

Durante esta etapa os *tokens* de cada termo são identificados através da aplicação de expressões regulares, sendo-lhes de seguida atribuído um conjunto de classificações lexicais e gramaticais previamente definidas. Para uma identificação mais precisa do *token* pretendido, podem ainda ser aplicadas ordens de precedência entre as várias regras aplicadas (figura Figura 13).

```
<ns0:rule id="1970" isValid="true" weight="100" priorTo="1950" description="" isCaseSensitive="false">
  <ns0:search>(((de)?\s|-)*corte|portuguesas?)\b</ns0:search>
  <ns0:prerequisite>\b(couves?)\b</ns0:prerequisite>
  <ns0:lexicalMeaning>Tipo de couve</ns0:lexicalMeaning>
  <ns0:grammaticalMeaning>tipo de produto</ns0:grammaticalMeaning>
  <ns0:owldata>
    <ns0:subClassOf>
      <ns0:dicName>Produtos</ns0:dicName>
      <ns0:ruleID>1950</ns0:ruleID>
    </ns0:subClassOf>
  </ns0:owldata>
</ns0:rule>
```

Figura 13. Sintaxe de um elemento de um dicionário

Por forma a garantir a adaptabilidade da aplicação a mudanças de âmbito, as regras supra indicadas são definidas através de um conjunto de ficheiros XML (*rulesets*). É ainda disponibilizado um esquema em formato XSD, o qual deve validar qualquer conjunto de regras criado/editado, garantindo assim que não existirão problemas por parte da aplicação em interpretar o mesmo.

Tomando como base este esquema, assim como o elemento demonstrado na Figura 13, será de seguida exposta a sintaxe a ser respeitada por cada uma das regras a aplicar.

Em termos de atributos, cada regra deve dispor de um ID único de identificação, o qual

deverá ser representado por meio de um número inteiro positivo. Deverá ainda estar presente um atributo “isValid” responsável por indicar ao sistema se a regra deve ou não ser considerada. O atributo “weight” é também de preenchimento obrigatório, indicando qual o peso do elemento em situações em que duas ou mais regras classificam um único *token*, ou seja, indicando qual das classificações será aplicada.

Os atributos “priorTo” é por sua vez de preenchimento opcional, oferecendo a possibilidade de indicar à aplicação a precedência com que as regras devem ser processadas. O seu valor deve representar o ID válido de um elemento do próprio *ruleset*. Como resultado, teremos a garantia de que a regra em causa será executada antes da regra referida no atributo. Cada elemento poderá ainda fazer-se acompanhar de uma descrição textual (atributo “description”).

Por último surge ainda o atributo “isCaseSensitive”, o qual é de preenchimento obrigatório e deve conter um valor booleano (true/false). Este visa indicar à aplicação se esta deve ou não ser sensível à diferença entre letras maiúsculas e minúsculas das palavras identificadas através da regra em causa.

De preenchimento obrigatório, o elemento “search” representa a mais básica funcionalidade da regra, identificando qual o conjunto textual a capturar. Tal como já referido, o conteúdo deste elemento deve ser apresentado sob a forma de uma expressão regular. Cada regra poderá especificar um (e só um) elemento do tipo “search”.

Cada regra pode ainda dispor de um conjunto indeterminado de pré-requisitos (elemento “prerequisite”), os quais se apresentam também sob a forma de expressões regulares. Caso estes não se verifiquem, a regra não será aplicada pelo motor de classificação. Este elemento é de preenchimento facultativo, sendo que uma regra poderá não apresentar qualquer tipo de pré-requisitos.

É também possível definir o significado lexical atribuído aos *tokens* identificados através de uma determinada regra (empregando para isso o elemento “lexicalMeaning”). Este é de preenchimento obrigatório.

Os elementos “grammaticalMeaning” e “owldata” serão posteriormente abordados ao longo das secções 6.2 e 6.3.

Terminada a caracterização dos elementos “rule”, passaremos a descrever os vários atributos do seu elemento pai – “ruleset” (Figura 14).

```
<ns0:ruleset xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'  
  xmlns:ns0='http://xml.semanticsearch.pt/schema/model'  
  xsi:schemaLocation='http://xml.semanticsearch.pt/schema/model model.xsd'  
  grammaticalMeaning="produto"  
  name="Produtos"  
  version="2.0a"  
  description="Every single word that acts like a product by itself"  
  isValid="true"  
  priorToVerification="true">
```

Figura 14. Sintaxe da *tag* de abertura de um elemento “ruleset” (utilizado na definição de dicionários)

Para além do atributo “grammaticalMeaning”, cada *ruleset* deve ainda especificar um nome, uma versão, uma descrição textual, e por fim os atributos “isValid” e “priorToVerification”. Estes dois últimos atributos indicam, respetivamente, se o *ruleset* deve ou não ser considerado (é possível desativar um *ruleset*, tornando-o “invisível” aos motores de classificação), e se o motor de processamento semântico deve ou não proceder à verificação dos atributos “priorTo” definidos no ficheiro. Todos estes atributos são de preenchimento obrigatório.

Qualquer ficheiro XML colocado na diretoria de dicionários será interpretado como tal, desde que respeite o modelo XSD aqui apresentado.

Ao todo foram elaboradas um total de 4924 regras repartidas por 8 dicionários XML. O ficheiro XSD, assim como os *rulesets* elaborados, poderão ser consultados através do anexo C. Dada a extensão destes ficheiros, apenas o dicionário “Receitas” será fisicamente incluído nos anexos do documento. Os restantes dicionários estarão disponíveis em formato digital.

6.2 Análise gramatical

O estudo da linguística define a gramática como sendo o conjunto de regras que especifica de que forma os elementos de um conjunto de palavras (ou léxico) fazem sentido quando combinados. Por outras palavras, o conjunto de regras gramaticais de um idioma define onde e como empregar as palavras que dele fazem parte.

Aplicando este conceito ao trabalho apresentado, podemos considerar o processo de análise gramatical como aquele que atribui a cada *token* ou conjunto de *tokens* um significado

válido relativo ao contexto em que se insere.

Analisando a estrutura da língua portuguesa, podemos identificar um total de 10 classes gramaticais distintas (entre elas, substantivo, artigo, verbo, advérbio, etc.). No entanto, e dado o contexto do trabalho realizado, a adoção destas classificações não se mostra preponderante na análise aos termos dos *datasets*.

Posto isto, tomaremos como referência as seguintes classes gramaticais:

- Produto;
- Característica;
- Receita;
- Sabor;
- Tipo de produto;
- Finalidade.

Através destas classes ser-nos-á possível determinar, não apenas o significado de um determinado *token*, mas também qual a função que este tem no termo em que se insere.

O resultado prático da aplicação desta forma de análise morfológica pode ser comprovado através do exemplo representado na Figura 15.

Costeletas (produto)	assadas (característica)	com	arroz (produto)	e	feijão verde (produto)
-------------------------	-----------------------------	-----	--------------------	---	---------------------------

Figura 15. Exemplo da classificação gramatical de um termo

A segunda etapa do processo desenvolvido baseia-se numa nova fase de análise aos *tokens* já classificados, assim como dos restantes elementos do termo (ainda não identificados). Este processo identifica um conjunto de palavras-chave da língua portuguesa (determinantes, advérbios, etc.) aplicando de seguida um conjunto de regras relacionais sobre o termo. Isto permitirá ao nosso motor de classificação acrescentar informação a um determinado conjunto de *tokens*. Por outras palavras, são aplicadas regras da língua portuguesa sobre os termos em análise. Caso o *token* tenha já sido classificado, a sua

classificação original é descartada em detrimento da mais recente.

Para além do seu significado gramatical, este processo acrescenta ainda a cada *token* (caso aplicável) a identificação do *token* a que se refere (Figura 16).

Bolo (produto)	de -	Bolacha (atributo de produto referente a “Bolo”)	tradicional (característica)
-------------------	---------	---	---------------------------------

Figura 16. Exemplo da classificação gramatical através de regras relacionais

Esta informação é particularmente útil quando lidamos com características, pois torna-se possível estabelecer uma relação entre estas e o produto a que se referem.

6.2.1 Implementação

No seguimento da abordagem feita durante o capítulo 6.1.1 à sintaxe a respeitar por cada regra de dicionário, resta ainda referir que o modelo apresentado permite, através do elemento “*grammaticalMeaning*”, definir o significado gramatical a atribuir aos *tokens* identificados através de uma determinada regra. Este elemento é de preenchimento facultativo, sendo que, caso nada seja definido, será aplicado o significado gramatical definido como atributo do *ruleset* (Figura 14). Caso contrário, a classificação gramatical mais específica (definida em cada regra) terá sempre precedência face à mais genérica (definida no *ruleset*).

As regras de gramática relacional são, também elas, definidas através de *rulesets* em formato XML, os quais devem respeitar um XML Schema fornecido para o efeito.

Cada uma destas regras deve especificar a palavra-chave que pretende identificar, através de uma expressão regular e utilizando para isso o elemento “*search*”. O significado gramatical a atribuir deve também ser especificado, devendo para isso ser utilizado o elemento “*grammaticalMeaning*”.

O universo de *tokens* que será abrangido pela classificação em causa poderá também ser especificado (através elemento “*appliesTo*”). Embora não seja obrigatório, é ainda possível identificar o *token* ao qual o elemento classificado faz referência (apenas quando aplicável e através do elemento “*refersTo*”).

Para uma maior flexibilidade na utilização destes dois últimos elementos assim como na

utilização do elemento “grammaticalMeaning”, foi criada uma sintaxe especial, a qual nos permite fazer referência a outros *tokens* de forma dinâmica. Esta sintaxe será abordada em detalhe durante o capítulo 6.2.2.

6.2.2 Sintaxe dinâmica para regras de gramática relacional

De forma a garantir uma maior flexibilidade das regras, foi definido um conjunto de parâmetros dinâmicos, os quais podem ser utilizados na definição dos elementos “refersTo”, “appliesTo” e “grammaticalMeaning” de cada regra.

Através da Figura 17 é possível analisar a sintaxe definida, assim como as várias combinações que a mesma permite. Esta representação encontra-se formalizada através do padrão ISO/IEC 14977 / Extended Backus-Naur Form (eBNF) [13].

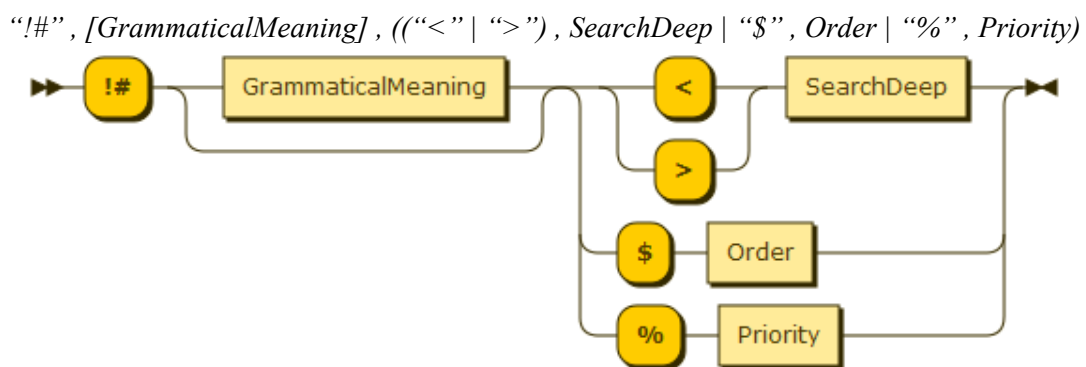


Figura 17. Sintaxe a respeitar para a definição de parâmetros dinâmicos

Segundo a sintaxe definida, qualquer parâmetro utilizado deve ser iniciado pela sequência de caracteres “!#”, seguindo-se (opcionalmente) a classe gramatical a que se faz referência. Após isto deve ser especificada a prioridade do *token* referido, ou a cardinalidade do mesmo (podendo esta ser absoluta – por exemplo, “segundo *token* identificado no termo” – , ou relativa – “penúltimo *token* identificado antes do *token* em análise”).

Deste modo torna-se possível replicar o significado gramatical de um *token*, atribuir a um *token* o papel de maior destaque dentro de uma classificação gramatical (tornando-o, por exemplo, no principal produto do termo onde se insere), entre outros. Vejamos alguns exemplos ilustrados na Tabela I:

Tabela I. Exemplos da utilização de parâmetros dinâmicos para gramática relacional

Expressão	Descrição
!#1	Primeiro <i>token</i> identificado no termo
!#<1	<i>Token</i> anterior
!#>2	Próximos dois <i>tokens</i>
!#produto\$1	Primeiro <i>token</i> identificado como "produto"
!#produto<1	Último dos <i>tokens</i> anteriores classificados como "produto"
!#produto%1	<i>Token</i> de prioridade máxima de entre os identificados como "produto" (prioridade=1)
!#produto%2	<i>Token</i> de prioridade nível 2 de entre os identificados como "produto" (prioridade=2)

A aplicação das regras de gramática relacional é feita através de um processo independente aos demais mecanismos de classificação.

Uma vez que o tipo de dados diverge do que focamos nos restantes mecanismos de classificação, as regras aplicadas são também armazenadas num conjunto de ficheiros distinto. Tal como em situações anteriores, são interpretados todos os ficheiros XML que sejam validados perante um modelo XSD fornecido para o efeito.

Para uma melhor perceção acerca do funcionamento destes parâmetros, assim como das próprias regras de gramática relacional, a Figura 18 representa uma das regras por nós definidas.

```
<ns0:rule id="5" isValid="true" weight="50" description="Atributos">
  <ns0:search>\b(de)\b</ns0:search>
  <ns0:grammaticalMeaning>atributo de produto</ns0:grammaticalMeaning>
  <ns0:refersTo>!#produto<1</ns0:refersTo>
  <ns0:appliesTo>!#>1</ns0:appliesTo>
</ns0:rule>
```

Figura 18. Sintaxe de um elemento do dicionário de gramática relacional

As restantes regras, assim como o modelo XSD responsável pela validação das mesmas encontram-se em anexo (anexo D).

6.3 Construção de ontologias

O conceito de ontologia não é mais que um modelo de dados capaz de representar um domínio de conhecimento, permitindo acrescentar aos seus elementos informação que os contextualiza perante os demais.

Através deste recurso ser-nos-á possível executar ações de inferência sobre um determinado produto ou característica, tornando-se assim possível à aplicação deduzir fatores de comparação até aqui desconhecidos.

Durante este capítulo será apresentado um mecanismo desenvolvido com vista a criar uma ontologia a partir da informação extraída dos ficheiros de dicionário já apresentados.

6.3.1 Tecnologias utilizadas

Para a implementação deste módulo, recorreremos à utilização do padrão Web Ontology Language 2 (OWL2) [14], um standard World Wide Web Consortium (W3C) de Outubro de 2009 que opera sobre a especificação do modelo de metadados Resource Description Framework (RDF) [15] (Figura 19).

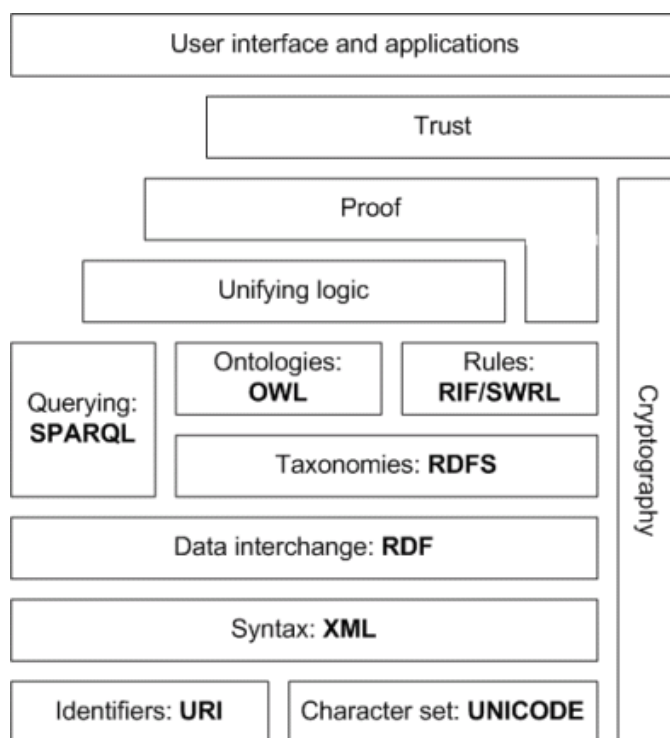


Figura 19. Camadas da arquitetura de sistemas baseados em Web Semântica[16]

Como principal característica, estes padrões oferecem notações abertas. Uma vez que o padrão RDF tem como base a notação XML, também este é representado através de um simples ficheiro de texto devidamente formatado, não estando a sua consulta ou edição dependente de qualquer aplicativo externo.

Para que a integração entre o código desenvolvido e a ontologia produzida fosse mais simples, recorreremos ainda à utilização de uma Application Programming Interface (API) própria para o efeito. Após uma busca pelas alternativas existentes recorreremos à utilização da ferramenta “The OWL API” na sua versão 3.2.4 (Julho de 2011) [17], pois à data da realização deste projeto esta era a solução que mais e melhor documentação fornecia. Mais recentemente, em Abril de 2012, uma das soluções alternativas, o Jena RDF API [18], obteve o estatuto de “Top Level Project” para a Apache Software Foundation¹². Através deste estatuto esta solução obteve um forte impulso no que toca à comunidade que a suporta.

Finalmente, e apenas para efeito de depuração do processo de criação de ontologias, recorreremos ainda à ferramenta Protégé (versão 4.1.0 – Julho de 2011). Esta é uma aplicação multiplataforma e de código aberto desenvolvida pela universidade americana de Stanford [19] que permite obter uma representação visual e simplificada de documentos OWL.

6.3.2 Implementação

Um dos pontos focados na implementação deste mecanismo foi a necessidade de criar um nível de abstração sobre o padrão OWL, permitindo ao utilizador que se abstraia totalmente da existência do ficheiro de ontologia. Aplicando esta abordagem, foi desenvolvido um sistema capaz de gerar automaticamente o ficheiro de ontologia com base na informação presente nos dicionários XML já mencionados.

Deste modo, para além de tornar mais simples a interação do utilizador com o sistema, garantimos ainda um melhor nível de adaptabilidade da aplicação, uma vez que os dados utilizados para a criação da ontologia fazem parte do conjunto de dicionários onde se encontra também toda a informação lexical e gramatical (existindo assim um ponto único de edição em caso de alterações nos *datasets*).

Estes dados são incluídos no dicionário XML através de um elemento “owldata” colocado dentro de cada regra (elemento “rule”). A Figura 20 representa uma das regras de dicionário, a qual inclui a informação necessária para a criação de uma ontologia.

¹² <http://incubator.apache.org/projects/jena.html>

```

<ns0:rule id="10" isValid="true" weight="100" priorTo="0" description="Água sem gás" isCaseSensitive="false">
  <ns0:search>\b(carvalhos|luso|serrana)\b</ns0:search>
  <ns0:prerequisite/>
  <ns0:lexicalMeaning>Bebida não alcoólica</ns0:lexicalMeaning>
  <ns0:grammaticalMeaning/>
  <ns0:owldata>
    <ns0:subClassOf>
      <ns0:dicName>Produtos</ns0:dicName>
      <ns0:ruleID>2810</ns0:ruleID>
    </ns0:subClassOf>
    <ns0:sameAs>
      <ns0:dicName>Produtos</ns0:dicName>
      <ns0:ruleID>2830</ns0:ruleID>
    </ns0:sameAs>
    <ns0:contains>
      <ns0:dicName>Características</ns0:dicName>
      <ns0:ruleID>1020</ns0:ruleID>
    </ns0:contains>
  </ns0:owldata>
</ns0:rule>

```

Figura 20. Sintaxe de um elemento de um dicionário (com foco na informação de ontologia)

Tal como demonstra a regra indicada na Figura 20, podem ser incluídos no conjunto de regras para criação da ontologia os elementos “subClassOf”, “sameAs” e “contains”. A presença de cada um destes elementos é opcional, sendo que apenas o elemento “subClassOf” apresenta restrições relativamente à sua cardinalidade, podendo surgir no máximo por uma vez em cada regra.

Cada um destes elementos deve indicar o ID da regra a que se refere, assim como o nome do dicionário ao qual esta pertence. Caso o conjunto de ID de regra e nome de dicionário não sejam válidos, o elemento não será considerado.

As regras especificadas através do elemento “owldata” serão aplicadas sobre os tokens identificados durante a fase de classificação lexical e gramatical. Estes tokens são tratados, à luz do padrão OWL, como *individuals*.

Partindo para uma especificação mais precisa dos vários elementos interpretados, o elemento "subClassOf" será mapeado para o elemento “type” implementado pelo padrão RDF. Este permitir-nos-á determinar a hierarquia dos *individuals* assim como das classes que os contêm. Uma vez que representa o significado de cada *token*, o campo "lexicalMeaning" é tomado como base para a criação das classes. Para uma melhor organização da árvore de *tokens* identificados, é ainda utilizado o significado gramatical de cada elemento como nó de nível superior (Figura 21).

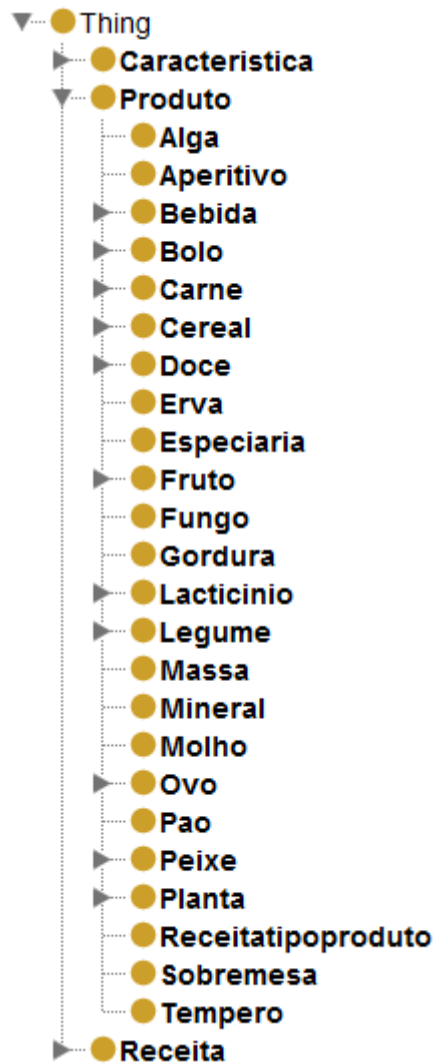


Figura 21. Organização de classes da ontologia gerada (representação gráfica pela aplicação Protégé)

Tal como pode ser comprovado pela Figura 21, todas as classes definidas descendem da classe “Thing”, sendo esta sucedida pelas classes criadas a partir do significado gramatical de cada *token* (classes de nível 2). Cada uma das classes de nível 2 é por sua vez composta pelas classes criadas através do significado lexical de cada *token* identificado. O elemento “sameAs” permite por sua vez descrever relações de equivalência entre dois *individuals*.

Para além das relações definidas através da área “owldata”, são também considerados como sinónimos todos os *tokens* que, embora não coincidentes, sejam identificados através da mesma regra. Esta consideração permite-nos capturar as variações em número e género dos elementos do *dataset*. Por exemplo, os *tokens* “ananás” e “ananases” serão considerados pela aplicação como sinónimos.

Por fim, é ainda possível definir elementos do tipo "contains", os quais permitirão especificar relações mais genéricas entre os *tokens*, definindo se um *individual* contém outro (Figura 22).

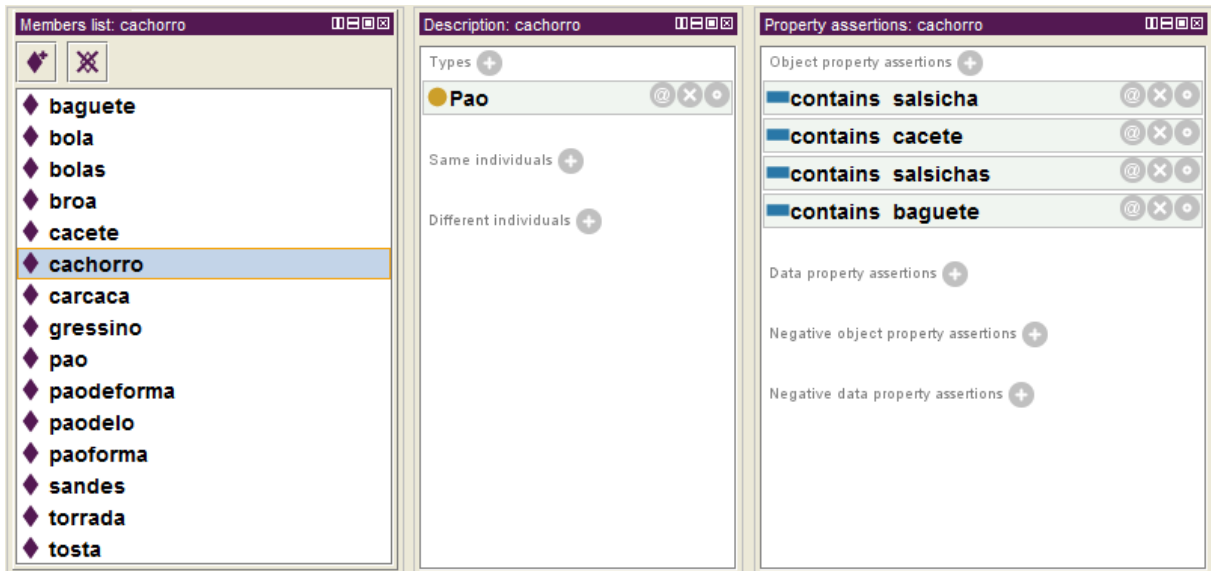


Figura 22. Organização de classes da ontologia gerada (representação gráfica pela aplicação Protégé)

Através da Figura 22 podemos analisar o efeito da relação “contains” no caso prático do produto “cachorro”. Assim, e de acordo com as relações estabelecidas, um cachorro poderá conter os produtos “cacete”, “baguete” e “salsicha”. É ainda de notar que, uma vez que “salsicha” e “salsichas” são considerados como sinónimos pela aplicação, os dois produtos surgem na lista de *tokens* abrangidos pelo elemento “contains”. Para além disto, podemos ainda verificar o efeito do elemento “subClassOf” através da secção “Types”. Este define “cachorro” como pertencendo à *class* OWL “Pão”.

Motor de processamento semântico

Uma vez terminado o processo de enriquecimento dos termos da base de dados, passaremos a apresentar o processo de processamento semântico.

Este processo terá como função utilizar os atributos já obtidos por forma a estabelecer relações entre os *tokens* identificados. Como resultado final, cada relação será quantificada através de um sistema de pontuação absoluta.

7.1 Algoritmo de processamento

Em termos gerais, o algoritmo de comparação focar-se-á em dois pontos específicos da execução: a comparação entre termos (utilizando para isso os produtos de cada BD como um todo) e a comparação entre os atributos inferidos durante o processo de classificação.

Estes pontos podem ser consultados através da Figura 23, na qual surgem a negrito.

```
Para cada produto do POS
  Obter pesos a aplicar
  Para cada produto da BD Central
    Quantificar relação entre produtos
    Se relação > 0
      Armazenar o resultado
    Senão
      Para cada classe gramatical considerada
        Para cada token do produto do POS
          Para cada token do produto da BD central
            Quantificar relação entre tokens
            Se relação > 0
              Armazenar o resultado
  Fim.
```

Figura 23. Algoritmo de processamento semântico: Visão geral

Tal como demonstrado através do excerto de pseudocódigo acima, a comparação de *tokens* é executada apenas caso a comparação direta entre termos não retorne qualquer resultado.

Passaremos desta forma a descrever em detalhe o processo de comparação de *tokens*. O processo de comparação direta não será diretamente abordado, pois funciona de forma

idêntica a este. Em suma, o seu algoritmo consiste na execução dos três primeiros passos do processo de comparação de *tokens*.

Por forma a simplificar a representação do algoritmo, token_A e token_B representarão, respetivamente, os *tokens* obtidos através do produto da BD do POS e da BD central.

```
Se tokenA == tokenB
  retorna resultado
Senão
  Se tokenA normalizado == tokenB normalizado
    retorna resultado
  Senão
    Se radical de tokenA == radical de tokenB
      retorna resultado
    Senão
      Se OWL individual de tokenA == OWL individual de tokenB
        retorna resultado
      Senão
        Se um OWL individual contem o outro
          incrementa resultado
        Se os OWL individuals dos tokens são irmãos
          incrementa resultado
        Se um token coincide com uma das classes OWL do outro
          incrementa resultado
        Se existe uma relação entre as classes OWL dos tokens
          incrementa resultado
      retorna resultado
Fim.
```

Figura 24. Algoritmo de processamento semântico: Comparação entre *tokens*

Resumidamente, o motor de comparação de *tokens* (assim como o motor de comparação direta) deverá quantificar cada uma das relações possíveis. As restantes questões relativas ao mecanismo de *scoring* serão abordadas no decorrer do capítulo 7.2.

A comparação acima representada divide-se em duas etapas: a primeira destas visa determinar se os *tokens* em causa coincidem, partilham o mesmo significado ou se apenas têm semelhanças do ponto de vista morfológico, enquanto que a segunda etapa se destina a quantificar relações de outra ordem (tais como relações hierárquicas, etc.).

A primeira das validações verifica se os *tokens* coincidem, ou seja, se se trata da mesma sequência de caracteres. Caso os *tokens* não coincidam, esta validação será novamente executada, no entanto os *tokens* serão previamente submetidos a um processo de normalização. Este processo visa remover todo e qualquer caractere especial detectado em ambos os *tokens*.

O terceiro passo deste processo é bastante semelhante ao anterior, sendo no entanto utilizado como base de comparação o radical de cada *token*.

O radical da palavra é-nos fornecido por um *stemmer*, uma ferramenta destinada a aplicar as regras lexicais de um determinado idioma por forma a obter a maior parte invariável de uma palavra. Isto permitir-nos-á, nomeadamente, abstrairmo-nos das variações em número ou em género que uma palavra pode assumir.

O resultado da aplicação do *stemmer* não tem obrigatoriamente de coincidir com a raiz morfológica da palavra, bastando muitas vezes que forneça uma base de comparação entre palavras da mesma área lexical (Tabela II).

Tabela II. Exemplo da decomposição de palavras nos seus radicais

Palavra	Radical
Aromatizado	Arom
Aroma	Arom
Aromas	Arom

O sistema desenvolvido está preparado para fazer uso de três *stemmers* adaptados à língua portuguesa – Porter stemmer [20], Orenge stemmer [21] e Savoy stemmer [22] [23].

O último passo desta primeira fase de comparação visa verificar se os OWL *individuals* criados a partir dos *tokens* em causa são sinónimos, ou seja, se partilham uma relação OWL do tipo “sameAs”. Os passos acima descritos podem ser consultados através do diagrama de atividade UML exibido na Figura 25.

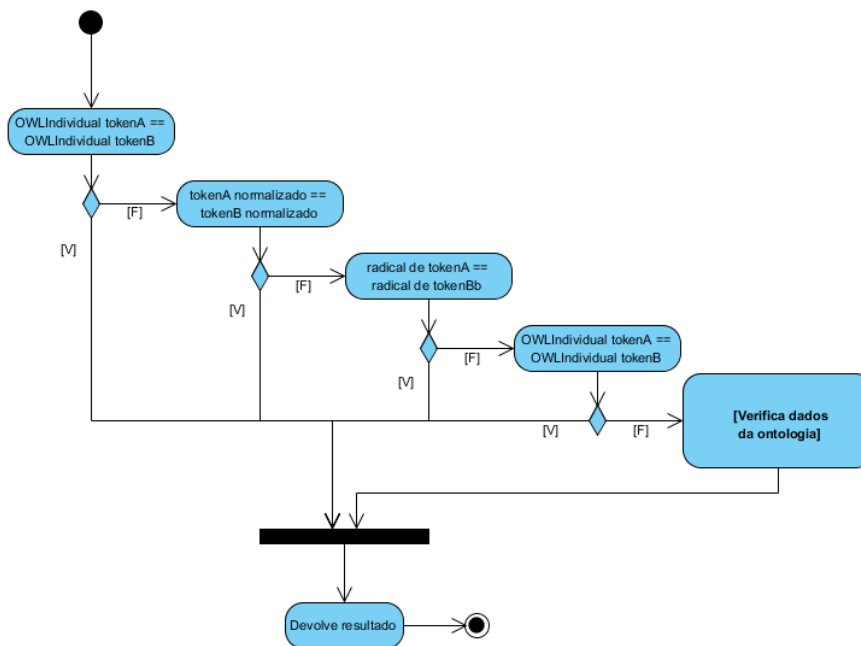


Figura 25. Algoritmo de processamento semântico: Comparação simples entre *tokens*

Uma vez apresentada a primeira parte do mecanismo de comparação de *tokens*, resta descrever a segunda etapa que compõe este processo. Esta é composta por quatro verificações sequenciais, as quais serão apresentadas de seguida. Uma vez que se complementam mutuamente, o resultado destas etapas é acumulado. Este processo será descrito com mais detalhe durante o capítulo 7.2.

O primeiro dos passos consiste em verificar se um dos OWL *individuals* contém o outro. Esta verificação toma como base as relações do tipo “subClassOf” previamente definidas durante a construção do ficheiro de ontologia. Após isto verificaremos se ambos os OWL *individuals* descendem diretamente da mesma classe OWL, ou seja, se em termos hierárquicos podem ser considerados irmãos.

De seguida averiguaremos se qualquer um dos *tokens* coincide com a classe dos demais, e finalmente verificaremos qual a relação hierárquica entre as classes de ambos os *tokens* (quantificando a distância entre as classes imediatamente acima de cada um destes).

A representação gráfica deste processo pode ser analisada através do diagrama de atividade UML exibido na Figura 26. O processo de comparação pode ser consultado com mais detalhe através dos meios fornecidos em formato digital.

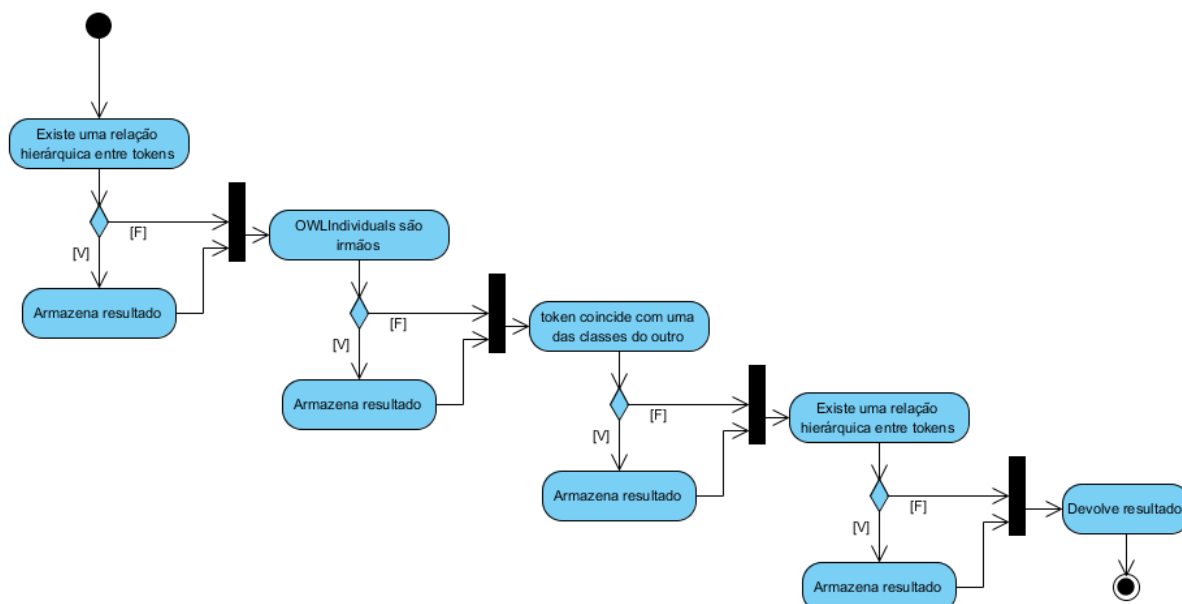


Figura 26. Algoritmo de processamento semântico: Comparação de *tokens* com base na ontologia

É ainda de realçar que a execução de cada um destes passos é opcional, sendo possível personalizar estes parâmetros à medida dos requisitos do utilizador.

7.2 Mecanismo de pontuação

O mecanismo de pontuação é sem dúvida uma das componentes mais importantes de qualquer motor de busca. Uma vez que é este mecanismo quem define o grau de similaridade entre dois produtos/descrições, é fundamental que esteja calibrado para oferecer o máximo de acertos possível.

Por forma a garantir que será viável confiar nos resultados fornecidos, é também importante que estes sejam normalizados, ou seja, que o intervalo de pontuação atribuído não varie de execução para execução. Por outras palavras, cada execução deve ser avaliada através de uma escala fixa (como uma percentagem), sendo que os resultados devem ser pontuados de forma proporcional.

Com o intuito de agilizar o processo de desenvolvimento começámos por analisar a possibilidade de adaptar a biblioteca Apache Lucene às nossas necessidades [24]. Esta solução, mantida pela Apache Software Foundation é distribuída sob uma licença de código aberto, e disponibiliza as funcionalidades necessárias ao funcionamento de um motor de busca.

De entre as funcionalidades disponíveis destacam-se a indexação de dados, mas também o processamento dos mesmos, assim como a aplicação de um mecanismo de pontuação.

Esta biblioteca destina-se principalmente à pesquisa de informação em documentos, no entanto, é por vezes possível adaptar as suas capacidades às necessidades de outros tipos de motor de busca. Apesar disto, acabámos por concluir que o motor de busca aqui apresentado não representa um destes casos, não sendo assim possível fazer uso desta ferramenta.

Após uma análise mais cuidada da documentação existente, assim como dos requisitos da nossa aplicação, concluímos que a necessidade de lidar com resultados normalizados não poderia ser garantida caso nos baseássemos na biblioteca Lucene. Apesar de ser bastante personalizável, esta biblioteca não garante no entanto que os resultados por si devolvidos sejam comparáveis através de uma escala normalizada¹³.

Para além disto concluímos ainda que, mesmo sendo uma solução bastante personalizável,

¹³ <http://wiki.apache.org/lucene-java/ScoresAsPercentages>

esta se baseia apenas na comparação direta de termos. Este fator impossibilita a implementação de filtros baseados em informação mais elaborada, tal como os dados gramaticais ou a informação inferida através da ontologia criada.

Posto isto, partimos para a implementação de um mecanismo de pontuação desenhado especificamente para as necessidades da nossa aplicação.

7.2.1 Implementação

Durante este capítulo passaremos a descrever o funcionamento do mecanismo de pontuação implementado.

Tal como já referido, o seu modo de operação visa garantir a precisão dos resultados obtidos, sendo para isso necessário que estes sejam produzidos de forma proporcional. Para isso decidimos basear a nossa implementação na utilização de razões (ou fatores de multiplicação).

Em termos práticos o mecanismo de pontuação baseia-se em dois valores: o resultado da comparação e o melhor resultado possível para cada processo de comparação. O primeiro destes valores representa o resultado da comparação e materializa-se através de uma razão compreendida entre 0 e 1. O segundo valor representa o resultado máximo ao qual a comparação pode chegar.

Após obter estas duas variáveis, o mecanismo de pontuação calculará o resultado efetivo obtido (multiplicando o resultado da comparação pelo resultado máximo) e armazena-o em conjunto com o melhor resultado possível. No final do ciclo de comparações é calculado o resultado absoluto, o qual será apresentado sob a forma de uma percentagem.

$$Resultado_{absoluto} = \frac{\sum(valor_{comparação} * resultado_{máximo})}{\sum resultado_{máximo}}$$

Figura 27. Fórmula de cálculo para a obtenção do resultado absoluto de uma comparação

Para que este processo possa ser mais facilmente compreendido, passamos a apresentar alguns exemplos.

a) Exemplo 1

Neste primeiro exemplo focaremos na comparação da descrição de dois produtos

morfologicamente diferentes mas cujo significado coincide. Consideremos então que “frango” representa o produto_A e que “galinha” representa o produto_B.

De acordo com os fluxos de processamento descrito através da Figura 25, ambos os termos passarão por três etapas das quais não sairá qualquer resultado positivo: a comparação direta (letra a letra), a comparação normalizada (ignorando caracteres especiais), e finalmente a comparação de radicais obtidos através dos termos.

Na quarta etapa da primeira fase de comparação a aplicação analisa o ficheiro OWL criado e deduz que existe uma relação de equidade entre os termos. Esta relação é quantificada através de uma razão entre 0 e 1 definida pelo utilizador (consideremos o valor 0,9).

Concluída a comparação dos termos, será invocado o mecanismo de pontuação. Este recebe e armazena não apenas o resultado obtido (0,9), como também o melhor resultado possível (neste caso, 1).

Seguindo o diagrama de atividade, e uma vez que a comparação direta de termos gerou resultados, não nos será necessário avançar para a comparação entre *tokens*.

Deste modo resta apenas ao mecanismo de pontuação determinar o resultado absoluto desta comparação. Para isto, e de acordo com a fórmula exibida na Figura 27, este mecanismo calculará o resultado final da comparação da seguinte forma:

$$Resultado_{absoluto} = \frac{0,9 \times 1}{1} = 0,9$$

Terminado este processo, o resultado é armazenado junto de todos os outros. No final de todas as comparações os resultados serão ordenados por ordem de relevância.

a) Exemplo 2

Neste segundo exemplo demonstraremos os passos executados pela aplicação para comparar os produtos “Cornetto” e “Gelado de leite”. Tal como no exemplo acima, os produtos em causa serão representados por produto_A e produto_B, respetivamente.

Uma vez que os termos (por si só) não apresentam qualquer similaridade do ponto de vista morfológico ou de significado, a primeira fase de comparação será concluída sem que possamos obter qualquer resultado. Deste modo, o algoritmo avançará para a comparação

baseada em tokens.

A primeira tarefa desta etapa será percorrer a lista de classes gramaticais disponíveis verificando se cada uma destas está presente em ambos os produtos. Neste caso começaremos por obter o token “Cornetto” (classificado como “produto”), o qual será comparado com cada um dos tokens do produto_B com esta classe gramatical. Posto isto, o algoritmo começará por comparar os tokens “Cornetto” e “Gelado”.

Após percorrer o fluxo de comparações acima apresentado a aplicação concluirá, através dos dados da ontologia, que os tokens são sinónimos, atribuído a esta relação um valor entre 0 e 1 (consideremos o valor 0,8). A aplicação calculará ainda o peso máximo desta comparação (consideremos 80). Uma vez obtidos, estes dados serão armazenados para que os possamos considerar no final do processo de comparação.

Continuando o processamento de tokens cuja classe gramatical seja “produto”, a aplicação comparará ainda “Cornetto” com o token “leite”.

Ao analisar a ontologia criada, esta identificará que o primeiro dos tokens contem o segundo, atribuindo a esta relação um valor positivo. Para efeitos de demonstração consideremos 0,4 como sendo o resultado da comparação e 10 como sendo o peso máximo que a comparação pode atingir. Estes dados serão guardados junto dos anteriormente calculados.

Uma vez que o produto_A não contém qualquer token classificado através de outra classe gramatical, o processamento terminará aqui, restando agora à aplicação calcular o resultado efetivo da comparação entre produtos.

Seguindo a fórmula apresentada na Figura 27, o resultado absoluto desta comparação será calculado da seguinte forma:

$$Resultado_{absoluto} = \frac{(0,8 \times 80) + (0,4 \times 10)}{80 + 10} = 0,76$$

Finda a comparação dos termos, teremos um grau de certeza de 76% em como estes representam o mesmo produto.

O produto da BD do POS (“Cornetto”) será ainda comparado com os restantes produtos da

BD central. Qualquer resultado originado por este processo será igualmente armazenado. Terminadas as comparações, os resultados serão ordenados por grau de relevância/similaridade.

Experimentação

A qualidade do algoritmo implementado é facilmente mensurável através de métricas quantitativas. Com vista a tornar possível a implementação de um processo de avaliação do mesmo, revelou-se necessário estabelecer associações estáticas (previamente conhecidas) entre cada um dos produtos de ambas as bases de dados. Ou seja, foi necessário associar manualmente um termo da BD central a cada um dos produtos da BD do POS.

Neste contexto foi-nos possível identificar um novo pressuposto – a necessidade de que cada produto do POS considerado corresponda obrigatoriamente a uma entrada da BD central. Esta condição garantir-nos-á que o algoritmo desenvolvido estará à partida apto a determinar a correspondência correta para todos os termos do *dataset* considerado.

Deste modo, é importante ter em conta que nem todos os produtos do POS dispõem de uma entrada equivalente na tabela central. Assim, e dado não ser possível verificar a veracidade dos dados gerados pelo algoritmo nestes casos particulares, estas entradas não serão consideradas pelo motor de processamento semântico. A lista das relações estabelecidas está disponível através do anexo E.

Como resultado deste processo foi elaborada uma lista composta por 615 associações únicas. Este é um processo de grande importância na garantia de qualidade por parte do algoritmo desenvolvido, uma vez que algumas das associações estabelecidas não são facilmente associáveis (por exemplo, o produto “Almofadinha Mista” da BD do POS está associado ao produto “Pastel folhado” da BD central).

Após determinar cada uma das correspondências a aplicação verificará os atributos nutricionais do produto esperado (através das relações estáticas definidas), mas também do produto obtido. Após isto será calculada a percentagem média de erro entre os valores nutricionais do produto esperado e do produto obtido. O resultado deste cálculo servirá de métrica à avaliação dos resultados gerados.

A Tabela III representa de que forma é determinado este valor.

Tabela III. Exemplo prático sobre a avaliação de resultados da aplicação

Produto	Valor nutricional	Valor esperado	Valor obtido	Desvio (%)
Produto _A	Energia	150	100	50
	Gordura	4	4	0
	Proteínas	40	30	33
Produto _B	Energia	30	30	0
	Gordura	2	2	0
	Proteínas	10	10	0

Os dados apresentados na tabela acima permitem-nos determinar um desvio médio de 28% entre os valores nutricionais do produto esperado e os valores nutricionais do produto calculado para o item do POS Produto_A. É ainda possível concluir que, no caso do Produto_B, os valores nutricionais coincidem, logo o desvio será nulo.

Como consequência dos resultados acima apresentados, esta execução da aplicação será avaliada com uma percentagem média de falha de 14%.

A par do objetivo de minimização do erro médio, existe ainda o objetivo de minimização do número de associações calculadas cujo resultado seja diferente do esperado, ou seja, o número de resultados errados obtidos pela aplicação.

Num cenário ideal ambos os objetivos deverão ser minimizados, ou seja, pretende-se obter uma percentagem média de erro nula, e uma contagem de associações falhadas também nula.

8.1 Otimização de resultados

Esta aplicação, tal como qualquer motor de busca, baseia-se num conjunto de pesos previamente definidos. No caso da aplicação aqui apresentada estes valores foram definidos manualmente, o que significa que não existe qualquer garantia que sejam aqueles que melhores resultados oferecem à aplicação.

Dado isto, resolvemos implementar um algoritmo de otimização multiobjetivo sobre o conjunto de pesos definido. Este visa determinar os valores a aplicar para que a aplicação se aproxime o mais possível dos resultados pretendidos.

Neste caso específico pretendemos que a aplicação minimize ambas as funções objetivo

estipuladas – a percentagem média de erro entre valores nutricionais e o número de correspondências falhadas.

Para isto resolvemos utilizar o algoritmo Nondominated Sorting Generic Algorithm II (NSGA-II). A escolha por este algoritmo evolucionário deveu-se principalmente ao facto de ser bastante popular na resolução de problemas genéricos, o que faz com que disponha de uma grande quantidade de documentação. Este algoritmo baseia-se nos conceitos de não-dominância e de *crowding distance* (ou densidade de vizinhos) para assim encontrar os pontos mais próximos do resultado pretendido [25].

Ao conjunto dos pontos mais próximos do resultado ideal ou ótimo é dado o nome de *pareto front*. Aos pontos que formam esta linha é dado o nome de pontos não-dominados.

Dado que a aplicação apresentada foi desenvolvida utilizando a linguagem Java, resolvemos adaptar a *framework* jMetal às nossas necessidades. Esta disponibiliza o acesso a um conjunto de algoritmos de otimização multiobjetivo, entre os quais o já referido NSGS-II. Para além dos algoritmos implementados, esta solução disponibiliza ainda a implementação de vários problemas clássicos, assim como de um conjunto de indicadores de qualidade [26].

A implementação deste mecanismo passou por disponibilizar na nossa aplicação uma *façade* acessível ao motor jMetal. Deste modo, o motor de otimização recorre aos métodos disponibilizados pela aplicação desenvolvida.

Em primeiro lugar este força a leitura dos pesos definidos (armazenados em formato XML), sendo que de seguida os modifica por forma a corresponderem aos valores calculados durante o processo de otimização. A aplicação desenvolvida para comparação e avaliação do erro utiliza assim o conjunto de pesos encontrados pelo algoritmo NSGA-II, devolvendo os resultados obtidos a este motor de otimização. Este utilizá-los-á para calcular o próximo conjunto de pesos a considerar no ciclo evolutivo característico deste algoritmo de otimização.

Este processo encontra-se representado através da Figura 28.

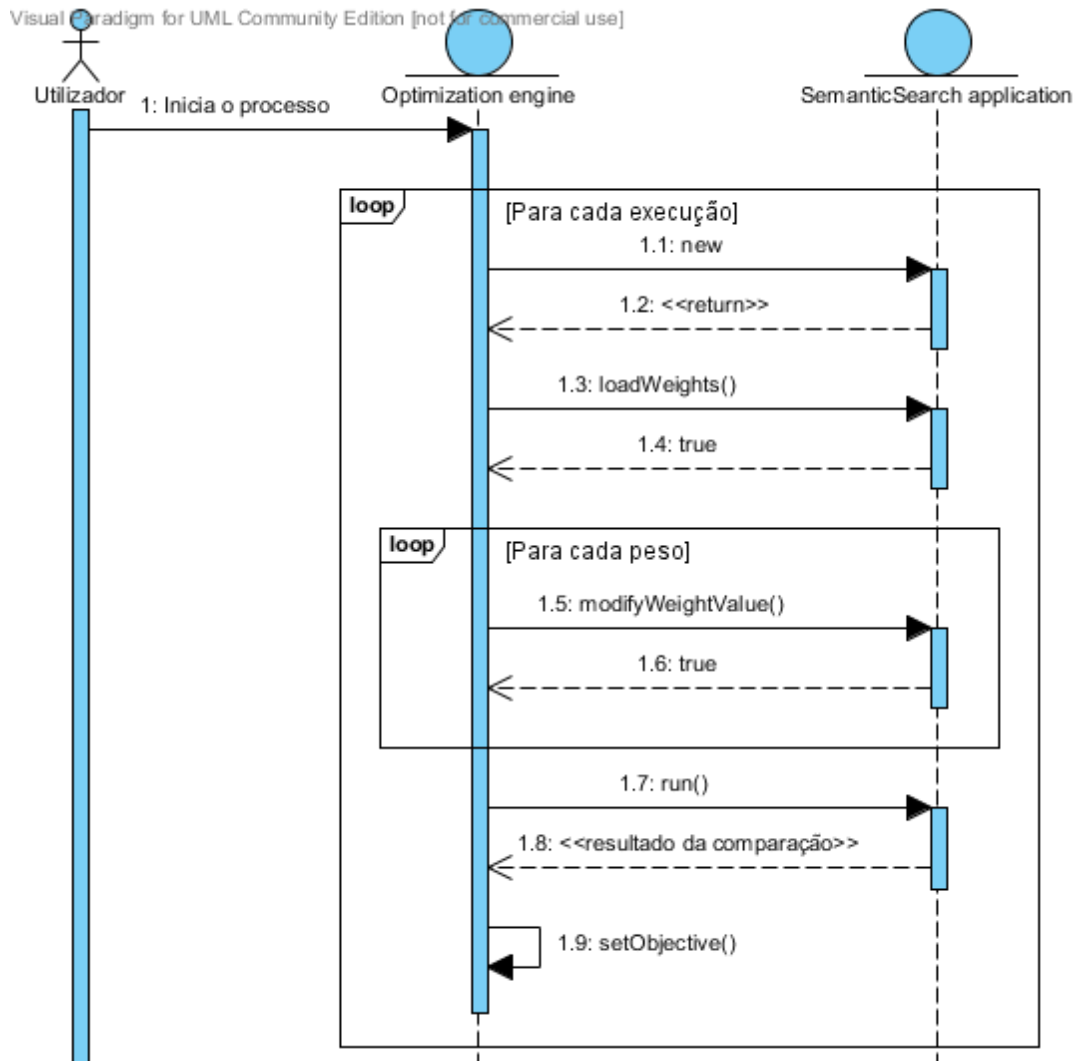


Figura 28. Diagrama de sequência: Comunicação entre motor de otimização e aplicação desenvolvida

Tal como já referido, o propósito do processo de otimização será minimizar ambas as funções objetivo definidas – percentagem média de erro entre valores nutricionais e número de associações incorretas determinadas pela aplicação.

Para a execução deste mecanismo de otimização resolvemos utilizar a parametrização definida por omissão pela própria framework jMetal para o algoritmo NSGA-II.

Dado o elevado peso computacional implicado no cálculo das funções objetivo, bem como a correspondente duração das experiências, fomos forçados a redimensionar o número de avaliações do NSGA-II para 4000. Caso contrário o tempo consumido pelo processo de otimização excederia largamente os tempos aceitáveis para a realização desta dissertação.

Cada experiência teve ainda assim a duração aproximada de uma semana num computador com processador *quad-code* AMD e 2GB de memória RAM. Para comprovar que os resultados de otimização são consistentes, repetiram-se 3 experiências independentes (ao invés das 30 inicialmente definidas pela *framework* jMetal).

À parte destas duas exceções, todos os restantes parâmetros de configuração do NSGA-II foram mantidos. Assim, foi considerada uma população de 100 indivíduos (neste caso cada indivíduo representa um conjunto de pesos correspondente às pontuações usadas na avaliação de semelhança entre descrições de produtos).

Forão também empregues os operadores genéricos de *binary tournament selection*, *SBX crossover* e *polynomial mutation*. Estes dois últimos, executados com um índice de distribuição = 20 e com uma probabilidade de 0,9 e $1/\text{número de variáveis de decisão}$, respetivamente.

Concluída a execução, pudemos comprovar que a tendência dos resultados obtidos correspondeu ao esperado, ou seja, os resultados das três experiências apresentaram valores bastante semelhantes (Figura 29).

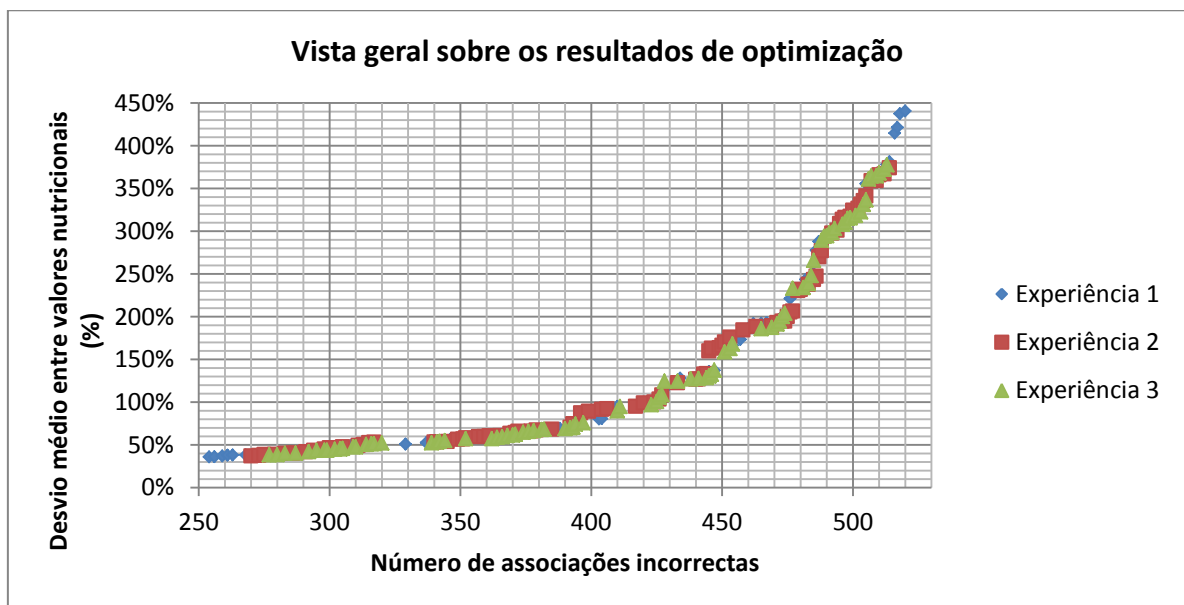


Figura 29. Resultados da execução do motor de otimização multiobjetivo: Vista geral

Analisando estes valores conseguimos concluir que o resultado que mais se aproxima dos valores esperados foi obtido durante a primeira experiência. O conjunto dos 10 melhores resultados obtidos é apresentado com maior detalhe através da Figura 30.

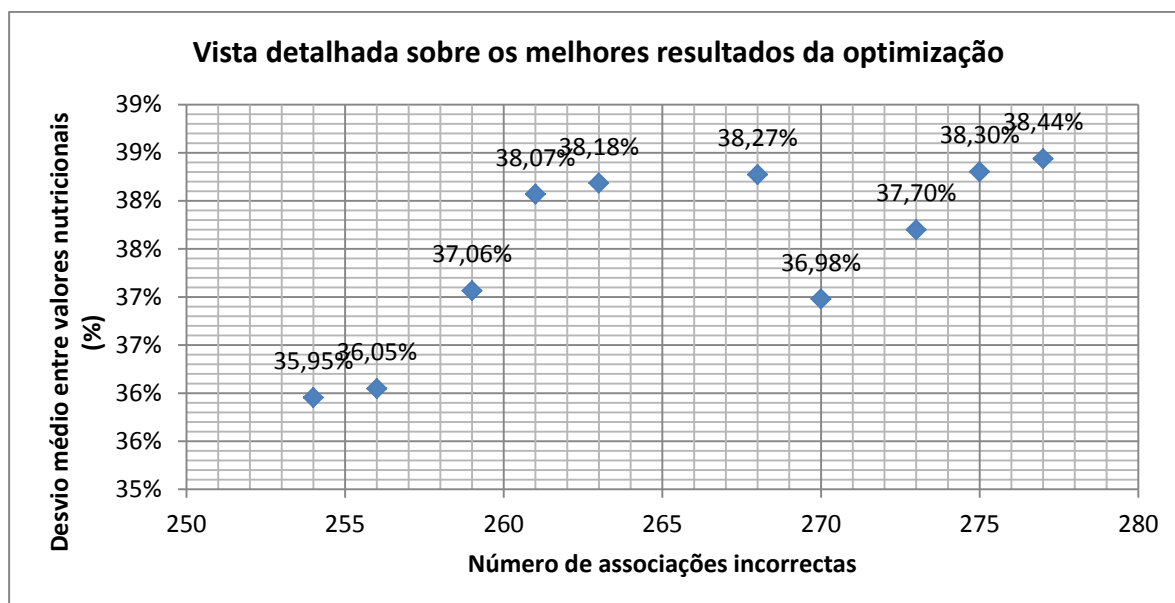


Figura 30. Resultados da execução do motor de otimização multiobjetivo: Vista detalhada sobre os 10 melhores resultados obtidos

Dados os resultados apresentados, podemos concluir que existe apenas um ponto não-dominado, uma vez que este supera os demais em ambos os objetivos definidos. O conjunto de pesos aqui calculado resulta numa percentagem média de erro entre valores nutricionais de 35,95%, permitindo-nos determinar com sucesso 59% das relações estáticas definidas (identificando com sucesso 361 das 615 relações).

Estes dados podem ser consultados com mais detalhe através da Tabela IV. A lista completa dos 50 melhores resultados obtidos poderá ser consultada através do anexo F.

Tabela IV. Valores obtidos no melhores resultados de otimização

#	Desvio (%)	Acertos	Falhas	Falhas (%)
1	35,95%	361	254	41,30%
2	36,05%	359	256	41,63%
3	36,98%	345	270	43,90%
4	37,06%	356	259	42,11%
5	37,70%	342	273	44,39%
6	38,07%	354	261	42,44%
7	38,18%	352	263	42,76%
8	38,27%	347	268	43,58%
9	38,30%	340	275	44,72%
10	38,44%	338	277	45,04%

Terminado o processo de otimização, restou-nos recolher os pesos associados ao resultado #1, passando estes a ser considerados como os valores ideais para a solução implementada.

Conclusões

Com este capítulo pretende-se apresentar uma síntese das conclusões mais relevantes deste trabalho.

Em primeiro lugar serão analisados os objetivos inicialmente definidos para este projeto, bem como se estes foram ou não devidamente cumpridos. De seguida serão expostas as principais dificuldades sentidas durante o desenvolvimento do trabalho. Serão ainda apresentadas as contribuições que julgamos mais importantes para o meio científico e académico, e por fim serão enumerados alguns tópicos que julgamos possam vir a ser alvo de desenvolvimentos futuros.

9.1 Cumprimento dos objetivos definidos

Para a implementação deste projeto definimos com principal motivação a necessidade de uma aplicação capaz de tornar possível a integração semântica entre sistemas heterogéneos e que não partilhem qualquer forma de identificação de recursos.

O projeto contou ainda com um conjunto de três metas indispensáveis ao seu sucesso: adaptabilidade a mudanças nos *datasets*, fácil manutenção, permitindo que o sistema seja mantido em produção fora de um ambiente fechado, e finalmente precisão (ou rigor) nos resultados calculados.

Terminada a fase de implementação, podemos concluir que a solução proposta abrange cada um destes objetivos.

Em primeiro lugar, foi dado grande destaque à implementação de um mecanismo de classificação facilmente adaptável a mudanças de âmbito do *dataset*. Deste esforço resultou um conjunto de regras baseadas em padrões abertos, tais como XML ou OWL.

Por forma a tornar o sistema facilmente administrável, as regras acima definidas foram implementadas por forma a potenciar a independência da aplicação pelo seu código fonte. Por outras palavras, tentámos manter a lógica da aplicação fora do código-fonte desenvolvido.

O último dos requisitos apresentados (precisão dos resultados obtidos) foi satisfatoriamente atingido, tal como descrito no capítulo 8. O nosso principal objetivo durante a implementação do motor de pontuação foi o desenvolvimento de um mecanismo capaz de fornecer resultados absolutos, pois só assim será possível avaliar até que ponto os resultados são ou não satisfatórios.

9.2 Principais limitações

A principal limitação sentida durante o desenvolvimento deste trabalho foi sem dúvida a restrição imposta ao nível dos recursos humanos disponíveis. Sendo este um trabalho individual, sentiu-se por vezes alguma incapacidade em agilizar tarefas facilmente paralelizáveis. Isto levou naturalmente a um custo acrescido no desenvolvimento do projeto.

Na sequência dos fatores referidos acima, as restrições ao nível do prazo de entrega do projeto obrigaram-nos também a um esforço adicional, impedindo mesmo que nos alongássemos mais em alguns dos temas expostos.

Por fim, resta referir alguma dificuldade em lidar com os *datasets*. Enquanto que os dados que compõem a BD central estão completos e devidamente formatados, os dados provenientes da BD do POS contém erros ortográficos, falhas de formação (tais como problemas de *encoding*, etc.). O processo de normalização destes dados revelou-se mais demorado do que seria à partida espetável.

9.3 Contribuição

A principal contribuição deste projeto materializa-se na aplicação desenvolvida, a qual cumpre os objetivos apresentados.

Em suma, é disponibilizado um mecanismo de integração de sistemas, altamente adaptável aos *datasets* utilizados e personalizável à medida das necessidades.

Este mecanismo baseia-se em técnicas de integração semântica tais como a análise léxico-gramatical ou a construção de ontologias, garantindo assim altos níveis de fiabilidade sobre os resultados.

Esta implementação abre portas para outros sistemas muito mais completos, cuja principal

base passa pela integração semântica de sistemas. Esperamos desta forma poder contribuir para que a investigação científica e tecnológica evolua no sentido de demonstrar a viabilidade destes mesmos sistemas.

9.4 Trabalho futuro

Uma vez que estamos a lidar com uma aplicação recetiva a mudanças nos *datasets*, será necessário prever também a normalização dos mesmos.

Muito por culpa das restrições ao nível dos recursos e do tempo disponíveis, acabou por ser executada uma normalização manual dos *datasets*. Apesar de funcional, esta abordagem limita claramente a escalabilidade da aplicação, assim como a sua adaptabilidade a novos *datasets*.

Posto isto, a implementação de um mecanismo de normalização de produtos seria um grande passo no sentido de tornar a aplicação verdadeiramente aberta a novos *datasets*.

Este motor deve, nomeadamente, implementar as funções de um corretor ortográfico, sendo capaz de detetar palavras ou frases incorretas, e interagindo com o utilizador a fim de corrigir de forma definitiva este mesmo termo.

A aplicação das técnicas utilizadas deve, em fases futuras, ser analisada de forma mais cuidada, garantindo um controlo mais rígido sobre os resultados obtidos.

Uma das sugestões seria a aplicação gradual das diversas técnicas a grupos de produtos aos quais a sua aplicabilidade fosse garantida. Este é um processo bastante mais exigente ao nível da sua implementação, no entanto permitir-nos-ia ter garantias de que técnicas de manipulação lexical como o *stemming* não introduziriam ruído nos resultados gerados.

O algoritmo de pontuação desenvolvido poderá também vir a ser alvo de melhoramentos, uma vez que a sua implementação representa, por si só, um esforço demasiado exigente para o âmbito deste projeto.

Bibliografia

- [1] Yan, X., Peng, Y., Meng, J., Ruzante, J., Fratamico, P., Huang, L., Juneja, V., Needleman, D.: From Ontology Selection and Semantic Web to an Integrated Information System for Food-borne Diseases and Food Safety (2009). In: Software Tools and Algorithms for Biological Systems
- [2] Hendler, J.: Web 3.0: The dawn of semantic search (2010). pp: 77–80.
- [3] Menzel, J.: Deeper understanding with Metaweb. [Online] Acedido em Setembro de 2012: <http://googleblog.blogspot.pt/2010/07/deeper-understanding-with-metaweb.html>
- [4] Lopez, X., Das, S.: Semantic Technologies in Oracle Database 11g Release 2: Capabilities, Interfaces, Performance (2009).
- [5] Boulos, M.: Semantic Wikis: A Comprehensible Introduction with Examples from the Health Sciences (2009). In: Journal of Emerging Technologies in Web Intelligence, Vol. 1, No. 1
- [6] Goel, K.: Slice and dice your recipe search results. [Online] Acedido em Setembro de 2012: <http://googleblog.blogspot.pt/2011/02/slice-and-dice-your-recipe-search.html>
- [7] Beckwith, R., Miller, A., Teng, R.: Design and Implementation of the WordNet Lexical Database and Searching Software. Description of WordNet (1993).
- [8] International Organization for Standardization: ISO/IEC 9075-1:2011: Information technology -- Database languages -- SQL -- Part 1: Framework (2011).

- [9] MySQL Test Labs: A Look at MySQL 5.0 Performance Benchmarks (2006).
- [10] Object Management Group: OMG Unified Modeling Language (OMG UML), Infrastructure, version 2.4.1 (2011).
- [11] International Organization for Standardization: Maintenance agency for ISO 3166 country codes. [online] Acedido em Setembro de 2012: http://www.iso.org/iso/country_codes.htm
- [12] World Wide Web Consortium: Extensible Markup Language (XML) 1.0 (Fifth Edition) (2008).
- [13] International Organization for Standardization: ISO/IEC 14977: Extended Backus-Naur Form (1996).
- [14] World Wide Web Consortium: OWL 2 Web Ontology Language Document Overview (2009).
- [15] World Wide Web Consortium: RDF/XML Syntax Specification (Revised) (2009).
- [16] Obitko, M.: Semantic Web Architecture (2007). [Online] Acedido em Setembro de 2012: <http://obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html>
- [17] University of Manchester: The OWL API Website. [Online] Acedido em Setembro de 2012: <http://owlapi.sourceforge.net/index.html>
- [18] Apache Software Foundation: Jena Website. [Online] Acedido em Setembro de 2012: <http://jena.apache.org/>
- [19] Standord University: Protégé Website. [Online] Acedido em Setembro de 2012: <http://protege.stanford.edu/>
- [20] Porter, M.: “An algorithm for suffix stripping”. In: Program, 14(3) pp 130-137 (1980)
- [21] V., Orengo.: “Assessing Relevance Using Automatically Translated Documents for Cross-Language Information Retrieval”. In: School of Computing Science,

Middlesex University, London, UK. p. 258 PhD Thesis (2004)

- [22] Savoy, J.: “Stemming of French words based on grammatical categories”. In: Journal of the American Society for Information Science, 44: 1-9 (1993)
- [23] Savoy, J.: “Light stemming approaches for the French, Portuguese, German and Hungarian languages”. In: Proceedings of the 2006 ACM symposium on Applied computing (SAC 2006) ISBN 1-59593-108-2 (2006)
- [24] Apache Software Foundation: Apache Lucene Website. [Online] Acedido em Setembro de 2012: <http://lucene.apache.org/>
- [25] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II (2002). In: IEEE Transactions on Evolutionary Computation 6, no. 2, pp: 182.197.
- [26] Durillo, J., Nebro, A., Alba, E.: The jMetal Framework for Multi-Objective Optimization: Design and Architecture (2010). In: CEC 2010, pp: 4138-4325.

Anexos

Anexo A

Pressupostos

Este anexo especifica as alterações consumadas no âmbito dos pressupostos identificados durante a fase de estudo e adaptação dos recursos disponíveis.

Produtos fora do contexto do projeto

Carne - Porco Cabeça s/osso	Restaurante Dose de Legumes
Colaborador SAS - Almoço	Restaurante Dose Peixe
Colaborador SAS - Jantar	Restaurante Dose Peixe Alternativo
Convidado FS	Restaurante Dose Peixe Convidado
Fruta Prato Fruta Variada (bar)	Restaurante Prato de Pão
Iogurte Pedacos Magro 125g	Senha Almoço Carne
Menu	Senha Almoço Carne
Mercearia - Açúcar Amarelo KG	Senha Almoço Carne Bar
Mercearia - Açúcar Branco KG	Senha Almoço Carne Externo
Mercearia - Massa Espirais Tricolor	Senha Almoço Peixe
Mercearia - Massa Laços	Senha Almoço Peixe Bar
Nali - Acendalhas	Senha Almoço Peixe Externo
Nali - Baldes c/ espremedor	Senha Funcionario Almoço Carne
Nali - Esfergao Bravo	Senha Funcionario Almoço Carne
Nali - Palitos Individuais	Senha Funcionario Almoço Peixe
Nali - Papel anti-aderente 40x60cm	Senha Funcionario Jantar Carne
Nali - Pas do Lixo c/cabo	Senha Funcionario Jantar Peixe
Nali - Resma Papel Vegetal Coz. 50x75c	Senha Funcionario Vegetariana Almoço
Nali - Rodos c/esponja 45cm c/cabo	Senha Funcionario Vegetariano Jantar
Nali - Rolo Papel Chemin?	Senha Jantar
Nali - Sacos cristal 1lts	Senha Jantar Carne
Nali - Sacos pretos p/lixo 30lts	Senha Jantar Carne Bar
Nali - Saquetas	Senha Jantar Carne Externo
Nali - Toalhas de Mao Tissue	Senha Jantar Peixe
Nali - Toalhas Mesa 0.81x1.20mt	Senha Jantar Peixe Bar
Nali - Toalhetes de Mesa 30x40cm	Senha Jantar Peixe Externo
Prato Almoço Carne S/B	Senha Pré-Comprada
Prato Almoço Peixe S/B	Senha Pré-Comprada
Prato Almoço Salgados Misto	Senha R.Servida Almoço Carne
Prato Almoço Salgados Misto	Senha R.Servida Almoço Carne Convidado
Prato Funcionario	Senha R.Servida Almoço Peixe
Prato Funcionario Vegetariano Almoço	Senha R.Servida Almoço Peixe Convidado
Prato Funcionario Vegetariano Jantar	Senha R.Servida Almoço Vegetariana
Prato Jantar Carne S/B	Senha R.Servida Jantar Carne
Prato Jantar Peixe S/B	Senha R.Servida Jantar Carne Convidado
Prato Jantar Salgados Misto	Senha R.Servida Jantar Peixe
Prato Vegetariano Almoço R/R	Senha R.Servida Jantar Peixe Convidado
Prato Vegetariano Almoço R/R	Senha R.Servida Jantar Vegetariana
Prato Vegetariano Jantar R/R	Senha R.Servida Veg. Almoço Convidado

R.S.Carne	Senha R.Servida Veg.Jantar Convidado
R.S.Peixe	Senha Vegetariana Almoço
Restaurante 1/2 Dose Carne	Senha Vegetariana Jantar
Restaurante 1/2 Dose Carne Alternativo	Senha Vegetariana Pre-Comprada Almoço
Restaurante 1/2 Dose Carne Convidado	Senha Vegetariana Pre-Comprada Jantar
Restaurante 1/2 Dose Peixe	Senha Vegetariano Almoço Externo
Restaurante 1/2 Dose Peixe Alternativo	Senha Vegetariano Jantar Externo
Restaurante 1/2 Dose Peixe Convidado	Temperos - Açafrão
Restaurante Dose Carne	Temperos - Molho Piri-Piri Maçarico
Restaurante Dose Carne Alternativo	Restaurante Dose Carne Convidado

Nomes de produtos corrigidos por estarem truncados

1/2 Dose Lombo à Transm. c/ Arroz e Legumes	Legumes - Espinafres Pte cat.II (molho)
Bebidas - Vinho Lavradores 0.75cl Tinto	Mercearia - Cogumelo Inteiro Lata Grande
Bebidas - Vinho Tapada H. C. 075cl Tinto	Mercearia - Salsicha Cachorro 8 Unidades
Bebidas - Vinho Qt.Aveledada 0.75cl Branco	Mercearia - Salsicha Frankfurt 25/pares
Congelados - Batata Pre-F. Redondas Rodelas	Rebuçados - Halls Vitamina C s/ Açúcar
Dose Lombo à Transm. c/ Arroz, Bat.e Legumes	Sobremesas - Bolo de Bolacha Tradicional
Gelado - Cornetto Love Chocolate	Vinho Qt.Aveledada 0.75cl Branco
Lact - Iogurte Sveltesse Aromas/Natural	

Termos corrigidos por apresentarem problemas na representação de caracteres especiais

Carne - Porco Osso Espinhaço	Rebuçados - Halls Vitamina C s/ Açúcar
Peixe - Pescada nº1	Sandes Presunto c/Queijo (Carça)
Peixe - Solha Posta nº3	

Termos corrigidos por apresentarem erros ortográficos

Bolachas de Água e Sal	Folhados - Merendas Triangulares
Bebidas - Chá de Tília	Folhado Merendas Triangulares
Bebidas - Macieira	Gelado - Cornetto Choco Disc
Bubbaloo	Gelado Cornetto Choco Disc
Bubblicious Frutis	Rissóis Camarão
Bubblicious Mentol	Rissóis Carne
Bubblicious Morango	Rissóis Pescada
Chiclets	Snack - Fatia Pizza (bar)
Chiclets Ice	Snacks - Piza Baguete Fiambre
Dose Costeletas de Porco Grelhadas	Temperos - Maionese 1.800Grs
Folhado Mini Piza	Vgt-Piza Queijo e Fiambre 1,550KG

Termos corrigidos por apresentarem nomes ou códigos de países não normalizados

Fruta - Clementinas PT/ES cat.II	Legumes - Cenoura PT/ES cat.II
Fruta - Kiwi 33/36 PT cat.I	Legumes - Coentros PT cat.II
Fruta - Laranja 5/6 PT cat.II	Legumes - Couve Flor PT cat.II
Fruta - Laranja 7/8 PT cat.II	Legumes - Couve Lombarda PT cat.II
Fruta - Limao PT/ES cat.II	Legumes - Couve Roxa PT cat.II
Fruta - Pera Rocha 55/60 PT cat.II	Legumes - Espinafres PT cat.II (molho)
Fruta - Pera Rocha 60/70 PT cat.II	Legumes - Espinafres PT cat.II (saco)
Legumes - Aipo PT cat.II	Legumes - Feijao Verde PT/MA cat.II
Legumes - Alface Frisada PT cat.II	Legumes - Pepino PT cat.II
Legumes - Alface Roxa PT cat.II	Legumes - Pimento Amarelo ES cat.II
Legumes - Alho Frances PT cat.II	Legumes - Pimento Verde PT/ES cat.II
Legumes - Azeitona 5/6 PT	Legumes - Pimento Vermelho ES cat.II
Legumes - Beringela PT/ES cat.II	Legumes - Salsa PT cat.II
Legumes - Broculos PT/ES cat.II	Legumes - Tomate Cherry ES. cat.II
Legumes - Cabeça Nabo PT cat.II	Legumes - Tomate PT/ES cat.II
Legumes - Cebola 80/110 PT/ES cat.II	

Categorias da tabela central totalmente removidas

Não Alcoólicas, Água -	Alcoólicas Fermentadas -
Não Alcoólicas,	Café (Infusão) -
Alcoólicas Destiladas -	Café Infusão -

Categorias da tabela de POS totalmente removidas

Bebidas	Padaria
Carne	Pastilhas
Charcutaria	Peixe
Fruta	Restaurante
Gelado	Salgados
Ingrediente	Snack
Lact	Snacks
Legumes	Sobremesas
Leguminosas	Temperos
Mercearia	

Categorias da tabela de POS parcialmente removidas

Pastelaria	Rebuçados
Folhados	

Anexo B

Regras para normalização automática de termos

Este anexo especifica as regras elaboradas com vista a fornecer à aplicação uma base para que esta possa executar a correção e normalização automática de termos. Aos conjuntos de regras aqui apresentados é dado o nome de *data import filters*.

10-cleaning.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:ruleset xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ns0='http://xml.semanticsearch.pt/schema/data_import_filters'
  xsi:schemaLocation='http://xml.semanticsearch.pt/schema/data_import_filters model.xsd'
  priority="10"
  name="cleaning"
  description="It stores all the deleting actions that should be done
  immediately after the data import in order to remove all the
  meaningless words and expressions we could identify"
  version="0.1a">

  <ns0:rule description="Empty brackets">
    <ns0:search>\(\s*\)</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>

  <ns0:rule description="Products ending with 'Bar'">
    <ns0:search>\(?:Bar\)?(\s)*$</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>

  <ns0:rule description="Products ending with 'Gr'">
    <ns0:search>((/|\s)Gr[.]?) (\s)*$</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>

  <ns0:rule description="Products starting with 'Restaurante'">
    <ns0:search>^\(?:Restaurante\s\)?</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>

  <ns0:rule description="N parts of something">
    <ns0:search>\(?:\d\s?partes\)?</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>

  <ns0:rule description="Medium value (1st part)">
    <ns0:search>\((valor(es)?\sm(e|é)dio)s?.*\)\*?</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>

  <ns0:rule description="Medium value (2nd part)">
    <ns0:search>\b(valor(es)?\sm(e|é)dio)s?\b</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>
</ns0:ruleset>
```

```

</ns0:rule>

<ns0:rule description="Asterisks">
  <ns0:search>\*</ns0:search>
  <ns0:replace></ns0:replace>
</ns0:rule>

<ns0:rule description="Quotes">
  <ns0:search>\"</ns0:search>
  <ns0:replace></ns0:replace>
</ns0:rule>

<ns0:rule description="N varieties of something">
  <ns0:search>\b\d\s?variedades?\b</ns0:search>
  <ns0:replace></ns0:replace>
</ns0:rule>

<ns0:rule description="Words ending or beginning with special chars">
  <ns0:search>^\s*[/\\\:.\-]\s*|\s*[/\\\:.\-]\s*$</ns0:search>
  <ns0:replace></ns0:replace>
</ns0:rule>

<ns0:rule description="Loose special chars">
  <ns0:search>\s+[/\\\:.\-]\s+</ns0:search>
  <ns0:replace> </ns0:replace>
</ns0:rule>

<ns0:rule description="Products ending with 'R.P.'">
  <ns0:search>R(\?.)P(\.)?</ns0:search>
  <ns0:replace></ns0:replace>
</ns0:rule>
</ns0:ruleset>

```

20-abbreviations.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<ns0:ruleset xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ns0='http://xml.semanticsearch.pt/schema/data_inport_filters'
  xsi:schemaLocation='http://xml.semanticsearch.pt/schema/data_inport_filters model.xsd'
  priority="20"
  name="abbreviations"
  description="It stores all the replacing actions that should be done
  immediately after the data import in order to remove all the
  abbreviations we could identify"
  version="0.2c">

  <ns0:rule>
    <ns0:search>\bc/(\s)*</ns0:search>
    <ns0:replace>com </ns0:replace>
  </ns0:rule>

  <ns0:rule>
    <ns0:search>\bs/(\s)*\b</ns0:search>
    <ns0:replace>sem </ns0:replace>
  </ns0:rule>

```

```

<ns0:rule>
  <ns0:search>\bp/(\s)*\b</ns0:search>
  <ns0:replace>para </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>bat\.\s*</ns0:search>
  <ns0:replace>batata </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\bcen[/\.]?\s*\b</ns0:search>
  <ns0:replace>cenoura </ns0:replace>
</ns0:rule>

  <ns0:rule>
    <ns0:search>\bpres[/\.]?\s*\b</ns0:search>
    <ns0:replace>presunto </ns0:replace>
  </ns0:rule>

<ns0:rule>
  <ns0:search>\balf[/\.]?\s*\b</ns0:search>
  <ns0:replace>alface </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\bve?ge?t(\.|\s|-)\s*\b</ns0:search>
  <ns0:replace>vegetariano </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\bpr[eé][\s-]Fr?i?t?.?\s*</ns0:search>
  <ns0:replace>pré-fritas </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\btransm\.\s*</ns0:search>
  <ns0:replace>transmontana </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\btradicion\.\b\s*</ns0:search>
  <ns0:replace>tradicional </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(biol?[oó]?g?.?\s*)\b</ns0:search>
  <ns0:replace>biológico </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(cong\.\s*)\b</ns0:search>
  <ns0:replace>congelado </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(q(ueij)?\.\s*)\b</ns0:search>
  <ns0:replace>queijo </ns0:replace>
</ns0:rule>

```

```

<ns0:rule>
  <ns0:search>\b(qu/\s*)\b</ns0:search>
  <ns0:replace>queijo </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(chouri[çc][ãa]o/\s*)\b</ns0:search>
  <ns0:replace>chourição </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(alf(ac)?\.\?\s*)\b</ns0:search>
  <ns0:replace>alface </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(cen\.\?\s*)\b</ns0:search>
  <ns0:replace>cenoura </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(mant[/\.\.]\s*)\b</ns0:search>
  <ns0:replace>manteiga </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(coe(1h)?\.\?\s*)\b</ns0:search>
  <ns0:replace>coelho </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(h(erd)?\.\?\s*)\b</ns0:search>
  <ns0:replace>herdade </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(m[-\/\s]gordo)\s*\b</ns0:search>
  <ns0:replace>meio-gordo </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(g\.\?\s*)\b</ns0:search>
  <ns0:replace>grande </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(p(eq)?\.\?\s*)\b</ns0:search>
  <ns0:replace>pequeno </ns0:replace>
</ns0:rule>

<ns0:rule>
  <ns0:search>\b(med\s*)\b</ns0:search>
  <ns0:replace>médio </ns0:replace>
</ns0:rule>
</ns0:ruleset>

```

30-cleaning.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:ruleset xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ns0='http://xml.semanticsearch.pt/schema/data_import_filters'
  xsi:schemaLocation='http://xml.semanticsearch.pt/schema/data_import_filters model.xsd'
  priority="30"
  name="cleaning"
  description="It stores all the replacing actions that should be done
  after every other removing or replacing action in order to remove any
  meaningless term that could be originated"
  version="0.1">

  <ns0:rule description="Empty brackets">
    <ns0:search>\(\s*\)</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>

  <ns0:rule description="double spaces">
    <ns0:search>\s{2,}</ns0:search>
    <ns0:replace> </ns0:replace>
  </ns0:rule>

  <ns0:rule description="Word trimming">
    <ns0:search>^\s+|\s+$</ns0:search>
    <ns0:replace></ns0:replace>
  </ns0:rule>
</ns0:ruleset>
```

Anexo C

Dicionários elaborados

Este anexo apresenta os dicionários elaborados para efeitos de classificação lexical e gramatical. Dada a extensão destes ficheiros, apenas o dicionário “Receitas” será fisicamente incluído nos anexos do documento. Os restantes dicionários estarão disponíveis apenas em formato digital.

receitas.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns0:ruleset xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ns0='http://xml.semanticsearch.pt/schema/dictionary'
  xsi:schemaLocation='http://xml.semanticsearch.pt/schema/dictionary
model.xsd'
  grammaticalMeaning="receita"
  name="Receitas"
  version="3.0b"
  description="Every single word that acts like a recipe by itself"
  isValid="true"
  priorToVerification="true">
  <ns0:rule id="10" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(a[çc]orda)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
      <ns0:contains>
        <ns0:dicName>Produtos</ns0:dicName>
        <ns0:ruleID>6640</ns0:ruleID>
      </ns0:contains>
    </ns0:owldata>
  </ns0:rule>
  <ns0:rule id="20" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(br[áa]s)\b</ns0:search>
    <ns0:prerequisite>(frango|bacalhau|gambas)</ns0:prerequisite>
    <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning/>
    <ns0:owldata>
      <ns0:contains>
        <ns0:dicName>Produtos</ns0:dicName>
        <ns0:ruleID>4850</ns0:ruleID>
      </ns0:contains>
      <ns0:contains>
        <ns0:dicName>Produtos</ns0:dicName>
        <ns0:ruleID>1570</ns0:ruleID>
      </ns0:contains>
    </ns0:owldata>
  </ns0:rule>
```

```

    <ns0:rule id="30" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(gomes\sde\ss[áa])\b</ns0:search>
    <ns0:prerequisite>(bacalhau)</ns0:prerequisite>
    <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning/>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>4850</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1570</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1540</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

    <ns0:rule id="40" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(caldeirada)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1570</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1540</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

    <ns0:rule id="50" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(burger)\b</ns0:search>
    <ns0:prerequisite>\b(feij([áa]o|[ó]es))\b</ns0:prerequisite>
    <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning/>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1730</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>7820</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>

```

```

        <ns0:ruleID>4730</ns0:ruleID>
    </ns0:contains>
</ns0:owldata>
</ns0:rule>

<ns0:rule id="60" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(feijoada)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1730</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>3820</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>3670</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>3690</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>3720</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="70" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(passarinho)\b</ns0:search>
    <ns0:prerequisite>(frango)</ns0:prerequisite>
    <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning/>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>3830</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Caracteristicas</ns0:dicName>
            <ns0:ruleID>370</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="80" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(jardineira)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>

```

```

        <ns0:owldata>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>1570</ns0:ruleID>
            </ns0:contains>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>1550</ns0:ruleID>
            </ns0:contains>
        </ns0:owldata>
    </ns0:rule>

    <ns0:rule id="90" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
        <ns0:search>\b(lasanha)\b</ns0:search>
        <ns0:prerequisite/>
        <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
        <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
        <ns0:owldata>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>7770</ns0:ruleID>
            </ns0:contains>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>3800</ns0:ruleID>
            </ns0:contains>
        </ns0:owldata>
    </ns0:rule>

    <ns0:rule id="100" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
        <ns0:search>\b(transmontan[oa])\b</ns0:search>
        <ns0:prerequisite>\b(lombo)\b</ns0:prerequisite>
        <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
        <ns0:grammaticalMeaning/>
        <ns0:owldata>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>3954</ns0:ruleID>
            </ns0:contains>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>6540</ns0:ruleID>
            </ns0:contains>
        </ns0:owldata>
    </ns0:rule>

    <ns0:rule id="110" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
        <ns0:search>\b(b[ée]chamel)\b</ns0:search>
        <ns0:prerequisite>\b(molho)\b</ns0:prerequisite>
        <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
        <ns0:grammaticalMeaning/>
        <ns0:owldata>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>5930</ns0:ruleID>
            </ns0:contains>
        </ns0:owldata>
    </ns0:rule>

```

```

        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>2670</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>2660</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="120" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(om[oe]leta)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>4760</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>4630</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="130" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(pizz?a)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>2720</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>4630</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="140" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(pur[ée])\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1570</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

```

```

        <ns0:dicName>Produtos</ns0:dicName>
        <ns0:ruleID>2660</ns0:ruleID>
    </ns0:contains>
</ns0:owldata>
</ns0:rule>

<ns0:rule id="150" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(quiche(\s*Lorraine)?)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>5930</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>2680</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>2720</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="160" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(rabanadas?)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>6640</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>2660</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>4760</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="170" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(salada\s*russa)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1570</ns0:ruleID>

```

```

        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1610</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>7820</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="180" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(sopas?)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1570</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>2830</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="190" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(lavrador)\b</ns0:search>
    <ns0:prerequisite>\b(sopas?)\b</ns0:prerequisite>
    <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning/>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1570</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1730</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>4580</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="200" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(caldo\s*verde)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>

```

```

        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1950</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1570</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="210" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(canjas?)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>3830</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>7770</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="220" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(juliana)\b</ns0:search>
    <ns0:prerequisite>\b(sopas?)\b</ns0:prerequisite>
    <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning/>
    <ns0:owldata>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1570</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1950</ns0:ruleID>
        </ns0:contains>
        <ns0:contains>
            <ns0:dicName>Produtos</ns0:dicName>
            <ns0:ruleID>1550</ns0:ruleID>
        </ns0:contains>
    </ns0:owldata>
</ns0:rule>

<ns0:rule id="230" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
    <ns0:search>\b(tapiocas?)\b</ns0:search>
    <ns0:prerequisite/>
    <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:owldata>
        <ns0:contains>

```

```

                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>1880</ns0:ruleID>
            </ns0:contains>
        </ns0:owldata>
    </ns0:rule>

    <ns0:rule id="240" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
        <ns0:search>\b(tofu)\b</ns0:search>
        <ns0:prerequisite/>
        <ns0:lexicalMeaning>Receita tipo produto</ns0:lexicalMeaning>
        <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
        <ns0:owldata>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>1690</ns0:ruleID>
            </ns0:contains>
        </ns0:owldata>
    </ns0:rule>

    <ns0:rule id="250" isValid="true" weight="100" priorTo="0"
description="" isCaseSensitive="false">
        <ns0:search>\b(tomatada)\b</ns0:search>
        <ns0:prerequisite/>
        <ns0:lexicalMeaning>Receita tipo
caracteristica</ns0:lexicalMeaning>
        <ns0:grammaticalMeaning/>
        <ns0:owldata>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>1710</ns0:ruleID>
            </ns0:contains>
            <ns0:contains>
                <ns0:dicName>Produtos</ns0:dicName>
                <ns0:ruleID>1540</ns0:ruleID>
            </ns0:contains>
            <ns0:contains>
                <ns0:dicName>Caracteristicas</ns0:dicName>
                <ns0:ruleID>350</ns0:ruleID>
            </ns0:contains>
        </ns0:owldata>
    </ns0:rule>
</ns0:ruleset>

```

model.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://xml.semanticsearch.pt/schema/dictionary"
elementFormDefault="qualified">
    <xs:element name="ruleset">
        <xs:complexType>
            <!-- RULE ELEMENTS AND ATTRIBUTES -->

```

```

<xs:sequence>
  <xs:element name="rule" maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <!-- The regex to be matched -->
        <xs:element name="search" type="xs:string"
minOccurs="1" maxOccurs="1"/>
        <!-- Some prerequisites that should also be
matched -->
        <!-- When there are more than one prerequisite,
all of than should be verified (AND operation) -->
        <xs:element name="prerequisite" type="xs:string"
minOccurs="0" maxOccurs="unbounded"/>
        <!-- The token's lexical meaning -->
        <xs:element name="lexicalMeaning"
type="xs:string" minOccurs="0" maxOccurs="1"/>
        <!-- The token's grammatical meaning. If set, it
will override the global value -->
        <xs:element name="grammaticalMeaning"
type="xs:string" minOccurs="0" maxOccurs="1"/>
        <!-- Ontology axioms -->
        <xs:element name="owldata" minOccurs="0"
maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="subClassOf"
minOccurs="0" maxOccurs="1">
                <xs:complexType>
                  <xs:sequence>
                    <!-- Dictionary name for the
reference. If empty, the current dictionary name will be considered -->
                    <xs:element name="dicName"
type="xs:string" minOccurs="0" maxOccurs="1"/>
                    <!-- The rule ID we are
pointing to -->
                    <xs:element name="ruleID"
type="xs:positiveInteger" minOccurs="1" maxOccurs="1"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="sameAs" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <!-- Dictionary name for the
reference. If empty, the current dictionary name will be considered -->
                    <xs:element name="dicName"
type="xs:string" minOccurs="0" maxOccurs="1"/>
                    <!-- The rule ID we are
pointing to -->
                    <xs:element name="ruleID"
type="xs:positiveInteger" minOccurs="1" maxOccurs="1"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="contains" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>

```

```

        <!-- Dictionary name for the
reference. If empty, the current dictionary name will be considered -->
        <xs:element name="dicName"
type="xs:string" minOccurs="0" maxOccurs="1"/>
        <!-- The rule ID we are
pointing to -->
        <xs:element name="ruleID"
type="xs:positiveInteger" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>

    <!-- The rule's unique ID -->
    <xs:attribute name="id" type="xs:positiveInteger"
use="required"/>

    <!-- Should the rule be considered? (Non valid rules
won't be called by the application) -->
    <xs:attribute name="isValid" type="xs:boolean"
use="required"/>

    <!-- The rule's weight (it should stand between 0 and
100) -->
    <xs:attribute name="weight" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="100"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>

    <!-- If the rule should be interpreted prior to an
other, these rule ID should be referred here -->
    <!-- Only one level of precedence will be considered
-->
    <xs:attribute name="priorTo"
type="xs:nonNegativeInteger" use="optional"/>
    <!-- The rule's human-readable description -->
    <xs:attribute name="description" type="xs:string"
use="optional"/>

    <!-- Should the algorithm be case sensitive for this
rule? -->
    <xs:attribute name="isCaseSensitive"
type="xs:boolean" use="required"/>

    </xs:complexType>
</xs:element>
</xs:sequence>

    <!-- RULESET ATTRIBUTES -->

    <!-- The ruleset name -->
    <xs:attribute name="name" type="xs:Name" use="required"/>
    <!-- The ruleset version -->
    <xs:attribute name="version" type="xs:string"
use="required"/>
    <!-- The ruleset description -->

```

```
        <xs:attribute name="description" type="xs:string"
use="required"/>
        <!-- The rulesets's grammatical meaning. It will be applied
to every rule -->
        <xs:attribute name="grammaticalMeaning" type="xs:string"
use="required"/>
        <!-- Should the rule be considered? (Non valid rules won't be
called by the application) -->
        <xs:attribute name="isValid" type="xs:boolean"
use="required"/>
        <!-- Should the "prior to" verification be performed? -->
        <xs:attribute name="priorToVerification" type="xs:boolean"
use="required"/>

        </xs:complexType>
    </xs:element>

</xs:schema>
```

Anexo D

Regras de gramática relacional

Este anexo apresenta as regras de gramática relacional elaboradas para efeitos de classificação gramatical da aplicação.

grammar_rules.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:ruleset xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:ns0='http://xml.semanticsearch.pt/schema/lex_rules'
  xsi:schemaLocation='http://xml.semanticsearch.pt/schema/lex_rules
model.xsd'
  name="RelationalGrammarRules"
  version="0.2b"
  description="Grammar rules that allow us to interconnect the tokens
of a term">

  <ns0:rule id="5" isValid="true" weight="50" description="Atributos">
    <ns0:search>\b(de)\b</ns0:search>
    <ns0:grammaticalMeaning>atributo de produto
</ns0:grammaticalMeaning>      <!-- It clones the the last token's
grammatical meaning -->
    <ns0:refersTo>!#produto&lt;1</ns0:refersTo>
<!-- It refers to the last "produto" we found -->
    <ns0:appliesTo>!#>1</ns0:appliesTo>
<!-- It applies to the next token -->
  </ns0:rule>

  <ns0:rule id="10" isValid="true" weight="50" description="Conjunção">
    <ns0:search>\b(com\s*adi[çç][ãa]o\s*(de)?|com)\b</ns0:search>
    <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
    <ns0:appliesTo>!#>1</ns0:appliesTo>      <!-- It applies to
the next token -->
  </ns0:rule>

  <ns0:rule id="20" isValid="true" weight="50" description="Disjunção">
    <ns0:search>\b(sem\s*adi[çç][ãa]o\s*(de)?|sem)\b</ns0:search>
    <ns0:grammaticalMeaning>produto (a
retirar)</ns0:grammaticalMeaning>
    <ns0:appliesTo>!#>1</ns0:appliesTo>      <!-- It applies to
the next token -->
  </ns0:rule>

  <ns0:rule id="30" isValid="true" weight="50"
description="Exclusividade">
    <ns0:search>\b(s[óo](\s[ao]s?)?|apenas|somente)\b</ns0:search>
    <ns0:grammaticalMeaning>!#produto%1</ns0:grammaticalMeaning>
    <!-- It will identify the next token as the main "produto" -->
    <ns0:appliesTo>!#>1</ns0:appliesTo>
  </ns0:rule>

  <ns0:rule id="40" isValid="true" weight="50" description="Indicador
de receita">
    <ns0:search>\b(à)\b</ns0:search>
```

```

        <ns0:grammaticalMeaning>receita</ns0:grammaticalMeaning>
        <ns0:refersTo>!#produto<1</ns0:refersTo> <!-- It refers to
the last "produto" we found -->
        <ns0:appliesTo>!#>1</ns0:appliesTo> <!-- It applies to
the next token -->
        </ns0:rule>

        <ns0:rule id="50" isValid="true" weight="50" description="Indicador
de sabor">
        <ns0:search>\b((com\s*)?sabor(\s*a?))\b</ns0:search>
        <ns0:grammaticalMeaning>sabor</ns0:grammaticalMeaning>
        <ns0:refersTo>!#produto<1</ns0:refersTo> <!-- It refers to
the last "produto" we found -->
        <ns0:appliesTo>!#>1</ns0:appliesTo> <!-- It applies to
the next token -->
        </ns0:rule>

        <ns0:rule id="60" isValid="true" weight="50" description="Indicador
de tipo">
        <ns0:search>\b(tipo)\b</ns0:search>
        <ns0:grammaticalMeaning>tipo de produto</ns0:grammaticalMeaning>
        <ns0:refersTo>!#produto<1</ns0:refersTo> <!-- It refers to
the last "produto" we found -->
        <ns0:appliesTo>!#>1</ns0:appliesTo> <!-- It applies to
the next token -->
        </ns0:rule>

        <ns0:rule id="70" isValid="true" weight="50" description="Indicador
de finalidade">
        <ns0:search>\b(para)\b</ns0:search>
        <ns0:grammaticalMeaning>finalidade</ns0:grammaticalMeaning>
        <ns0:refersTo>!#produto<1</ns0:refersTo> <!-- It refers to
the last "produto" we found -->
        <ns0:appliesTo>!#>1</ns0:appliesTo> <!-- It applies to
the next token -->
        </ns0:rule>

        <ns0:rule id="80" isValid="true" weight="50"
description="Enumeração">
        <ns0:search>\,</ns0:search>
        <ns0:grammaticalMeaning>!#<1</ns0:grammaticalMeaning> <!-- It
clones the the last token's gramatical meaning -->
        <ns0:appliesTo>!#>1</ns0:appliesTo>
<!-- It applies to the next token -->
        </ns0:rule>

        <ns0:rule id="90" isValid="true" weight="50"
description="Enumeração">
        <ns0:search>\b(e|ou)\b</ns0:search>
        <ns0:grammaticalMeaning>!#<1</ns0:grammaticalMeaning> <!-- It
clones the the last token's gramatical meaning -->
        <ns0:appliesTo>!#>1</ns0:appliesTo>
<!-- It applies to the next token -->
        </ns0:rule>

        <ns0:rule id="100" isValid="true" weight="50"
description="Enumeração">
        <ns0:search>\b((alto|baixo)[-\\s]teor(\s*(de|em)))\b</ns0:search>
        <ns0:grammaticalMeaning>produto</ns0:grammaticalMeaning>
        <ns0:appliesTo>!#>1</ns0:appliesTo>

```

```

<!-- It applies to the next token -->
  </ns0:rule>
</ns0:ruleset>

```

model.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://xml.semanticsearch.pt/schema/lex_rules"
  elementFormDefault="qualified">
  <xs:element name="ruleset">
    <xs:complexType>
      <!-- RULE ELEMENTS AND ATTRIBUTES -->
      <xs:sequence>
        <xs:element name="rule" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <!-- The regex to be matched -->
              <xs:element name="search" type="xs:string"
minOccurs="1" maxOccurs="1"/>
              <!-- The grammatical meaning imposed by the
current rule -->
              <xs:element name="grammaticalMeaning"
type="xs:string" minOccurs="1" maxOccurs="1"/>
              <!-- The token that will be affected by the
grammatical meaning -->
              <!-- Please refer to the documentation if you
want to check the symbols you can use here -->
              <xs:element name="appliesTo" type="xs:string"
minOccurs="1" maxOccurs="1"/>
              <!-- When dealing with an attribute, we need to
specify the token to which it refers -->
              <!-- Please refer to the documentation if you
want to check the symbols you can use here -->
              <xs:element name="refersTo" type="xs:string"
minOccurs="0" maxOccurs="1"/>
            </xs:sequence>
            <!-- The rule's unique ID -->
            <xs:attribute name="id" type="xs:positiveInteger"
use="required"/>
            <!-- Should the rule be considered? (Non valid rules
won't be called by the application) -->
            <xs:attribute name="isValid" type="xs:boolean"
use="required"/>
            <!-- The rule's weight (it should stand between 0 and
100) -->
            <xs:attribute name="weight" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:integer">

```

```

        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="100"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<!-- The rule's human-readable description -->
<xs:attribute name="description" type="xs:string"
use="required"/>

    </xs:complexType>
</xs:element>
</xs:sequence>

<!-- RULESET ATTRIBUTES -->

<!-- The ruleset name -->
<xs:attribute name="name" type="xs:Name" use="required"/>
<!-- The ruleset version -->
<xs:attribute name="version" type="xs:string"
use="required"/>
<!-- The ruleset description -->
<xs:attribute name="description" type="xs:string"
use="required"/>

    </xs:complexType>
</xs:element>

</xs:schema>

```

Anexo E

Relações estáticas definidas

Este anexo apresenta as relações estáticas manualmente definidas por forma a permitir uma avaliação dos resultados obtidos pela aplicação.

Produto POS	Produto BD central
1/2 Dose Costoletas de Porco Grelhadas	Porco Costeleta gorda grelhada
1/2 Dose Entrecosto Grelhado	Porco Entrecosto grelhado
1/2 Dose Frango Grelhado	Frango Inteiro com pele grelhado
7 Up Lata	Bebida Refrigerante gasosa
7Up	Bebida Refrigerante gasosa
Abacaxi	Ananás
Abacaxi (bar)	Ananás
Abatanado	bica (3 marcas)
Abobora PT cat.II	Abóbora crua
Abrotea de tomatada c/arroz branco	Abrótea cozida
Abrotea Posta	Abrótea crua
Açucar Amarelo KG	Açúcar amarelo
Açucar Delta	Açúcar branco
Agriao	Agrião cru
Água 0.25L/Gr	Água mineral natural, "Luso"
Água 0.33L	Água mineral natural, "Luso"
Água 0.50L	Água mineral natural, "Luso"
Água 0.50L/Gr.	Água mineral natural, "Luso"
Água 1.5L	Água mineral natural, "Luso"
Água 1L Gr	Água mineral natural, "Luso"
Água c/ Gás	Água mineral natural gaseificada, "Pizões-Moura"
Água c/Gás Aroma	Água mineral natural gaseificada, "Vimeiro"
Agua Caramulo 0.33L	Água mineral natural, "Luso"
Agua Caramulo 0.50L	Água mineral natural, "Luso"
Agua Caramulo 1.5 L	Água mineral natural, "Luso"
Água Carvalhos 0.25L/Gr	Água mineral natural, "Luso"
Água Carvalhos 0.50L/Gr	Água mineral natural, "Luso"
Água Carvalhos 1L/Gr	Água mineral natural, "Luso"
Água Carvalhos OW 0.25L	Água mineral natural, "Luso"
Água Serrana Garrafao 5 L	Água mineral natural, "Luso"
Aipo PT cat.II	Aipo cru
Alface	Alface crua
Alface Frisada PT cat.II	Alface crua
Alface Roxa PT cat.II	Alface crua
Alho cru	Alho cru
Alho Frances PT cat.II	Alho francês cru
Alhos Mor cat.II	Alho cru

Alhos Saco
Almofadas Mistas Forte
Almofadas Mistas Mini
Almofadinha Mista
Ameijoa 50/70
Ameixa
Amendoa c/pele
Ananás
Ananas CR 5/7 cat.II
Arroz Agulha
Arroz Carolino 5Kg
Arroz Doce
Arroz Doce (bar)
Atum Lata Grande
Atum Lata Pequena
Azeite Galheteiro
Azeitona 5/6 PT
Bacalhau a Gomes de Sa

Bacalhau Graudo
Bacalhau Grelhado
Bacalhau Posta
Banana (Bar)
Banana Cavendish cat.I
Banana Estrangeira
Barra Bounty
Barra Dove
Barra Maltesers
Barra Mars
Barra Snickers
Barra Twix
Batata Frita
Batata Frita (Palha)
Batata Frita 45grs
Batata Miuda
Batata Palha
Batatas
Batatas Fritas Artesanais
Beringela PT/ES cat.II
Beterraba Roxa
Bife de Vaca Grelhado

Bolacha Agua Sal
Bolacha Belga
Bolacha ChipMix
Bolacha Chips Ahoy

Alho cru
Pastel folhado
Pastel folhado
Pastel folhado
Amêijoa crua
Ameixa branca
Amêndoa, miolo, com pele
Ananás
Ananás
Arroz comum cru
Arroz carolino branqueado cru
Arroz doce
Arroz doce
Atum conserva em óleo
Atum conserva em óleo
Azeite (4 marcas)
Azeitona
Bacalhau Seco e salgado, demolido à Gomes de Sá
Bacalhau Fresco cru
Bacalhau Seco e salgado, demolido grelhado
Bacalhau Fresco cru
Banana
Banana
Banana
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Batata frita caseira (em palitos)
Batata frita caseira (em palitos)
Batata frita caseira (em palitos)
Batata crua
Batata frita caseira (em palitos)
Batata crua
Batata frita caseira (em palitos)
Beringela crua
Beterraba (raiz) crua
Vaca Bife (valor médio de acém, alcatra e lombo) grelhado
Bolacha água e sal
Bolacha "Belga"
Bolacha chocolate
Bolacha chocolate

Bolacha Maria
Bolacha Principe
Bolacha Tuc
Bolachas Belgas
Bolachas Chipmix
Bolachas Chips Ahoy
Bolachas de Água e Sal
Bolachas Principe
Bolachas Tuc
Bolinha de Carne Mini
Bollycao
Bolo Bolacha
Bolo Bolacha (bar)
Bolo Brigadeiro
Bolo Chocolate
Bolo de Bolacha
Bolo de Bolacha Tradicional
Borrego Perna
Bounty
Brigadeiro
Brigadeiro (bar)
Broa de Milho
Broculos PT/ES cat.II
Bubbaloo
Bubblicious Frutis
Bubblicious Menta
Bubblicious Mentol
Bubblicious Morango
Cabeça Nabo PT catII
Cacete
Cafe
Calippo
Camarao 30/40
Camarao 60/80
Canela Moida
Canela Moida Frasco
Canela Moida Saqueta
Canela Pau
Carapau Medio
Carioca
Carne Vaca Bife Vacuo

Carne Vaca Hamburger
Carte D'or
Cebola 80/110 PT/ES cat.II

Bolacha Maria
Bolacha chocolate
Bolacha manteiga
Bolacha "Belga"
Bolacha chocolate
Bolacha chocolate
Bolacha água e sal
Bolacha chocolate
Bolacha manteiga
Croquete
Bolo de chocolate
Bolo de bolacha Maria
Bolo de bolacha Maria
Bolo de chocolate
Bolo de chocolate
Bolo de bolacha Maria
Bolo de bolacha Maria
Borrego, Costeleta ou Perna crua
Chocolate de leite
Bolo de chocolate
Bolo de chocolate
Pão de milho
Brócolos crus
Pastilha elástica
Pastilha elástica
Pastilha elástica
Pastilha elástica
Pastilha elástica
Nabo (raiz) cru
Pão de centeio
bica (3 marcas)
Gelado de água (sorvete)
Camarão cru
Camarão cru
Canela moída
Canela moída
Canela moída
Canela moída
Carapau cru
carioca (2 marcas)
Vaca Bife (valor médio de acém, alcatra e lombo) cru
Hamburger de vaca, cru
Gelado de leite
Cebola crua

Cebola 90+ Cat
Cenoura
Cenoura PT/ES cat.II
Cerejas
Cerejas (bar)
Cerejas Cristalizadas
Cerveja Sagres Branca
Cerveja Sagres Preta
Cerveja Super Bock Branca
Cerveja Super Bock Preta
Cerveja SuperBock Branca
Cerveja SuperBock Preta
Cha Camomila
Cha Cidreira
Cha de Tília
Cha Earl Grey
Cha Ervas c/Mel
Cha Menta
Cha Preto
Cha Tília
Cha Verde
Cha Verde c/Menta
Cha Vermelho
Cha Vermelho c/Laranja
Chamuça Forte
Chamuça Mini
Chamuças
Chavena de Leite
Cherne Grelhado
Chiclets
Chiclets Ice
Choco c/tinta 100/120
Choco c/tinta 50/100
Choco limpo
Chocolate - Bounty
Chocolate - Cadbury
Chocolate - Crunch
Chocolate - Crunchie 40grs
Chocolate - Dove
Chocolate - Dove Caramelo
Chocolate - Kinder 4 Barras
Chocolate - Kinder 8 Barras
Chocolate - Kinder Bueno
Chocolate - Kinder Delice
Chocolate - Kit-Kat

Cebola crua
Cenoura crua
Cenoura crua
Cereja (4 variedades)
Cereja (4 variedades)
Cereja, cristalizada
Cerveja branca
Cerveja preta
Cerveja branca
Cerveja preta
Cerveja branca
Cerveja preta
Chá, infusão, de ervas
Chá, infusão, de ervas
Chá, infusão, de ervas
Chá, infusão, preto
Chá, infusão, de ervas
Chá, infusão, de ervas
Chá, infusão, preto
Chá, infusão, preto
Chá, infusão, de ervas
Chá, infusão, verde
Chá, infusão, verde
Chá, infusão, de ervas
Chá, infusão, de ervas
Chamuça
Chamuça
Chamuça
Leite Vaca UHT gordo
Cherne grelhado
Pastilha elástica
Pastilha elástica
Choco cru
Choco cru
Choco cru
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite

Chocolate - Kit-Kat Crunchie
Chocolate - Lion
Chocolate - M&M's
Chocolate - M&M's Peanuts
Chocolate - Maltesers
Chocolate - Maltesers Branco
Chocolate - Mars
Chocolate - Mars Delight
Chocolate - Mars Miniatura
Chocolate - Nestle 20grs
Chocolate - Nestle 50grs
Chocolate - Nestle Leite/Amendoas
Chocolate - Nuts
Chocolate - Snickers
Chocolate - Snickers Crunch
Chocolate - Toblerone
Chocolate - Twix
Chocolate Culinaria Barra
Chocolate Kinder Fiesta
Chocolate Kinder Schoko-Bons
Chocolate Kit Kat Chunky
Chocolate Milk Mini 40Gr
Chocolate Po
Chocolate Regina c/Amendoas 24Gr
Chocolate Snack de Leite
Chocolate Toffee Crisp
Chocos Grelhados
Chouriço Caseiro
Chouriço Corrente
Clementinas PT/ES cat.II
Coca Cola Gr
Coca Cola Lata
Coca-Cola Gr
Coentros PT cat.II
Cogumelo Inteiro Lata Grande
Cogumelo Laminada
Cogumelo Laminado 780Gr
Compal Alperce Pacote
Compal Alperce Pacote 200ml
Compal Ananas Pacote
Compal Ananas Pacote 200ml
Compota Pessego 20G
Congelados - Batata Pre-F. Redondas
Rodelas
Congelados - Brocolos

Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate em barra, culinária
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate de leite
Chocolate em pó
Chocolate de leite
Chocolate de leite
Chocolate de leite
Choco grelhado
Chouriço de sangue cru
Chouriço de sangue cru
Clementina
Bebida Refrigerante cola
Bebida Refrigerante cola
Bebida Refrigerante cola
Coentros crus
Cogumelos enlatados, escorridos
Cogumelos crus
Cogumelos enlatados, escorridos
Néctar alperce
Néctar alperce
Néctar ananás
Néctar ananás
Compota Damasco
Batata frita, de pacote (em rodelas)
Brócolos crus

Congelados - Cenoura	Cenoura crua
Congelados - Couve Bruxelas	Couve de Bruxelas crua
Congelados - Couve Flor	Couve-flor crua
Congelados - Espinafres	Espinafres crus
Congelados - Milho	Milho, grão seco cru
Copo Cafe (Grande)	bica (3 marcas)
Copo Cafe (Pequeno)	bica (3 marcas)
Copo Leite Grande	Leite Vaca UHT gordo
Copo Leite Simples	Leite Vaca UHT gordo
Cornetto	Gelado de leite
Cornetto Choco Disc	Gelado de leite
Cornetto Love Chocolate	Gelado de leite
Corvina s/ Cabeça	Corvina crua
Costeleta Grelhada	Porco Costeleta gorda grelhada
Couve Corte PT cat.II	Couve portuguesa crua
Couve Flor PT cat.II	Couve-flor crua
Couve Lombarda PT cat.II	Couve lombarda crua
Couve Roxa PT cat.II	Couve roxa crua
Croissant	Croissant
Croissant c/ Fiambre	Croissant
Croissant c/ Manteiga	Croissant
Croissant c/ Queijo	Croissant
Croissant c/ Queijo e Manteiga	Croissant
Croissant Chocolate	Croissant
Croissant Misto	Croissant
Croissant Misto c/ Manteiga	Croissant
Croissant Ovo	Croissant
Croissant Simples	Croissant
Croissants 7 Days Chocolate	Croissant
Croquetes	Croquete
Croquetes Carne Forte	Croquete
Croquetes Carne Mini	Croquete
Descafeinado	Café solúvel (pó) descafeinado (2 marcas)
Digestivo Aldeia Velha	Aguardente
Digestivo Macieira	Brandy
Digestivo Martini	Gin - Rum- Whisky
Digestivo Porto	Vinho generoso do Porto, doce
Digestivo Whisky Velho 12 Anos	Gin - Rum- Whisky
Donuts c/ Recheio	"Donut" recheado com doce de fruta
Donuts C/Recheio	"Donut" recheado com doce de fruta
Donuts Simples	"Donut"
Donuts Simples Glace	"Donut"
Dose Bacalhau Cozido/Grelhado	Bacalhau Fresco cozido
Dose Bife de Vaca Grelhado	Vaca Bife (valor médio de acém, alcatra e lombo) grelhado

Dose Cherne Cozido/Grelhado	Cherne cozido
Dose Choquinhos	Choco grelhado
Dose Corvina Cozida/Grelhada	Corvina cozida
Dose Costeletas de Porco Grelhadas	Porco Costeleta gorda grelhada
Dose Entrecosto Grelhado	Porco Entrecosto grelhado
Dose Frango Grelhado	Frango Inteiro com pele grelhado
Dose Lombo à Transm. c/ Arroz, Bat.e Legumes	Vitela Lombo assado com azeite e margarina
Dose Maruca Cozida/Grelhada	Maruca cozida
Dose Pescada no Forno c/ Batata	Pescada (valor médio)* cozida
Empada Atum	Empada
Empada Carne	Empada
Empada Frango	Empada
Empada Vegetariana	Empada
Empadinha	Empada
Empanadilha Frango	Empada
Entrecosto Grelhado	Porco Entrecosto grelhado
Epá	Gelado de leite
Ervilha	Ervilhas secas cruas
Espinafres PT cat.II (molho)	Espinafres crus
Espinafres PT cat.II (saco)	Espinafres crus
Farinha	Farinha de trigo tipo 150
Farinha Milho Branca	Farinha de milho tipo 70
Farinheira	Farinheira crua
Fatia Piza (bar)	Pizza de queijo e tomate
Feijao Branco	Feijão branco cru
Feijao Branco Lata	Feijão branco cru
Feijao Frade	Feijão frade (feijão miúdo) cru
Feijao Frade Lata	Feijão frade (feijão miúdo) cru
Feijao Manteiga	Feijão manteiga cru
Feijao Manteiga Lata	Feijão manteiga cru
Feijao Verde PT/MA cat.II	Feijão verde fresco cru
Fermento Po Grande	Fermento em pó
Fermento Po Pequeno	Fermento em pó
Fiambre	Fiambre
Filetes Pescada	Pescada (valor médio)* frita
Flocos Cereais Mel	Flocos de cereais e frutos secos tipo "Muesli"
Folhado Carne	Pastel folhado
Folhado Frango	Pastel folhado
Folhado Merendas Triangulares	Pastel folhado
Folhado Misto Queijo/Carne	Pastel folhado
Folhado Misto Queijo/Fiambre	Pastel folhado
Folhado Misto Queijo/Salsicha	Pastel folhado
Folhado Queijo/Salsicha	Pastel folhado
Folhado Vegetariano (Bar)	Pastel folhado

Folhado Vegetarianos
Folhados - Carne
Folhados - Frango
Folhados - Merendas Triangulares
Folhados - Miniatura Doce
Folhados - Queijo/Carne
Folhados - Queijo/Fiambre
Folhados - Queijo/Salsicha
Frango a Passarinho
Frango C/Miudos (4 Partes)
Frango da Quintinha
Frango Fresco
Frango Grelhado
Frango Peito
Frango Peito Congelado
Frango Perna
Frango Perna Congelada
Frango S/Miudos (4 Partes)
Galinha Grande
Gelatina Ananas Carte Dor
Gelatina Ananas Nestle
Gelatina Folhas Branca
Grao Bico
Halls Eucalipto
Halls Extra Forte
Halls Mel/Limao
Halls Mentol S/Açucar
Halls Mentol/Eucalipto
Halls S/Açucar
Halls Vitamina C
Halls Vitamina C s/ Açucar
Halls Vitamina C S/Açucar
Hamburguer
Iogurte Aromas

Iogurte Aromas 125g

Iogurte Aromas Magro 125g
Iogurte Bifidus Liquido
Iogurte Copo Vidro c/Fruta

Iogurte Liquido Aromas 185ml
Iogurte Liquido Magro 185ml
Iogurte Natural
Iogurte Pedações 125g
Iogurte Pedações Fruta

Pastel folhado
Pastel folhado
Pastel folhado
Pastel folhado
Pastel folhado
Pastel folhado
Pastel folhado
Pastel folhado
Frango Inteiro com pele assado, sem molho
Frango Inteiro com pele cru
Frango, Pele crua
Frango Inteiro com pele cru
Frango Inteiro com pele grelhado
Frango, Pele crua
Frango Peito com pele cru
Frango (1/4 de Frango) Perna com pele crua
Frango (1/4 de Frango) Perna com pele crua
Frango, Pele crua
Frango Inteiro com pele cru
Gelatina desidratada (pó ou folha)
Gelatina desidratada (pó ou folha)
Gelatina desidratada (pó ou folha)
Grão-de-bico cru
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Rebuçados
Hamburger de porco, grelhado
Iogurte Aromatizado açucarado batido meio gordo
Iogurte Aromatizado açucarado batido meio gordo
Iogurte Aromatizado açucarado sólido magro
Iogurte Aromatizado açucarado líquido magro
Iogurte Aromatizado açucarado sólido meio gordo
Iogurte Açucarado líquido meio gordo
Iogurte Aromatizado açucarado líquido magro
Iogurte Natural sólido meio gordo
Iogurte Açucarado batido gordo com fruta
Iogurte Açucarado batido gordo com fruta

Iogurte Pedacos Magro 125g	Iogurte Natural sólido magro
Iogurte Suissinho Frutas	Iogurte Açucarado batido meio gordo com fruta
Iogurte Sveltesse Aromas/Natural	Iogurte Aromatizado açucarado sólido magro
Iogurte Sveltesse Liquido	Iogurte Aromatizado açucarado líquido magro
Iogurte Yoggi Aromas/Summer	Iogurte Aromatizado açucarado líquido meio gordo
Ketchup Top Down	Molho de tomate, "Ketchup"
Kiwi 23/25 cat.II	Kiwi
Kiwi 33/36 PT cat.I	Kiwi
Laranja	Laranja (3 variedades)
Laranja (bar)	Laranja (3 variedades)
Laranja 5/6 PT cat.II	Laranja (3 variedades)
Laranja 7/8 PT cat.II	Laranja (3 variedades)
Leite Achocolatado 200 ml	Leite Vaca Esterilizado achocolatado meio gordo
Leite Achocolatado UCAL Gfa 250ml	Leite Vaca Esterilizado achocolatado meio gordo
Leite Amanhecer UHT M/Gordo LT	Leite Vaca UHT meio gordo
Leite c/ Chocolate (Pacote)	Leite Vaca Esterilizado achocolatado meio gordo
Leite c/Chocolate Gr	Leite Vaca Esterilizado achocolatado meio gordo
Leite Creme (bar)	Leite creme
Leite Creme Alsa	Leite creme
Leite Creme Royal	Leite creme
Leite Gresso Chocolate 0.20	Leite Vaca Esterilizado achocolatado meio gordo
Leite Leziria UHT M/Gordo LT	Leite Vaca UHT meio gordo
Leite UHT M/Gordo Gresso L	Leite Vaca UHT meio gordo
Limao PT/ES cat.II	Limão
Linguiça	Linguiça
Lombos Medalhoes Pescada	Pescada (valor médio)* crua
Lombos Tamboril 500/100 Brasil	Tamboril cru
Lula Limpa	Lula crua
Lula Limpa 20/40	Lula crua
Maça	Maça com casca
Maça Assada	Maça assada com açúcar
Maça Golden 70/75 PT cat.II	Maça com casca
Maça Golden 75/80 PT cat.II	Maça com casca
Maça Golden 80+ cat.II	Maça com casca
Maça Jongold 75/80 PT cat.II	Maça com casca
Maça Raineta 70/80 PT cat.II	Maça com casca
Maça Royal FR 70/75cat.II	Maça com casca
Maça Royal Gala 70/75 cat.II	Maça com casca
Maça Royal GF 75/80cat.I	Maça com casca

Maça Stark 70/75 cat.II
Maça Stark 75/80 cat.II
Macieira
Maionese
Maionese 1.800Grs
Maionese Frasco Grande
Maionese Top Down
Manga
Manga (bar)
Manga BR
Manteiga
Manteiga C/sal 10grs
Manteiga Grosso c/sal
Manteiga Mimosa c/ sal
Manteiga Vaca c/sal 250g R.P.
Martini
Maruca
Maruca cozida c/batata e legumes
Massa Lasanha
Melancia
Melão
Melão (bar)
Melo
Melo (bar)
Mentos Fruta
Mentos Max Air
Mentos Mentol
Mini - Croissants
Mini Piza
Minimilk
Morangos
Morangos (bar)
Morcela
Morcela de Arroz
Mousse Chocolate
Mousse Chocolate (bar)
Mousse Chocolate Alsa
Mousse Chocolate Carte Dor
Mousse Chocolate Nestle
Natas UHT 1LT
Oleo 10 Litros
Orbit
Orbit Melao
Ovo - Clara
Ovo - Cozido Balde

Maça com casca
Maça com casca
Brandy
Maionese caseira, com ovo e azeite
Maionese caseira, com ovo e azeite
Maionese caseira, com ovo e azeite
Maionese caseira, com ovo e azeite
Manga
Manga
Manga
Manteiga com sal
Manteiga com sal
Manteiga com sal
Manteiga com sal
Manteiga com sal
Gin - Rum- Whisky
Maruca crua
Maruca cozida
Massa para lasanha crua
Melancia
Melão (3 variedades)
Melão (3 variedades)
Melo
Melo
Rebuçados
Rebuçados
Rebuçados
Croissant
Pizza de queijo e tomate
Gelado de leite
Morango
Morango
Morcela crua
Morcela crua
Mousse de chocolate
Mousse de chocolate
Mousse de chocolate
Mousse de chocolate
Mousse de chocolate
Nata não maturada UHT 35% gordura
Óleo alimentar
Pastilha elástica
Pastilha elástica
Ovo (de galinha) clara crua
Ovo (de galinha) cozido

Ovo - Cozido Individual	Ovo (de galinha) cozido
Ovo - Gema	Ovo (de galinha) gema crua
Ovo - Inteiro	Ovo (de galinha) inteiro cru
Paio Lombo	Paio de lombo
Panadinhos Pescada KG	Pescada (valor médio)* frita
Pão	Pão de trigo
Pao Bolas	Pão de trigo
Pao c/Manteiga	Pão de trigo
Pao Cachorro	Pão de trigo
Pao Centeio	Pão de centeio
Pao Embalado (Mistura 40g)	Pão de mistura de trigo e centeio
Pão Fatia	Pão de trigo
Pao Leite	Pão de leite (trigo)
Pao Leite c/ Fiambre	Pão de leite (trigo)
Pao Leite c/ Manteiga	Pão de leite (trigo)
Pao Leite c/ Queijo	Pão de leite (trigo)
Pao Leite Mistto	Pão de leite (trigo)
Pao Leite Simples	Pão de leite (trigo)
Pao Mistura 1000g	Pão de mistura de trigo e centeio
Pao Ralado	Pão ralado
Pao Simples	Pão de trigo
Papaia	Papaia
Papaia (bar)	Papaia
Papaia cat II	Papaia
Pasteis Bacalhau	Pastel de bacalhau
Pastéis de Bacalhau	Pastel de bacalhau
Pastel Nata Grande	Pastel de nata
Pastilhas Orbit	Pastilha elástica
Pastilhas Trex S/Açucar	Pastilha elástica
Pataniscas Bacalhau	Pastel de bacalhau
Pato	Pato com pele cru
Peixe Espada Branco Posta	Peixe-espada-branco cru
Pepino PT cat.II	Pepino cru
Pepsi Lata	Bebida Refrigerante cola
Pera	Pêra (5 variedades)
Pera (Bar)	Pêra (5 variedades)
Pera Rocha 55/60 PT cat.II	Pêra (5 variedades)
Pera Rocha 60/70 PT cat.II	Pêra (5 variedades)
Perna Frango no forno c/batata assada	Frango (1/4 de Frango) Perna com pele cozida
Perna Pau	Gelado de leite
Perna Pau Mega	Gelado de leite
Peru Peito	Peru Peito com pele cru
Pescada	Pescada (valor médio)* crua
Pescada frita c/salada russa	Pescada (valor médio)* frita
Pescada nº1	Pescada (valor médio)* crua

Pêssego	Pêssego (2 variedades)
Pessego Calda Grande	Pêssego, conserva em calda de açúcar
Pessego Calda Pequena	Pêssego, conserva em calda de açúcar
Pimenta Moida Branca	Pimenta moída
Pimento Amarelo ES cat.II	Pimento cru
Pimento Verde PT/ES cat.II	Pimento cru
Pimento Vermelho ES cat.II	Pimento cru
Piza Baguete Fiambre	Pizza de queijo, tomate e fiambre
Polpa Manga	Manga
Polvo 800/1200	Polvo cru
Porco Costeletas do Cachaço	Porco Costeleta gorda crua
Porco Costeletas do Lombo	Porco Costeleta gorda crua
Porco Costeletas do Pe	Porco Costeleta gorda crua
Porco Entrecosto	Porco Entrecosto cru
Porco Lombo	Porco Lombo cru
Porco Lombo Recheado	Porco Lombo cru
Porco Perna c/osso	Porco Perna gorda crua
Porco Perna s/osso	Porco Perna gorda crua
Presunto s/osso	Presunto
Pudim Caseiro	Pudim flan caseiro
Pudim Flan Alsa	Pudim flan caseiro
Queijo 22grs	Queijo Flamengo 30% gordura
Queijo Fresco	Queijo Serra fresco
Queijo Queru	Queijo Creme para barrar
Raia s/pele	Raia crua
Rebucado Fruta	Rebuçados
Rebucado Mentol	Rebuçados
Rebucado Solano	Rebuçados
Rebuçados - Chupas	Rebuçados
Rebuçados - De Fruta	Rebuçados
Rebuçados - De Mentol	Rebuçados
Red Fish	Cantarilho (Redfish) cru
Rissóis	Rissol
Rissóis Camarão	Rissol de camarão
Rissóis Carne	Rissol
Rissóis de Peixe	Rissol
Rissóis Pescada	Rissol
Rissol Camarao	Rissol de camarão
Rissol Camarao Forte	Rissol de camarão
Rissol Carne Forte	Rissol
Rissol Mini	Rissol
Rissol Pescada	Rissol
Rissol Pescada Forte	Rissol
Sal	Sal
Salame	Salame

Salgado - Empadinha
Salmão
Salmao Grelhado c/batata e legumes
Salmao grelhado c/batatas e cenoura
Salmao Posta
Salpicao
Salsa PT cat.II
Salsicha Cachorro 8 Unidades
Salsicha Churrasco
Salsicha Frankfurt 25/pares
Salsicha Frankfurt 5/pares
Salsicha Fresca
Salsicha P/Cachorro
Sardinha
Semi-Frio Bolacha
Semi-Frio Brigadeiro
Semi-Frio Morango (bar)
Smints
Solha à delicia c/puré
Solha Posta
Solha Posta nº3
Sopa Cebola
Sprite Gr
Sprite Lata
Sumo Fresco Laranja
Sumo Fresco Laranja 25cl
Sumo Fresco Maçã
SuperMaxi
Supremo Maça (bar)
Tamboril Inteiro
Tarte de Maça
Tarte Maça
Tomate Cherry ES cat.II
Tomate PT/ES cat.II
Tosta Fiambre
Tosta Queijo
Trex
Trident Advantage
Trident Fresh
Trident Fresh Sticks
Trident Fruit
Trident Kids
Trident Max Air
Trident Splash
Trident White

Empada
Salmão cru
Salmão grelhado
Salmão grelhado
Salmão cru
Salpicão
Salsa crua
Salsicha tipo Frankfurt
Salsicha fresca crua
Salsicha tipo Frankfurt
Salsicha tipo Frankfurt
Salsicha fresca crua
Salsicha tipo Frankfurt
Sardinha gorda crua
Bolo de bolacha Maria
Bolo de chocolate
Doce Morango
Rebuçados
Solha grelhada
Solha crua
Solha crua
Sopa cebola
Bebida Refrigerante gasosa
Bebida Refrigerante gasosa
Sumo de frutos 100%, laranja
Sumo de frutos 100%, laranja
Sumo de frutos 100%, maçã
Gelado de leite
Doce Maçã
Tamboril cru
Tarte de maçã
Tarte de maçã
Tomate cru
Tomate cru
Tosta de trigo simples
Tosta de trigo simples
Pastilha elástica
Pastilha elástica
Pastilha elástica
Pastilha elástica
Pastilha elástica
Pastilha elástica
Pastilha elástica
Pastilha elástica
Pastilha elástica

Trident White Spearmint
Ucal

Vaca Almondegas
Vaca Bife Alcatra

Vaca Bife Pojadouro

Vaca Bife Rabadilha

Vaca Bife Vazia

Vaca Hamburger

Vaca P/Cozer

Vaca P/Estufar (cubos)

Veget-Almondegas 22grs

Vegetarianos - Almondegas

Vegetarianos - Tofu Bio

Vgt-Piza Queijo e Fiambre 1,550KG

Vinagre Branco Galheteiro

Vinagre Tinto Galheteiro

Vinho Borba 0.75cl Tinto

Vinho Branco Garrafao

Vinho Casa 0.250cl Tinto

Vinho Casa 0.5Lt Tinto

Vinho Casa 1Lt Tinto

Vinho Casa Copo Tinto

Vinho do Porto

Vinho Esteva 0.75cl Tinto

Vinho Lavradores 0.75cl Tinto

Vinho M.Borba 0.375cl Tinto

Vinho M.Borba 0.75cl Tinto

Vinho M.Velho 0.75cl Branco

Vinho M.Velho 0.75cl Tinto

Vinho Marques Borba 0.375cl Tinto

Vinho Marques Borba 0.75cl Tinto

Vinho Montado 0.375cl Branco

Vinho Montado 0.75cl Tinto

Vinho Montado 375cl Tinto

Vinho Monte Velho 0.75cl Branco

Vinho Monte Velho 0.75cl Tinto

Vinho Olho Mocho Reserva 2006

Vinho Pias 0.75cl Branco

Vinho Planalto 0.75cl Branco

Vinho Qt.Aveledada 0.75cl Branco

Vinho Qt.Carmo 0.75cl Branco

Pastilha elástica

Leite Vaca Esterilizado achocolatado meio gordo

Almôndega cozinhada

Vaca Bife (valor médio de acém, alcatra e lombo) cru

Vaca Bife (valor médio de acém, alcatra e lombo) cru

Vaca Bife (valor médio de acém, alcatra e lombo) cru

Vaca Bife (valor médio de acém, alcatra e lombo) cru

Hamburger de vaca, cru

Vaca para Cozer ou Estufar meio gorda crua

Vaca para Cozer ou Estufar meio gorda crua

Almôndega cozinhada

Almôndega cozinhada

Tofu simples

Pizza de queijo, tomate e fiambre

Vinagre

Vinagre

Vinho maduro tinto

Vinho maduro branco

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro tinto

Vinho generoso do Porto, doce

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro branco

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro branco

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro branco

Vinho maduro tinto

Vinho maduro tinto

Vinho maduro branco

Vinho maduro branco

Vinho maduro branco

Vinho maduro branco

Vinho Qt.Carmo 0.75cl Tinto	Vinho maduro tinto
Vinho Qt.Portal 0.75cl Tinto	Vinho maduro tinto
Vinho Quinta Aveledada 0.75cl Branco	Vinho maduro branco
Vinho Quinta Carmo 0.75cl Tinto	Vinho maduro tinto
Vinho Quinta Carmo Branco 075cl	Vinho maduro branco
Vinho Quinta Portal 0.75cl Tinto	Vinho maduro tinto
Vinho Tapada H. C. 075cl Tinto	Vinho maduro tinto
Vinho Tapada Herd. Coe. 0.75cl Tinto	Vinho maduro tinto
Vinho Tinto Garrafao	Vinho maduro tinto
Whisky Velho 12 Anos	Gin - Rum- Whisky

Anexo F

Resultados do processo de otimização

Este anexo apresenta os melhores resultados obtidos em cada uma das execuções do algoritmo de otimização de pesos. Estes surgem ordenados por ordem decrescente, ou seja, do melhor para o pior resultado.

Experiência 1				Experiência 2				Experiência 3			
Acert os	Falh as	Desv io	% Falha	Acert os	Falh as	Desv io	% Falha	Acert os	Falh as	Desv io	% Falha
361	254	35,95	41,30	345	270	36,98	43,90	338	277	38,44	45,04
359	256	36,05	41,63	342	273	37,70	44,39	335	280	38,54	45,53
356	259	37,06	42,11	340	275	38,77	44,72	332	283	40,19	46,02
354	261	38,07	42,44	336	279	39,00	45,37	328	287	40,34	46,67
352	263	38,18	42,76	332	283	40,06	46,02	323	292	42,17	47,48
347	268	38,27	43,58	329	286	40,38	46,50	320	295	43,52	47,97
340	275	38,3	44,72	328	287	40,45	46,67	317	298	43,74	48,46
338	277	38,50	45,04	324	291	41,86	47,32	316	299	43,80	48,62
336	279	39,25	45,37	321	294	43,63	47,80	313	302	44,87	49,11
333	282	39,55	45,85	317	298	45,07	48,46	311	304	45,29	49,43
329	286	40,27	46,50	315	300	46,69	48,78	310	305	45,94	49,59
326	289	41,25	46,99	310	305	47,64	49,59	306	309	47,70	50,24
323	292	41,98	47,48	304	311	49,50	50,57	305	310	47,70	50,41
319	296	42,88	48,13	302	313	50,29	50,89	301	314	50,52	51,06
316	299	45,35	48,62	300	315	52,52	51,22	299	316	51,33	51,38
310	305	47,67	49,59	298	317	53,08	51,54	298	317	51,53	51,54
307	308	49,03	50,08	275	340	53,29	55,28	295	320	52,31	52,03
303	312	50,81	50,73	274	341	53,61	55,45	276	339	52,48	55,12
286	329	50,93	53,50	270	345	53,83	56,10	274	341	53,44	55,45
278	337	52,51	54,80	266	349	56,60	56,75	273	342	54,03	55,61
275	340	53,75	55,28	264	351	56,83	57,07	271	344	54,95	55,93
271	344	54,31	55,93	263	352	58,19	57,24	263	352	57,24	57,24
269	346	56,03	56,26	258	357	59,74	58,05	253	362	57,30	58,86
268	347	56,24	56,42	256	359	59,78	58,37	251	364	58,70	59,19
263	352	57,19	57,24	254	361	60,68	58,70	249	366	59,46	59,51
262	353	58,14	57,40	250	365	60,78	59,35	247	368	60,97	59,84
257	358	58,31	58,21	246	369	63,47	60,00	245	370	61,59	60,16
252	363	58,56	59,02	244	371	64,35	60,33	244	371	62,33	60,33
249	366	59,64	59,51	243	372	65,82	60,49	241	374	64,76	60,81
247	368	61,21	59,84	238	377	66,10	61,30	239	376	65,68	61,14
244	371	62,84	60,33	236	379	67,38	61,63	237	378	66,56	61,46
243	372	65,18	60,49	230	385	68,25	62,60	234	381	67,98	61,95
235	380	66,77	61,79	223	392	70,50	63,74	233	382	68,30	62,11
234	381	67,66	61,95	222	393	74,66	63,90	225	390	69,09	63,41
229	386	69,28	62,76	219	396	87,47	64,39	223	392	70,03	63,74

227	388	69,52	63,09	216	399	89,22	64,88	222	393	70,78	63,90
223	392	72,71	63,74	211	404	91,64	65,69	221	394	74,23	64,07
220	395	74,38	64,23	209	406	92,21	66,02	218	397	75,89	64,55
212	403	80,39	65,53	198	417	95,18	67,80	205	410	89,92	66,67
211	404	80,62	65,69	195	420	98,96	68,29	204	411	95,00	66,83
						100,1					
205	410	95,18	66,67	191	424	5	68,94	192	423	96,88	68,78
						103,5				100,7	
193	422	96,20	68,62	189	426	4	69,27	190	425	3	69,11
						108,4				105,8	
191	424	97,36	68,94	188	427	8	69,43	189	426	6	69,27
		102,0				122,7				108,6	
190	425	7	69,11	182	433	1	70,41	188	427	8	69,43
		103,9				126,7				124,4	
189	426	5	69,27	175	440	5	71,54	187	428	4	69,59
		105,9				127,1				124,5	
188	427	7	69,43	174	441	1	71,71	182	433	3	70,41
		115,0				132,8				127,0	
187	428	6	69,59	172	443	0	72,03	177	438	4	71,22
		128,1				133,3				127,7	
181	434	0	70,57	171	444	3	72,20	174	441	8	71,71
		128,6				159,9				128,6	
171	444	7	72,20	170	445	4	72,36	171	444	4	72,20
		135,6				163,1				130,6	
170	445	5	72,36	169	446	3	72,52	170	445	2	72,36