Universidade de Aveiro
2014

Departamento de Eletrónica, Telecomunicações e Informática

Hélio Salomão Pesanhane

Análise e Integração de Dados Espaciais em Sistemas de Web Semântica

Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2014

Hélio Salomão Pesanhane

Análise e Integração de Dados Espaciais em Sistemas de Web Semântica

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Sistemas de Informação, realizada sob a orientação científica do Prof. Doutor Hélder Troca Zagalo e coorientação científica do Prof. Doutor José Manuel Matos Moreira, Professores Auxiliares do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho à minha esposa Vanda Venâncio Nhassavele e aos meus filhos que são a minha fonte de inspiração.

O júri

Presidente **Prof. Doutor Joaquim João Estrela Ribeiro Silvestre Madeira**

Professor Auxiliar da Universidade de Aveiro

Vogais **Prof. Doutor Fernando Joaquim Lopes Moreira**

Professor Associado da Universidade Portucalense (Arguente)

Prof. Doutor Hélder Troca Zagalo

Professor Auxiliar da Universidade de Aveiro (Orientador)

Agradecimentos

Em primeiro lugar, agradecer ao meu orientador Prof. Doutor Hélder Troca Zagalo e ao meu coorientador Prof. Doutor José Manuel Matos Moreira, pela sua disponibilidade, dedicação, motivação, compreensão e, sobretudo, pelo apoio na realização deste trabalho.

Ao Ministério da Ciência e Tecnologias de Moçambique pela concessão da bolsa de estudos e ao Banco Mundial pelo financiamento da mesma.

Aos professores do curso de Mestrado em Sistema de Informação da Universidade de Aveiro pela formação de qualidade.

Aos meus familiares, especialmente, à minha esposa, aos meus pais, aos meus filhos e meus irmãos pelo carinho e apoio incondicional.

Por fim, para todos aqueles que me ajudaram directa ou indirectamente durante toda minha vida académica.

Palavras-chave

web semântica, ontologias, pesquisas semânticas, repositórios semânticos
dados espaciais, análise espacial, pesquisas espaciais, repositórios espaciais.

Resumo

Nos últimos anos tem-se verificado o crescimento na utilização de Sistemas de Informação Geográfico (SIG). Estes sistemas permitem indexar, armazenar e recuperar dados espaciais de forma eficiente. Graças às funções e operações espaciais, esta tecnologia tem capacidade de análise, por isso está presente em muitos sistemas de apoio a decisão (DSS). Com o crescimento e maturidade destas tecnologias e a tentativa de partilhar e agregar a informação de diferentes repositórios surge o problema de interoperabilidade. A web semântica possui ferramentas e técnicas que oferecem uma base tecnológica necessária para tratar das questões de interoperabilidade.

O objetivo desta dissertação é desenvolver uma solução tecnológica que permita o acesso, processamento e integração de dados espaciais com dados semânticos. O sistema proposto baseia-se num projeto de investigação que propõe e descreve processos de integração de dados espaciais em web semântica.

Os dados são armazenados e recuperados sem abdicar dos seus sistemas nativos de armazenamento e processamento. No sistema proposto os dados são armazenados em repositórios diferentes. Os dados espaciais são armazenados numa base de dados espacial e os dados semânticos numa base de dados semântica.

Para demonstrar o funcionamento do sistema proposto foi desenvolvido um caso de estudo para caraterizar zonas territoriais no Município de Aveiro. O sistema permite caraterizar as diversas zonas como comercial, residencial, rural, etc. e responde perguntas relacionadas com a localização dos edifícios, estradas, hidrovias e ferrovias.

Keywords

Web semantic, Ontology, Semantic search, Semantic repositories, Spatial data, Spatial analysis, Spatial search, Spatial repositories.

Abstract

The geographic information system has been growing in the last years. These systems allow indexing, storing, and retrieving spatial data efficiently. Because of the spatial funtions and operation, this technology has analyze capabilities therefore is present in many decision support systems (DSS). The growth and maturity of this technology and the attempt to share information from different repositories to add value to it naises the problem of interoperability. The Semantic Web has the tools and techniques that provide a necessary technology base to address issues of interoperability.

The objective of this dissertation is to develop a technology solution that allows accessing, processing and the integration of spatial data with semantic data. The proposed system is based on a research project that proposes and describes the processes of integration of spatial data in web semantic.

The data are stored and retrieved without giving up their native systems of storage and processing. In the proposed system the data is stored in different repositories. The spatial data are stored in a spatial data base and semantic data in the semantic data base.

To demonstrate the operation of the proposed system it was developed a case study to characterize territorial areas in the city of Aveiro. The system allows characterizing the areas as commercial, residential, rural, urban and accessible or inaccessible. It also answers other questions related to the location of buildings, roads, waterways and railways.

Conteúdo

Lista de figuras

Lista de tabelas

Lista de acrónimos

DSS - *decision support system*

EPSG - *European Petroleum Survey Group*

ESRI - *Environmental Systems Research Institute*

GML - *Geography Markup Language*

INE - Instituto Nacional de Estatística

OGC - *Open Geospatial Consortium*

OSM - *OpenStreetMap*

RCC - *Region Connection Calculus*

WGS - *World Geodetic System*

WKT - *Well-Known Text*

I Introdução

Esta introdução fornece uma visão geral da dissertação. Apresenta o contexto e motivação. Destaca os objetivos da dissertação e por fim, este capítulo explica como a dissertação está organizada.

1. Motivação e contexto

Os sistemas de informação geográfica (SIG) evoluem à medida que as tecnologias mudam. Estes sistemas possuem extensões espaciais que permitem armazenamento, indexação e recuperação de dados geoespaciais de forma eficiente. As funções e operações espaciais proporcionam a esta tecnologia a capacidade de análise, tornando-a desta forma muito utilizada para apoiar os processos de tomada de decisão (DSS). As funções e operações permitem responder de forma eficiente a perguntas como "Quais são os alunos que vivem num raio de um quilómetro da escola onde frequentam? ", “Quais são os hotéis localizados num raio de quinhentos metros de uma estação de comboios ou de um Aeroporto?”.

O crescimento da tecnologia espacial em sistemas de base de dados fez com que o processo de análise fosse independente do fornecedor da aplicação SIG, o que contribuiu para o desenvolvimento de aplicações SIG personalizadas. A maturidade da tecnologia espacial em sistemas de base de dados simplificou os processos de captura e armazenamento de dados espaciais. Porém, esta maturidade trouxe os seus problemas, os desenvolvedores disponibilizaram diferentes formatos para obter e armazenar dados espaciais, surgindo o problema de interoperabilidade. O crescimento explosivo das tecnologias de informação, em particular, a *World Wide Web* (WWW) tornou global o acesso à informação. Assim não faz sentido, trabalhar de forma independente e com um sistema com estrutura específica. A informação precisa de ser compartilhada, a fim de agregar valor à mesma.

Muitas entidades do domínio geoespacial têm significados que não podem ser totalmente expressos nos atuais sistemas de informação *geoespaciais*. Por exemplo, um rio pode ser

um curso de água e ao mesmo tempo ser um percurso de transporte. Além disso, os problemas do domínio geoespacial muitas vezes exigem juntar dados de múltiplas fontes para resolver um problema particular. Por exemplo combinar informação do hotel e informação da rota de viagem poderia levar à planificação de viagens significativamente mais sofisticadas do que é possível atualmente. Isso pode ser conseguido facilmente com a definição de ontologias e implementação de regras de inferências

As soluções de armazenamento de dados RDF tornam-se cada vez mais populares. Os repositórios semânticos são capazes de melhorar a resolução de vários tipos de problemas que as bases de dados relacionais e os SIG dificilmente conseguem resolver ou não têm intenção de resolver, como: consultas com muitas junções entre as entidades; consultas com várias propriedades; e inferências ontológicas em conjuntos de dados. Esta característica leva-nos aos problemas que envolvem exploração de dados, através da ligação de conjuntos de dados, e abstração de dados de baixo nível.

As ontologias em web semântica representam informações num formato que é legível para as máquinas. Elas são utilizadas para mapear a semântica dos dados, tornando possível a interpretação dos mesmos. Com as ontologias tornou-se fácil compreender o significado dos dados num contexto particular, uma vez que, os dados são mapeados semanticamente com os seus componentes relacionados. Portanto é possível compreender a semântica dos dados, distinguir os componentes diferentes e semelhantes uns aos outros, o que fornece uma base para a interoperabilidade.

A engenharia de ontologias ainda não tem maturidade suficiente para armazenar dados espaciais ou para realizar análises espaciais complexas. Porém as funções e operações espaciais contidas na tecnologia de base de dados podem colaborar com a tecnologia de web semântica para se interpretar conhecimento sobre dados espaciais. É neste contexto que a presente dissertação propôs o desenvolvimento de uma solução tecnológica que faz a análise e integração de dados espaciais em web semântica. Pressupõe-se que o sistema proposto irá receber pesquisas doutra aplicação cliente e devolver respostas. Essas pesquisas podem ser semânticas ou semânticas e espaciais. O resultado da pesquisa espacial enriquece a base do conhecimento definido na base de dados semântica. Este resultado espacial é obtido através da execução de funções e operações espaciais a nível da base de dados espacial. Sobre os dados inseridos na base de dados semântica são

implementadas regras da linguagem semântica no sentido de inferir conhecimento. Para demostrar o funcionamento do sistema proposto desenvolveu-se um caso de estudo para caracterizar zonas territoriais no Município de Aveiro. O sistema permite caracterizar as zonas como comercial, residencial, rural, urbana e em zona acessível ou de difícil acesso. Responde também outras perguntas relacionadas com a localização dos edifícios, estradas, hidrovias e ferrovias.

2. Objetivos

Com este trabalho pretende-se desenvolver uma solução tecnológica que permita:

- o acesso, processamento e integração de dados espaciais com dados semânticos;
- avaliar a aplicabilidade de diferentes soluções e padrões que propõem a integração de dados espaciais em web semântica;
- conceber o sistema de integração de dados espaciais em web semântica aplicando uma metodologia de análise orientada a objetos;
- testar e validar a solução desenvolvida.

3. Estrutura do documento

Este documento encontra-se dividido em cinco partes principais, onde cada capítulo representa uma parte.

Neste primeiro capítulo é feita a introdução, contextualização deste trabalho e são definidos os objetivos que o mesmo pretende atingir.

O Capítulo II descreve os dados espaciais e sistemas de informação geográficas, as tecnologias de web semântica, a integração dos dados espaciais em web semântica, o padrão GeoSPARQL, o projeto de investigação que foi seguido para o desenvolvimento do sistema proposto e por fim as tecnologias utlizadas para o desenvolvimento do sistema proposto.

O Capítulo III apresenta os requisitos, a arquitetura e a implementação do sistema.

O Capítulo IV apresenta e explica o caso do estudo.

Finalmente, o Capítulo V, que serve como conclusão e revisão de tudo o que foi feito ao longo deste trabalho, faz análise crítica do trabalho e formula os possíveis objetivos futuros a serem concretizados para este trabalho.

II Estado de arte

1. Dados espaciais e sistemas de informação geográfica.

Os **Sistemas de Informação Geográfica** ou simplesmente **SIG** designam um conjunto de tecnologias para recolha, armazenamento, manipulação e visualização de dados espaciais, muitas vezes usadas em geociências ou áreas afins [1]. Os dados espaciais, também designados dados geográficos ou dados geoespaciais, descrevem aspetos como a localização ou a topologia de objetos num contexto geográfico. Desta forma, é possível representar o mundo real por dois descritores: "O quê" e "Onde".O descritor "O quê" descreve o que se está a representar, os objetos em causa e os objetos com que se relaciona. O descritor "Onde" descreve a localização dos objetos e a extensão espacial dos mesmos, conforme mostra a Figura II-1.

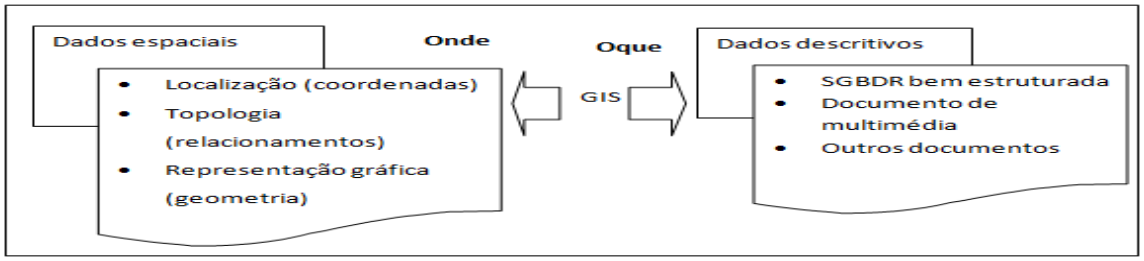


Figura II-1:Descritores de um SIG com os seus componentes, adaptado de A.Karmacharya [2].

1.1. Dados espaciais

Os dois tipos de representação de dados espaciais mais comuns são a representação vetorial e a representação matricial.

1.1.1. Representação vetorial

Os dados vetoriais SIG são constituídos por pontos, na sua forma mais simples, ou por linhas ou arcos definidos através de pontos, que se unem [3].

A representação vetorial de mapas utiliza primitivas geométricas como pontos, linhas, curvas e polígonos para representar dados espaciais. Os dados vetoriais são independentes da resolução e podem ser redimensionados sem perder detalhes ou clareza. Então, não há impacto sobre a qualidade quando são realizadas operações vetoriais como ampliar, rodar, redimensionar ou transformar dados espaciais A Figura II-2 mostra a representação vetorial de polígonos, linhas e pontos num cenário do mundo real.

Figura II-2 Representação vetorial de um cenário do mundo real [3].

A análise vetorial é o processo de consulta de dados recorrendo a funções e operações espaciais sobre dados vetoriais, em que o resultado é apresentado tanto em mapas como sob a forma de texto ou números.

1.1.2. Representação matricial

Uma representação matricial divide uma área geográfica em células. O tamanho das células determina a resolução da representação e depende das necessidades dos utilizadores. No caso de uma representação matricial regular, cada célula está associada a um par de coordenadas determinadas através dos números da linha e da coluna na matriz. Portanto, as coordenadas determinam a localização de um objeto. As células são geralmente codificadas com um valor que representa o valor geográfico ou nominal, como o valor de precipitação ou uma temperatura. A Figura II-3 ilustra o modo como uma área geográfica pode ser representada matricialmente.

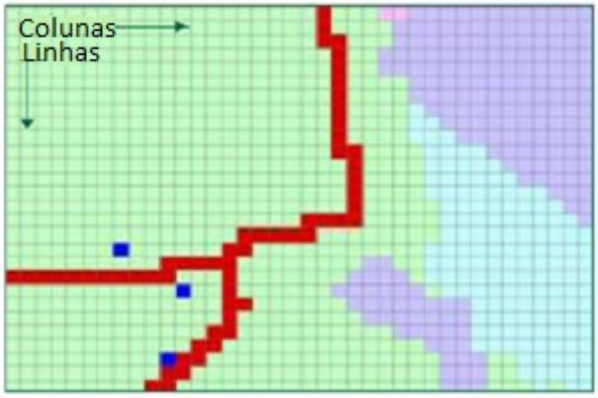


Figura II-3:Representação matricial [2].

A maioria dos operadores em análise matricial é baseada em análise trigonométrica, exponenciais e logarítmicas, reclassificação, seleção (com base em uma condição), estatística, aritmética.

Vantagens e desvantagens dos formatos de representação vetorial e matricial

A Tabela II-1 mostra a comparação das representações vetorial e matricial [2].

Formatos de representação	Vantagens	Desvantagens
Vetorial	• A aparência é esteticamente agradável; • A topologia é preservada e pode ser utilizada explicitamente; • Pode ser representada recorrendo a tipos de dados espaciais, facilitando a integração em sistemas de gestão de bases de dados; • Os dados podem ser manipulados usando operações e funções espaciais.	• Requer a transformação de dados capturados no formato matricial em dados vetoriais; • Dificuldade na representação de objetos em que a forma não pode ser definida com precisão.

Matricial	• Não há armazenamento explícito da informação; • Fácil de formular análise de dados devido à natureza dos dados (cálculo algébrico); • Frequentemente utilizado em Modelos Digitais de Elevação (DEM); • Ideal para a análise estatística.	• Armazenamento: Coordenadas representadas como linhas e colunas; • Apenas um atributo poderia ser usado em cada momento; • Resolução: depende do tamanho da célula; • Aparência: estrutura matricial, portanto, não agradável; • Topologia: implícita, portanto, não utilizados frequentemente; • Um sistema de base de dados pode fornecer armazenamento de dados no formato matricial mas requer uma estrutura complexa.

Tabela II-1: Comparação geral das representações vetorial e matricial

1.2. **Análise geoespacial**

O processo de análise espacial numa base de dados SIG permite interagir com dados espaciais, respondendo desta forma às muitas questões que podem ser colocadas por consultas espaciais [4].

Estas operações e funções podem ser classificadas de acordo com sua natureza, nomeadamente: consultas geoespaciais, medição geoespacial, processamento geoespacial e relacionamentos geoespaciais.

1.2.1. Consultas geoespaciais

Estas operações são realizadas de forma semelhante ao das consultas realizadas num sistema de gestão de bases de dados relacional. A diferença é caracterizada por componentes espaciais incluídos dentro das instruções. As operações espaciais incluídas

nas consultas levam em conta as geometrias para processar e o resultado deste processo é combinada com outros dados não espaciais para devolver os resultados.

1.2.2. Medição geoespatial

São operações que retornam as propriedades espaciais dos objetos através de cálculo geométrico. Exemplos de tais operações são "área ", " perímetro ", " comprimento ", etc. Algumas operações dentro desta categoria retornam uma relação entre dois objetos. Por exemplo, a distância entre dois objetos.

1.2.3. Processamento geoespatial

As operações nesta categoria alteram o conjunto de dados através da combinação das geometrias de dois ou mais objetos ou pelo processamento de um objeto para retornar a geometria. Em todos casos, as funções retornam uma forma geométrica. A Tabela II-2 mostra operações e funções para o processamento geoespacial.

Operações e funções espaciais	**Descrições**
Buffer	As operações *Buffer* geram uma nova geometria que se estende para fora ou para dentro de uma geometria existente por um raio específico.
União	A operação de união retorna a união de duas geometrias. Esta operação também pode ser aplicada a todos os tipos de objetos ou combinações de pelo menos dois objetos distintos.
Intersecção	A operação de intersecção retorna a geometria de intersecção das geometrias de dois objetos espaciais. Esta operação pode ser aplicada a todos os tipos de objetos ou a combinações de dois ou mais tipos de objetos diferentes.
Diferença	Retorna a geometria que não é intersecção de duas características.

Tabela II-2: Função para o processamento geoespacial

1.2.4. Relacionamentos geoespaciais

Estas operações retornam um valor booleano, isto é, verdadeiro ou falso quando executadas, e permitem, por exemplo, determinar a relação topológica entre duas geometrias. A Tabela II-3 mostra algumas funções para relacionamentos geoespaciais.

Funções	**Descrições**
Disjunto	As geometrias não possuem nenhuma parte em comum.
Igual	As geometrias são espacialmente iguais.
Toca	As geometrias tocam uma com a outra. Isso significa que os exteriores das geometrias podem tocar uns aos outros, mas não têm nenhum ponto interior em comum.
Dentro	A primeira geometria está completamente dentro da segunda geometria.
Sobreposição	A intersecção das geometrias resulta numa geometria que é diferente das geometrias originais.

Tabela II-3:Funções para à análise relacionamentos geoespaciais.

1.2.5. Bases de dados espaciais

Atualmente, os principais sistemas de gestão de bases de dados suportam estas operações espaciais através de extensões à linguagem SQL como exemplificado na Tabela II-4.

Categoria	**Funções ou operações espaciais**	**Declarações SQL**
Consultas Geoespaciais	Area/Length	SELECT column_name FROM table WHERE ST_Length (geom)>value;
Medição Geoespacial	Area/Length/Distance	SELECT ST_Length (geom,) From Table; SELECT ST_Distance(fromgeom,togeom)

		From Table;
Processamento Geoespacial	Buffer/Union/Intersectin/ Difference	SELECT ST_Buffer(geom) FROM Table; SELECT ST_Union (geom) FROM Table WHERE Country='Portugal'
Relacionamento Geoespacial	Disjoints/Equals/Touches/Within/Overlaps	SELECT Table1.name, Table2.name FROM Table1, Table2 WHERE ST_Overlaps(geom1, geom2);

Tabela II-4: Exemplos de operações espaciais num sistema de gestão de bases de dados (PostgreSQL / PostSIG)

As funções Area e *length* devolvem a área e o comprimento de uma geometria respetivamente. A instrução do Código II-1 mostra um exemplo de uma consulta geoespacial.

```
SELECT column_name FROM table WHERE ST_Length(geom)>value;
```

Código II-1: exemplo de uma consulta geoespacial

O Código II-1 permite obter todas geometrias em que o comprimento é maior que o valor representado pela variável *value*. Neste caso as funções são utilizados para determinar o critério de seleção das geometrias.

A função *distance* retorna a distância entre duas geometrias. O resultado desta medida representa a distância entre os pontos localizados no centro de cada geometria.

A instrução do Código II-2 ilustra exemplo de uma consulta de medição geoespacial.

```
SELECT ST_Length (geom) FROM Table;
```

Código II-2: exemplo de uma consulta de medição geoespacial

O Código II-2 devolve o comprimento das geometrias.

A declaração SQL apresentada pelo Código II-3 mostra um exemplo de uma consulta de processamento geoespacial.

```
 SELECT ST_Buffer(geom), FROM Table;
```

Código II-3: exemplo de uma consulta de processamento *geoespacial*

A consulta do Código II-3 permite gerar e retornar uma geometria estendendo para fora ou para dentro a geometria selecionada de acordo com o valor de extensão oferecido.

A instrução SQL do Código II-4 constitui um exemplo de processamento geoespacial envolvendo duas geometrias.

```
SELECT  ST_Union(geom)FROM Table WHERE Country='Portugal';
```

Código II-4: exemplo de consulta de processamento geoespacial envolvendo duas geometrias

O Código II-4 permite obter uma nova geometria que representa a união das geometrias associadas ao país com o nome 'Portugal'.

O SELECT Table1.name, Table2.name FROM Table1, Table2 WHERE ST_Overlaps(geom1,geom2);

Código II-5 representa uma instrução SQL que retorna um valor booleano em função do teste que é feito entre duas geometrias. O exemplo apresentado permite determinar quais são os pares de geometrias das tabelas 1 e 2 que se sobrepõem.

```
SELECT Table1.name, Table2.name FROM Table1, Table2 WHERE
ST_Overlaps(geom1,geom2);
```

Código II-5: exemplo de consulta de relacionamento geoespacial

1.3. **Recursos e geometrias**

Recursos e geometrias são dois conceitos fundamentais da ciência geoespacial. Um Recurso é simplesmente qualquer entidade no mundo real com alguma localização espacial. Este poderia ser uma escola, hospital, supermercado, hotel, aeroporto, um restaurante, etc. Um recurso pode ter uma localização espacial que não pode ser definido com precisão, como um pântano ou uma montanha.

Uma geometria é qualquer forma geométrica, tal como um ponto, um polígono ou uma linha, e é utilizada como uma representação da localização espacial e da forma de um recurso. Por exemplo, Universidade de Aveiro é um recurso geoespacial, porque é uma entidade que tem uma localização específica no mundo e uma geometria. A geometria pode ser definida em diversas resoluções; por exemplo, a Universidade de Aveiro pode ser

representada simplesmente por um ponto, representando as suas coordenadas geográficas, ou um polígono representando a sua fronteira ou geometria [5].

1.3.1. Sistemas de coordenadas de referência

Uma parte importante dos metadados associados a uma geometria é o seu sistema de coordenadas de referência (CRS), também denominado como sistema de referência espacial. Os elementos de um sistema de coordenadas de referência permitem descrever com precisão a posição e a geometria dos recursos, e estabelecer relações entre geometrias. Existem quatro partes que compõem um CRS: um sistema de coordenadas, um elipsoide, uma origem e uma projeção. Um sistema de coordenadas descreve uma localização relativa a algum centro. O sistema de coordenadas geocêntrico coloca a origem no centro da Terra.

Uma Coordenada geográfica (ou geodésica) utiliza uma superfície esférica para determinar a localização. Um ponto é definido pelos ângulos medidos a partir do centro da Terra para um ponto na superfície. Estes são também conhecidos como latitudes (horizontal) e longitudes (vertical). Um sistema de coordenadas cartesianas é um sistema plano de coordenadas da superfície que permite medições rápidas e precisas em pequenas distâncias.

Um elipsoide define uma aproximação para o centro e a forma da Terra. Este fornece um quadro de referência para a medição de locais e, recorrendo a pontos de referência locais, permite obter localizações precisas numa dada área geográfica.

Um mapa da terra deve ser projetado a partir de uma curva da superfície do plano. Esta projeção vai distorcer a superfície de alguma forma o que significa que as coordenadas para alguns locais são mais precisas do que as outras. Algumas projeções irão preservar área, de modo que o tamanho de todos os objetos seja relativo, enquanto outros preservam ângulos, e outros tentam fazer as duas coisas. Um sistema de coordenadas projetadas sobre um plano permite desempenho mais rápido na execução de cálculos num sistema cartesiano. No entanto, os cálculos são imprecisos quando lidam com grandes áreas e a curvatura da Terra não é tida em conta [6].

A combinação destes elementos define um CRS. Uma fonte comum de Sistema de referências bem definido é a *Europian Petroleum Survey Group Europeia* (EPSG) [7].

2. Web semântica

Genericamente, a web semântica consiste num conjunto de padrões e boas práticas, a adotar para aplicações informáticas, para partilha de dados e sua semântica na web.

Tim Berners-Lee fundou a *World Wide Web Consortium* (W3C) para supervisionar os padrões para a criação e interpretação de conteúdos na web [8]. A web semântica foi pensada, tendo por base padrões da W3C, tais como: os modelos de dados RDF – *Resource Description Framework* e RDFS – *RDF Schema*; o OWL – *Web Ontology Language* para a criação de vocabulários e ontologias; e a linguagem de pesquisa SPARQL – *SPARQL Query Language for RDF*.

Outro conceito, intimamente ligado à web semântica, é o *linked data*. Este consiste, genericamente, também na adoção de padrões e boas práticas para a exposição, partilha e integração de dados na web. Neste é recomendado a utilização de padrões como RDF, SPARQL e URIs – *Uniform Resource Identifier*, para a identificação únivoca de dados. O *linked data* ofereceu excelentes diretrizes para a criação de uma infraestrutura para a web semântica [9].

A título de exemplo, Berners Lee deu o nome de URL – *Uniform Resource Locator*, à combinação do nome de servidor, pastas e recursos que podem ser encontrados na web, utilizando um protocolo específico de Internet (ex: http://www.learningsparql.com/resources/index.html).

2.1. Principais características do RDF

O padrão RDF é um modelo de dados que permite representar recursos na web e possui as seguintes características:

- é crucial quando a informação precisa de ser processada por aplicações e não apenas visualizada para pessoas;
- a informação trocada entre diferentes aplicações é estruturada de uma forma comum sem que se perca o significado [10];

- os objetos, ou entidades, são identificados utilizando identificadores web chamados URIs e a descrição de recursos é feita através de propriedades e seus valores. Isto permite representar recursos, suas propriedades e valores como um grafo de nós e arcos. Um recurso é considerado sinónimo de entidade, isto é, designação genérica de qualquer coisa num dado domínio [10].

2.2. Declarações sobre recursos em RDF

O modelo de dados RDF permite a descrição de dados recorrendo a conjuntos de declarações. Estas necessitam de obedecer a uma estrutura na qual se identificam as seguintes partes:

- **sujeito** - identifica o objeto na declaração;
- **predicado** - identifica uma propriedade ou uma característica de um recurso;
- **objeto** - identifica o valor de uma propriedade.

Tomando como exemplo prático a necessidade de descrever uma dada entidade, como uma freguesia, por exemplo, na qual a freguesia é identificada pelo código 010506 e possui um nome cujo valor é Glória. Esta descrição pode ser feita através de uma declaração RDF, como:

- **sujeito** - código "010506";
- **predicado** - a palavra "nome";
- **objeto** – o valor "Glória".

A mesma informação pode ser representada utilizando um grafo conforme mostra a Figura II-4

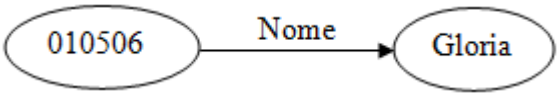


Figura II-4: Exemplo de declaração RDF.

As declarações sobre recursos representadas em RDF tornam fácil o seu processamento automatizado pelas máquinas. Este processamento automatizado requer [10]:

- um sistema automatizado que consiga identificar o sujeito, o predicado e/ou o objeto numa declaração sem ambiguidade;
- uma linguagem para representar estas descrições que facilitem o intercâmbio de informações entre máquinas.

A atual arquitetura web satisfaz estes requisitos. A web fornece uma forma de identificação dos recursos através de URLs, consistindo numa sequência de carateres que identifica a localização de uma página web.

A web semântica [11] utiliza URIs para a identificação geral dos recursos. Os URIs não se referem apenas aos recursos com endereço na web, mas sim, a qualquer coisa que precise de ser referenciada numa declaração RDF.

O RDF, nas suas declarações, utiliza URIs como base de identificação de sujeitos, predicados e objetos. Uma referência URI pode ser composta por uma parte genérica e uma parte específica. Por exemplo, o URI "*http://www.ua.pt/nuts/individuos#010506*" é constituído pela parte genérica "http://www.ua.pt/nuts/individuos" e pela parte específica "010506" separadas pelo carater #. A adoção desta estrutura permite o reaproveitamento da parte genérica do URI e a contextualização do recurso a indentificar.

Por forma a possibilitar o intercâmbio entre máquinas, as declarações RDF necessitam de ser representadas de alguma forma. O padrão XML – *Extensible Markup Language* pode ser utilizado para este efeito, passando a designar-se esta representação como: RDF/XML.

2.3. Modelo de dados

Nesta secção pretende-se descrever como os URIs são utilizados para fazer declarações RDF sobre os recursos. A declaração na linguagem natural "O edifício com o código 136668920 pertence à freguesia com o código 010506" pode ser representada da seguinte forma, em RDF:

- **sujeito**: http://www.ua.pt/nuts/individuos#136668920;

- **predicado**: http://www.ua.pt/nuts/predicados/pertence;
- **objeto**: http://www.ua.pt/nuts/individuos#010506.

Neste caso ao invés de se utilizar apenas os códigos "136668920" e "010506" e a palavra "pertence" para identificar o sujeito, predicado e objeto, foram utilizadas referencias URIs. Em RDF, o modelo de dados é representado por grafos. Uma declaração RDF consiste num nó para o sujeito, um nó para o objeto e um arco para o predicado. Como mostra a Figura II-4, o vocabulário de um grafo é o conjunto de nomes que constituem sujeitos, predicados e objetos de um grafo.

As declarações RDF podem ser representadas utilizando triplos. Na notação de triplos, cada declaração no grafo é também constituído por sujeito, predicado e objeto, o que consiste num triplo. Isto é, a declaração de um triplo é feito na forma: [sujeito, predicado, objeto]. Consideremos o exemplo da Figura II-5.

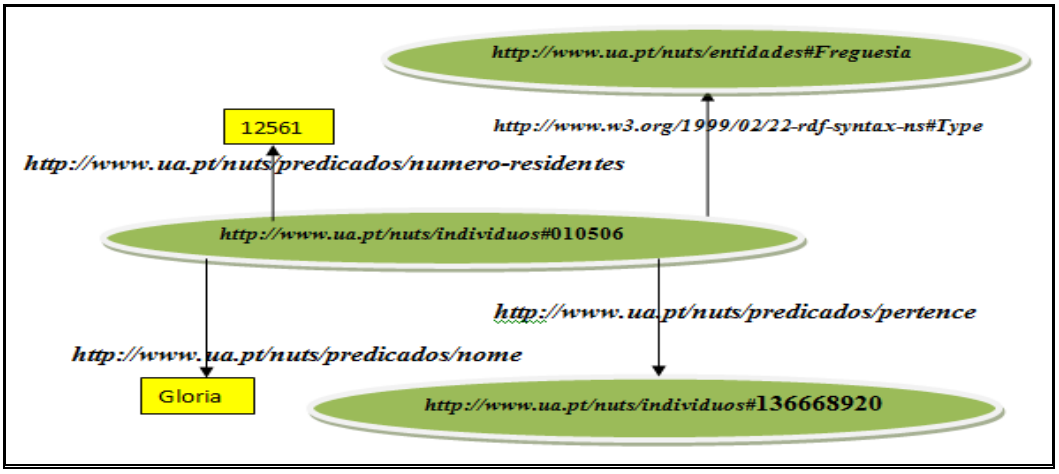


Figura II-5: Várias declarações sobre o mesmo recurso.

O Código II-1 mostra como as declarações RDF na forma de grafo da Figura II-5 podem ser feitas na forma de triplos.

```
1 <http://www.ua.pt/nuts/individuos#010506>
2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#Type>
3 <http://www.ua.pt/nuts/entidades#Freguesia> .
4
5 <http://www.ua.pt/nuts/individuos#010506>
```

```
6 <http://www.ua.pt/nuts/predicados/numero-residentes>
7 "12561" .
8
9 <http://www.ua.pt/nuts/individuos#010506>
10 <http://www.ua.pt/nuts/predicados/nome>
11 "Gloria" .
12
13 <http://www.ua.pt/nuts/individuos#010506>
14 <http://www.ua.pt/nuts/predicados/pertence>
15 <http://www.ua.pt/nuts/individuos#136668920> .
```

Código II-6: Declarações RDF em forma de triplos.

Na Figura II-5, a declaração do triplo (código 1, linha 1 a 3) corresponde ao arco, <http://www.w3.org/1999/02/22-rdf-syntax-ns#Type>, que começa no nó <http://www.ua.pt/nuts/individuos#010506> e termina no nó <http://www.ua.pt/nuts/entidades#Freguesia>.

A escrita de triplos pode ser abreviada e simplificada por forma a encurtar as linhas. A escrita abreviada substitui uma referência URI por um prefixo. Uma abreviatura é constituída por um nome qualificado XML (também chamado por *Qname*) não tendo os sinais maior (>) e menor (<). Um *Qname* é constituído por um prefixo associado a um URI [12], sucedido por dois pontos ":" e um nome. Por exemplo se o *Qname* edifício é associado ao *namespace* URI *http://www.ua.pt/nuts/edificios* então o *Qname edificio:escola* é abreviatura da escrita para a referência *http://www.ua.pt/nuts/edificios/escola*. Exemplo de utilização de prefixos:

- prefixo rdf: => *namespace* URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
- prefixo nse: => *namespace* URI: http://www.ua.pt/nuts/entidades#
- prefixo nsi: => *namespace* URI: http://www.ua.pt/nuts/individuos#
- prefixo nsp: => *namespace* URI: http://www.ua.pt/nuts/predicados

Os triplos apresentados no Código II-6 podem ser rescritos utilizando os *Qnames* construídos a partir destes prefixos. Como mostra o Código II-7.

```
nsi:010506 rdf:type nse:Freguesia .
nsi:010506 nsp:nome “Gloria” .
nsi:010506 nsp:numeroResidentes “12561” .
nsi:010506 nsp:pertence nsi:136668920 .
```

Código II-7: Declarações RDF em forma de triplos usando *qnames*.

2.4. **Nós em branco e sua utilidade**

Para explicar a utilidade dos nós brancos, consideremos um conjunto de triplos sem nenhum nó branco, e depois analisemos como um nó em branco pode ajudar a organizá-los melhor.

Consideremos alguns triplos que representam os dados pessoais num livro de endereços. Como mostra o Código II-8.

```
Prefix ab: <http://www.ua.pt/nuts/addressbook#> .

ab:i0432 ab:firstName "Salomao" ;
     ab:lastName "Noge" ;
     ab:postalCode "49345" ;
     ab:city "Maputo" ;
     ab:homeTel ""(351) 920-228472" ;
     ab:streetAddress "R. Carlos Cardoso" ;
     ab:region "Baixo Vouga" ;
     ab:email "noge@hotmail.com" .
```

Código II-8: Exemplo de triplos sem nós em branco.

A Figura II-6 representa o grafo de dados. Cada valor do sujeito ou objeto é um nó do grafo e os arcos que ligam os nós mostram os seus relacionamentos.

A ordem de triplos não importa em RDF, e a informação do endereço eletrónico da pessoa com nome Salomão pode ser difícil de encontrar entre as outras informações sobre a mesma.

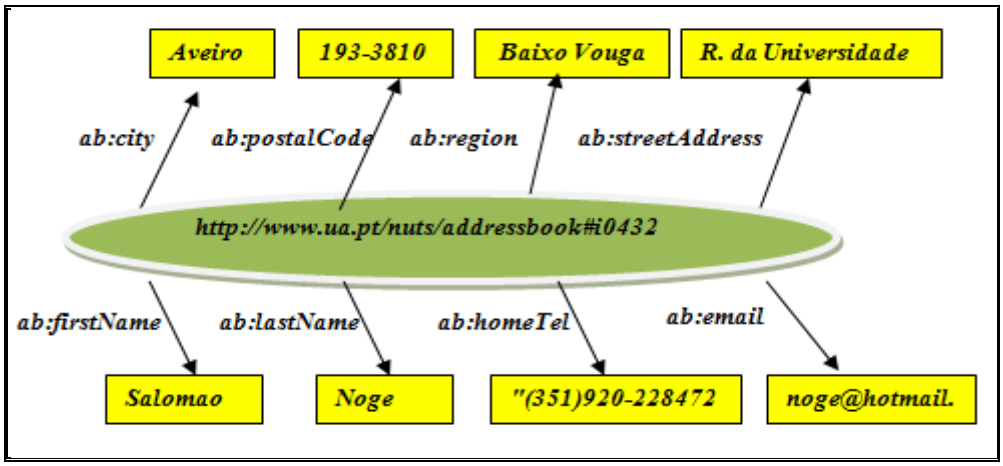


Figura II-6:Grafo de entradas do *ab:i0432* no livro de endereços sem nó branco

A seguinte versão tem um novo predicado, *ab:address*, mas o seu valor tem um prefixo que pode parecer estranho: um sublinhado. Esse valor, por sua vez tem os seus próprios valores, descrevendo os componentes individuais do endereço do indivíduo Salomão. Como é visível no Código II-9

```
Prefix ab: <http://www.ua.pt/nuts/addressbook#> .

ab:i0432 ab:firstName "Salomao" ;
     ab:lastName "Noge" ;
     ab:email "noge@hotmail.com" .
     ab:homeTel "(351) 920-228472" ;
     ab:address _:b1 .

_:b1    ab:postalCode "49345" ;
     ab:city "Maputo" ;
     ab:streetAddress "R. da Universidade" ;
     ab:region "Baixo Vouga" ;
```

Código II-9: Exemplo de triplos com um nó em branco.

O prefixo sublinhado significa que este é um tipo especial de nó, conhecido como um nó em branco ou *bnode*. Não possui uma identidade permanente; o seu objetivo é agrupar outros valores. O *b1* é utilizado como um identicador local e é apenas uma referência reservada para o caso de outras partes desse conjunto de dados precisar referir-se a estes triplos agrupados. As aplicações de *software* que leem estes dados podem ignorar o valor *b1*, mas devem "lembrarem-se" que o indivíduo de nome Salomão, ou recurso *ab:i0432*,

tem um valor para *ab:address* que aponta para quatro outros valores. A Figura II-7 mostra um grafo com um nó branco.

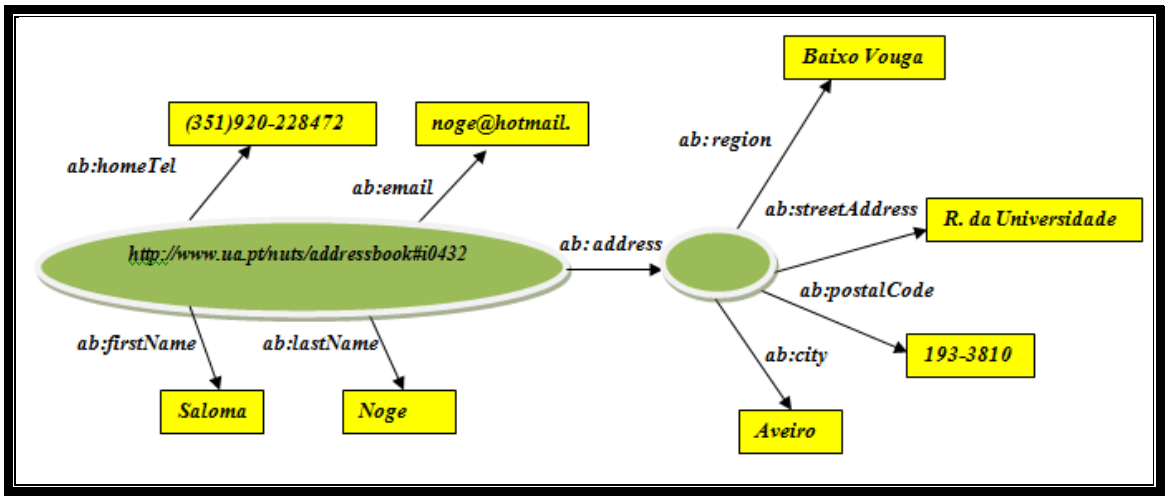


Figura II-7:Utilização do nó em branco para agrupar juntos dados do endereço

No Código II-9, o nó branco _: b1 é o objeto de um triplo e sujeito de vários outros. Isso é comum em RDF porque nós em branco são, na verdade, utilizados para ligar recursos entre si. Por exemplo, se tivermos os dados do livro de endereços estruturado como mostra o Código II-9, é possivel utilizar uma consulta SPARQL para procurar o endereço da rua do Salomão, pedindo a valor *ab:streetAddress* do nó *ab:address* a partir da entrada do catálogo de endereços que tem um nome *ab:firstName* "Salomão" e um sobrenome *ab:lastName* "Noge". O nó *ab:address* não tem um URI que possa ser utilizado para se referir a ele, mas a consulta não precisa dele porque se pode apenas dizer que se pretende os valores de endereço a partir da entrada com valor “Salomão” do predicado *ab:firstName* e valor “Noge” no predicado *ab:lastName*.

2.5. Grafos RDF com nomes

Os grafos com nomes são uma forma de referenciar um conjunto de triplos em RDF. Quando se atribui um nome a um conjunto de triplos, o nome é, na realidade, um URI, o que permite associar metadados a esse mesmo conjunto. Por exemplo, pode-se dizer que um determinado conjunto de triplos, numa *triplestore*, veio de uma determinada fonte num

determinado momento, ou que um determinado conjunto deve ser substituído por um outro conjunto [9].

2.6. Criação e reutilização de vocabulários: RDF *Schema* e OWL

Novos URIs podem ser criados para todos os recursos e propriedades nos triplos, mas quando são utilizados os que já existem é mais fácil ligar os dados com esses. Se se quiser utilizar vocabulários de propriedades existentes com outros semelhantes, que formato deve assumir o seu nome ou URI?

Os diferentes conjuntos de vocabulários necessários à caraterização das entidades nos grafos RDF são geralmente definidos através RDFS e OWL. A utilização destas normas permite, por exemplo, obter maior eficiência em pesquisas SPARQL e o conhecimento de todas propriedades existentes num conjunto de dados a ser pesquisado. As definições adicionam metadados sobre os termos do vocabulário declarado, o que permite que os sistemas "aprendam" sobre eles e consigam fazer um melhor uso dos mesmos em pesquisas.

No Código II-10, encontra-se um exemplo da utilização de RDF e RDFS para definir e descrever uma propriedade: neste caso particular, a propriedade *nomeFreguesia*.

```
prefix nsp: <http://www.ua.pt/nuts/predicados> .
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

nsp:nomeFraguesia
     rdf:type rdf:Property ;
     rdfs:comment "nome de uma freguesia.”@pt-PT;
     rdfs:label "Nome"@pt-PT .
```

Código II-10: Triplos descrevendo o termo "nomeFreguesia".

O primeiro triplo define *nsp:nomeFreguesia* como uma propriedade, utilizando a propriedade *rdf:type* e a classe *rdf:Property.* Os segundo e terceiro triplos utilizam as propriedades *rdfs:comment* e *rdfs:label,* do RDFS, para adicionar informação de contextualização.

O *RDF Schema* permite também definir novas classes para os recursos [10]. Por exemplo, o Código II-11 mostra como podemos declarar as classes *nse:Freguesia* e *nse:Edificio*.

```
prefix nse: <http://www.ua.pt/nuts/entidades> .
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

nse:Freguesia
     rdf:type rdfs:Class ;
     rdfs:label "Freguesia" ;
     rdfs:comment "A menor divisao administrative em
Portugal" .
nse:Edificio
     a rdfs:Class ;
     rdfs:label "edifícios " .
```

Código II-11: Exemplo de definição de novas classes.

Existem muitos metadados que podem ser atribuídos quando uma classe for declarada. Por exemplo, podemos determinar para uma propriedade, um domínio para sujeito e outro para objeto. Como se pode ver no Código II-12, a propriedade *rdfs:domain* significa que se utilizarmos a propriedade *nsp:contemEdificio* num triplo, então o sujeito do triplo é uma *nse:Freguesia*. A propriedade *rdfs:range* significa que o objeto do triplo é um edifício *nse:Edificio.*

```
prefix nsp: <http://www.ua.pt/nuts/predicados> .
prefix nse: <http://www.ua.pt/nuts/entidades> .
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

nsp:contemEdificio
     rdf:type rdf:Property ;
     rdfs:label "edifício contido" ;
     rdfs:comment " edifício contido na freguesia " .
     rdfs:domain nse:Freguesia;
     rdfs:range nse:Edificio .
```

Código II-12: definição do domínio do sujeito e objeto de uma propriedade

Na abordagem orientada a objetos, se se disser "os membros da classe Freguesia têm uma propriedade *contemEdificio*", isto significa que um membro da classe Freguesia deve ter um valor *contemEdificio*. A abordagem RDFS e OWL comporta-se de uma forma

diferente. Os últimos dois triplos dizem nos que se alguma coisa tem o valor nsp:*contemEdificio* é um membro da classe *nse:Freguesia* e o seu valor *nsp:contemEdificio* é membro da classe *nse:Edificio*.

```
prefix nsp: <http://www.ua.pt/nuts/predicados> .
prefix nsi: <http://www.ua.pt/nuts/individuos> .
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

nsi:010505 nsp:nome "Esgueira" ;
           nsp:contemEdificio nsi:136668920 .
```

Código II-13: propriedade com domínio pré definido.

O Código II-13 mostra que uma vez adicionado o triplo *nsi:010505 nsp:contemEdificio nsi:136668920*, os processadores *RDFS* e SPARQL sabem que "Esgueira" (ou o recurso *nsi:010505*) é agora membro da classe "Freguesia" porque *nsp:contemEdificio* tem um domínio *nse:Freguesia.* E porque *nsp:contemEdificio* possui uma gama (*range*) de *nse:Edificio*, *nsi:136668920* é agora membro da classe *nse:Edificio*, mesmo não sendo antes.

Quando solicitamos ao motor SPARQL RDFS para nos devolver os nomes de todas freguesias, ele irá listar Esgueira, mesmo embora não tenhamos nenhum triplo que diga explicitamente que Esgueira é um membro da classe Freguesia.

Utilizando padrões RDF, adicionando uma propriedade para os metadados sobre os recursos existentes *nsi:010505*, esse recurso torna-se membro de uma classe que não era antes.

Esta capacidade de recursos RDF tornarem-se membros de classes com base nos seus valores de dados tornou a tecnologia RDF popular em áreas como a investigação médica e as agências de inteligência.

O RDFS permite ainda definir classes como subclasses de outras e propriedades como subpropriedades de outras, o que amplia as possibilidades de o SPARQL ser utilizado para recuperar informação [11]. Por exemplo, se se declarasse que *nse:Freguesia* é uma subclasse de *nse:divisaoAdministrativa* e em seguida, se pedisse ao processador para devolver nomes de todas as instâncias de *nse:divisaoAdministrativa* num conjunto de

dados, o processador iria listar Esgueira também na categoria de divisão administrativa por ser da classe “Freguesia”.

O OWL baseia-se em RDFS para definir ontologias. ontologias são definições formais de vocabulários que permitem a definição de estruturas complexas bem como novas relações entre os termos de um vocabulário e entre os membros das classes definidas [9]. ontologias frequentemente descrevem domínios específicos tas como pesquisas científicas para que cientistas de diferentes instituições possam facilmente partilharem dados. Analise-se o Código II-14.

```
prefix ab: <http://learningsparql.com/ns/addressbook#> .
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
prefix owl: <http://www.w3.org/2002/07/owl#> .

ab:i0432
     ab:nome "Salomao" ;
     ab:apelido "Noge" ;
     ab: conjuge ab:i9771 .

ab:i8301
     ab:nome "Luis" ;
     ab:apelido "Costa" ;
     ab:paciente ab:i9771 .

ab:i9771
     ab:nome "Cintia" ;
     ab:apelido "Magaia" .

ab:conjuge
     rdf:type owl:SymmetricProperty ;
     rdfs:comment "Identifica conjuge" .

ab:paciente
     rdf:type rdf:Property ;
     rdfs:comment "Identifica o paciente do medico" .

ab:medico
     rdf:type rdf:Property ;
     rdfs:comment "Identifica o medico do paciente" ;
     owl:inverseOf ab:paciente .
```

Código II-14: Exemplo de definição de ontologias.

A propriedade *ab:conjuge* é definida como sendo simétrica e o recurso *ab:i0432*, "Salomao", tem como cônjuge, *ab:cônjuge* o recurso *ab:i9771*, "Cíntia". Neste caso, um processador OWL sabe que o recurso *ab:i9771* tem o recurso *ab:i0432* como seu cônjuge. Ele também sabe que, se a propriedade *ab:paciente* é inversa da propriedade *ab:medico* e o recurso *ab:i8301* tem um paciente *ab:paciente* com valor de *ab:i9771,* então o recurso ab:i9771 tem um médico com valor *ab:i8301.* Agora já sabemos o cônjuge e o médico da Cíntia, mesmo que esses factos não tenham sido explicitamente incluídos no conjunto de dados.

O OWL oferece muitas outras maneiras de definir propriedades e relacionamentos de classe para que um processador possa inferir novas informações a partir de um conjunto de dados existente.

De todos os padrões da web semântica, definidos pelo W3C, OWL é o padrão chave para colocar a "semântica" na "web semântica". O termo "semântica" é muitas vezes definido como o significado por trás das palavras.

3. Web semântica e dados espaciais

Nesta secção são descritos, o GeoSPARQL [5] que é um padrão para armazenamento e recuperação de dados RDF geoespaciais e um projeto de investigação apresentado em [13], que propõe a integração de processamento espacial com a web semântica. O GeoSPARQL e o projeto de pesquisa utilizam o padrão *Open Geoespatial Consortium* (OGC) [14].

3.1. GeoSPARQL

O GeoSPARQL padroniza a inserção e consulta de dados RDF geoespaciais. O padrão GeoSPARQL define uma Ontologia para representar recursos, geometrias, predicados e funções e fornece vários conjuntos de terminologia para especificar relacionamentos topológicos entre geometrias [15]. Cada aplicação pode escolher quais conjuntos de termos utilizar.

A utilização do novo padrão deve garantir duas coisas [5]:

- Se um provedor de dados combinar uma Ontologia espacial com uma Ontologia do seu domínio, os dados podem ser corretamente indexados e consultados em repositórios RDF espaciais;

- O provedor de dados deve ser capaz de processar a maioria dos dados RDF espaciais em conformidade com um repositório de triplos RDF.

Além de fornecer a capacidade de execução de consultas espaciais, a Ontologia da especificação do GeoSPARQL destina-se a proporcionar um formato de troca de dados geográficos numa variedade de casos de uso. A Ontologia está preparada para ser anexada a outras ontologias para vários domínios.

Como o GeoSPARQL define uma Ontologia que segue os padrões do OGC, permite resolver o problema de implementações diferentes e incompatíveis na representação e consulta de dados espaciais [15].

A especificação GeoSPARQL contém três componentes principais [5]:

- a definição de um vocabulário para representar recursos, geometrias e seus relacionamentos;

- um conjunto de funções espaciais, de domínio específico para utilizar em consultas SPARQL;

- e um conjunto de regras de transformação de consulta.

3.1.1. Ontologia GeoSPARQL

A Ontologia GeoSPARQL é baseada no modelo Simples de Recursos OGC, com algumas adaptações para RDF. Possui uma classe geo*:SpatialObject*, com duas subclasses principais, geo*:Feature* e *geo:Geometry*. Estas classes estão associadas a uma Ontologia que representa um domínio de interesse. Os recursos podem associar-se às suas geometrias através da propriedade *geo:hasGeometry*. Por exemplo, um aeroporto é um *geo:Feature* que consiste num objeto conceitual que existe no mundo real num determinado lugar. Esta localização no mundo real tem que ser medida e avaliada de algum modo. Uma representação de uma localização do mundo real que foi medida torna-se um

geo:Geometry. Assim, o aeroporto pode ser representado por diferentes *geo:Geometry*, que variam desde um único ponto representando o centro do aeroporto a um polígono detalhado que representa o contorno externo do aeroporto. Uma geometria que funcionará na maioria dos casos dentro de um conjunto de dados pode ser especificada como o *geo:defaultGeometry*.

O GeoSPARQL possui duas formas diferentes de representação de geometrias literais e hierarquia de tipos: o WKT (*Well-Known Text*) [16] e o GML (*Geography Markup Language*) [17]. Também fornece classes OWL para as hierarquias de geometrias associadas com as duas representações geométricas. Existem classes para vários tipos de geometria como ponto, polígono, curva, arco, e multi curva. As propriedades *geo:asWKT* e *geo:asGML* ligam as entidades geométricas com a representação geométrica literal.

3.1.2. Relacionamentos GeoSPARQL

O GeoSPARQL inclui também, um modo padrão para definir relações topológicas, tais como, a sobreposição entre entidades espaciais, disponibilizadas sob a forma de propriedades binárias entre as entidades obtidas recorrendo a funções de filtro geoespaciais. As propriedades topológicas binárias podem ser utilizadas em consultas SPARQL como uma propriedade normal. São utilizadas entre objetos do tipo *geo:geometry*.

As propriedades podem ser expressas usando três vocabulários distintos: O recurso simples *OGC* [18]*, "Egenhofer's 9-intersection model* [19]*"* e *"Region Connection Calculus (RCC8)* [20]*"*. O vocabulário a ser suportado depende da implementação da *triplestore*, embora seja provável que as implementações suportem todos os três. As relações topológicas de recursos simples são: *equals, disjoint, intersects, touches, within, contains, overlaps e crosses.*

As funções de filtro utilizam várias geometrias como predicados e produzem ou uma nova geometria ou outro tipo de dados como resultado. Um exemplo é o *ogcf:intersection*. Esta função recebe duas geometrias e retorna uma geometria que é a interseção espacial das duas geometrias iniciais. Outras funções como *ogcf:distance* produz um número real *xsd:double* como resultado.

3.1.3. Regras de transformação de consulta

As regras de reescrita de consulta permitem adicionar uma camada de abstração em consultas SPARQL. Em GeoSPARQL, as relações topológicas são realizadas pela combinação do uso da propriedade *geo:defaultGeometry* e as regras de reescrita de consultas. Se um *geo:Feature* é usado como sujeito ou objeto de uma relação topológica, a consulta é automaticamente reescrita para comparar a *geo:Geometry* ligada como um padrão, eliminando assim a abstração para processamento. O Código II-15 mostra uma consulta com uma relação entre *geo:Feature* de objetos antes e depois da reescrita.

```
# Antes
ASK
{
ex:DCA a geo:Feature;
     geo:within ex:Aveiro .
ex:Aveiro a geo:Feature .
}

# Depois
ASK
{
ex:DCA a geo:Feature;
     geo:defaultGeometry ?g1 .
ex:Aveiro a geo:Feature ;
     geo:defaultGeometry ?g2 .
?g1 geo:within ?g2 .
}
```

Código II-15: Exemplo de reescrita de consulta.

O objetivo desse recurso é fornecer uma abordagem mais intuitiva para consultas *geoespaciais*.

3.2. **Outra solução**

O trabalho apresentado em [13] descreve uma solução alternativa para a integração de dados espaciais em web semântica. Utiliza a terminologia de operações e funções espaciais padronizadas pelo OGC, onde são propostos termos padrão para formular regras em detrimento da utilização de termos baseados no domínio.

Para tal, é integrada na pilha da web semântica (exemplo Figura II-8) uma camada que contem informação espacial. Este ajustamento é feito através da inserção de dados espaciais juntamente com proposições semânticas na camada de Ontologia utilizando a sintaxe baseada em *OWL / RDF*. Esta camada utiliza a sintaxe padrão de *OWL / RDF*, por isso, permite executar consultas espaciais através de *SPARQL* ou inferir regras através de padrões como *SWRL*.

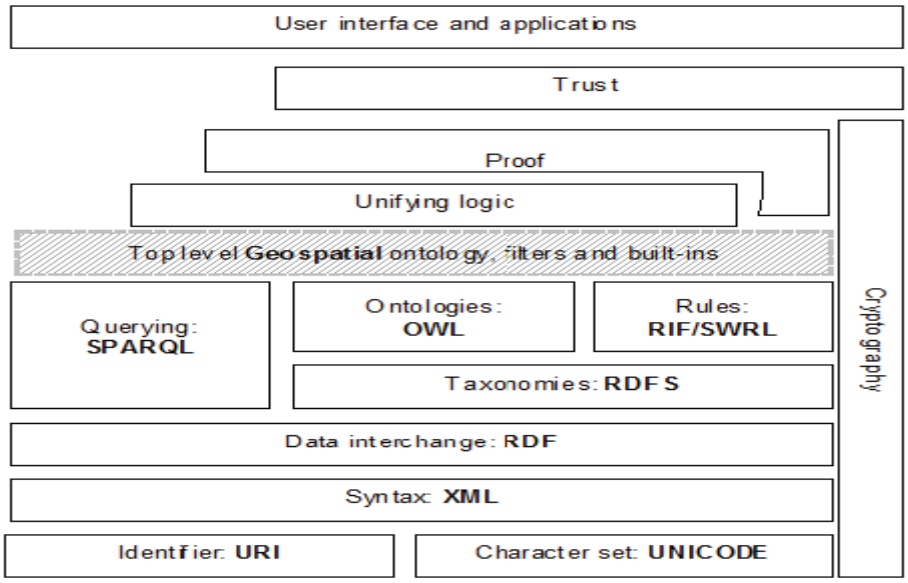


Figura II-8: Pilha da web semântica [13].

O processo de integração das tecnologias espaciais na pilha da web Semântica é feito através da definição de um novo tipo de filtros para consultas SPARQL e novos tipos de elementos embutidos *(Built-ins*) para regras SWRL. Estes novos filtros e *built-ins* permitem processar consultas e regras com dados geográficos relacionados com dados semânticos.

3.2.1. Ontologia do nível superior

Esta Ontologia fornece a base sobre a qual os objetos podem ser instanciados durante o processo de identificação de elementos espaciais. As classes são os blocos de construção da Ontologia, e portanto, essas classes, no contexto da Ontologia do nível superior da

aplicação, devem ser discutidas para fornecer uma visão geral do sistema. As principais classes desta Ontologia do nível superior são:

- Semântica - *spatial:Feature*;
- Geométrica - *shp:Shape*;
- Relações geométricas - *shp:hasShape*;
- Relações espaciais - *sa:hasSpatialRelations*;
- Relações espaciais da base de dados – *doc:hasDBDetails*.

A forma (geometria) tem uma representação numa base de dados espacial. Um indivíduo tem uma forma e tem relações espaciais com outros indivíduos que têm uma forma.

A classe *spatial:Feature* representa objetos espaciais. Esta é a classe generalizada de qualquer objeto com definição espacial. A classe *spatial:Feature* é uma classe abstrata, que não pode ser instanciada, mas possui subclasses a partir das quais é possível instanciar indivíduos que poderão ter um atributo espacial.

A classe *shp:shape* armazena geometria dos objetos. Esta classe genérica é especializada em subclasses *shp:_3D* e *shp:_2D* para representar objetos espaciais bidimensionais e tridimensionais.

A semântica dos objetos na base de conhecimento é definida através da propriedade do objeto *feat:objRel*. Antes os objetos relacionam-se com as suas assinaturas que são as coordenadas através da propriedade *shp:hasShape*. Por exemplo, o conceito de *win:Region* como uma subclasse de *spatial:Feature* tem a propriedade *shp:hasShape* que pode ser um *shp:_2D* ou *shp:_3D*.

As funções e operações espaciais devolvem geometrias como resultado de uma execução. A classe genérica *sa:spatialOperation* é introduzida no topo da Ontologia. As funções de processamento apresentadas em [13] são *Buffer, Union, intersection e Difference.* Para o efeito, são definidas novas subclasses da classe *sa:spatialOperation,* nomeadamente: *sa:sp_Buffer, sa:sp_Union, sa:sp_Intersection e sa:sp_Difference.* Estas classes são instanciadas quando a operação espacial desta categoria é chamada e o resultado é

armazenado dentro do indivíduo instanciado como propriedade de dados *feat:localPlacement*. As funções nesta categoria precisam de ter um objeto da classe *feat:Feature*. A propriedade do objeto *sa:hasSpatialRelations* é adicionada no topo da Ontologia para manter as relações entre operações espaciais que representam as subclasses da classe *sa:spatialOperation* e recursos da classe *feat:Feature* na Ontologia. As propriedades especializadas da propriedade *sa:hasSpatialRelations* relacionam indivíduos da classe *sa:spatialOperation* e classe *feat:Feature*. Por exemplo, cada instância da classe *sa:sp_Buffer* (subclasse de *sa:spatialOperation*) possui a propriedade *sa:hasBuffer* (propriedade especializada do objeto de *sa:hasSpatialRelations*) que relaciona a classe *sa:sp_Buffer* e as subclasses da classe *feat:Feature*.

As propriedades *sa:hasBuffer*, *sa:hasUnion*, *sa:hasIntersection*, *sa:hasDifference* são especializações da propriedade *sa:hasSpatialRelations* e correspondem a cada uma das funções de geoprocessamento. A Tabela II-5 mostra as funções de processamento espacial.

Funções	**Conceito**	**Propriedade de objetos**
Buffer	*Sa:sp_Buffer*	*Sa:hasBuffer(x,c)*
Union	*Sa:sp_Union*	*Sa:hasUnion(x,c)*
Intersection	*Sa:sp_Intersection*	*Sa:hasIntersection(x,c)*
Difference	*Sa:sp:Difference*	*Sa:hasDifference(x,c)*

Tabela II-5: Funções de processamento espacial.

A Tabela II-6 mostra as funções de georrelacionamento e suas características em web semântica.

Funções	**Propriedade de objetos**	**Características**
Disjoint	*Sa:hasDisjoint(x,y)*	Simétrico
Touches	*Sa:hasTouches(x,y)*	Simétrico
Within	*Sa:hasWithin(x,y)*	Transitivo
Overlaps	*Sa:hasOverlaps(x,y)*	
Equals	*Sa:hasEquals(x,y)*	Simétrico, Transitivo
Crosses	*Sa:hasCrosses(x,y)*	Simétrico
Intersects	*Sa:hasIntersects(x,y)*	Simétrico

Contains	*Sa:hasContains(x,y)*	Transitivo

Tabela II-6: Funções de georrelacionamento.

Se uma propriedade P é simétrica, e relaciona um indivíduo "x" ao indivíduo "y", então o indivíduo "y" também está relacionado ao indivíduo "x" através da propriedade P.

Por exemplo, se a freguesia Glória está relacionada com a freguesia Vera Cruz através da propriedade *Touches* (toca), então Vera Cruz também está relacionado a Glória através da propriedade *Touches*. Em outras palavras, se a freguesia Glória toca a freguesia Vera Cruz então Vera Cruz toca a freguesia Glória.

Se uma propriedade P transitiva relaciona o indivíduo "x" ao indivíduo "y", e também um indivíduo "y" ao indivíduo "z", infere-se que o indivíduo "x" está relacionado ao indivíduo "z" através da propriedade P. Por exemplo, se o município de Aveiro contém (contains) a freguesia Glória, e Glória contem uma secção 010501005, então o município de Aveiro contém a secção 010501005.

Estas funções demostram as relações espaciais entre objetos e são ajustadas como propriedades especializadas da propriedade *sa:hasSpatialRelations.* As funções *Disjoint, Touch, within* e *overlaps*, que são representados através das propriedades *sa:hasDisjoint*, *sa:hasTouch*, *sa:hasWithin* e *sa:hasOverlaps*.

3.2.2. Motor de tradução

O motor de tradução permite processar consultas espaciais SPARQL e regras espaciais SWRL. Em ambos casos, este interpreta as instruções a fim de analisar os componentes espaciais, utilizando funções e operações espaciais fornecidas ao nível de base de dados para processar as componentes espaciais analisadas. Depois as declarações espaciais são convertidas para declarações padrão para as execuções através do seu próprio motor SPARQL e o motor SWRL. Quanto ao motor de inferência, o enriquecimento e a população da Ontologia sobre os resultados do processo de inferência são armazenados na Ontologia. A Figura II-9 mostra o motor de tradução descrito.

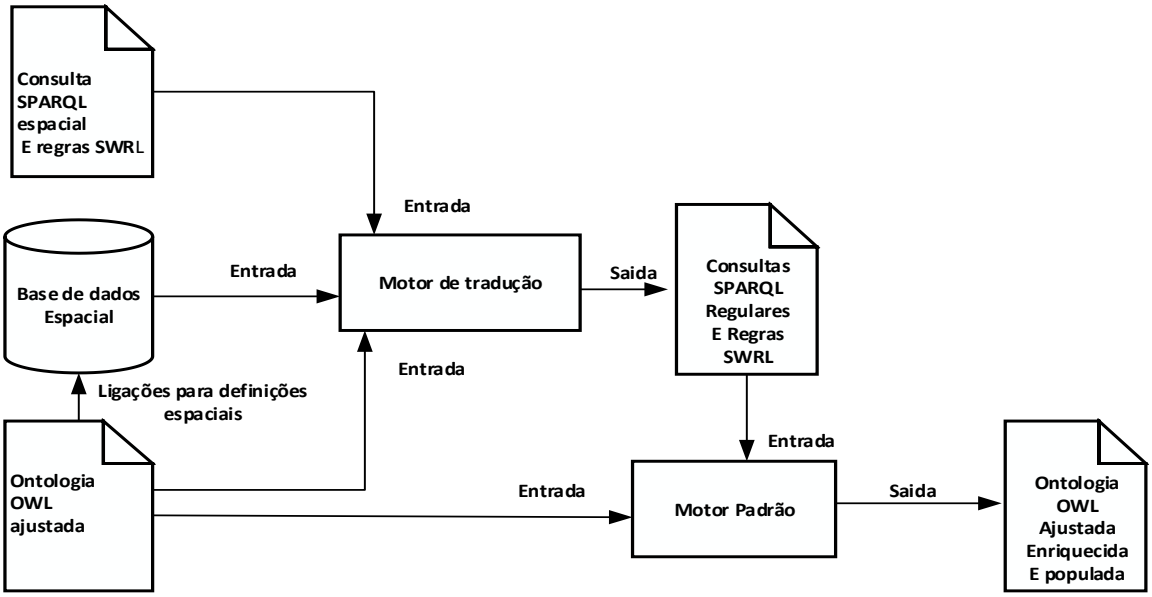


Figura II-9: Processamento espacial do motor de tradução traduzindo consultas SPARQL e regras OWL [13].

As secções seguintes apresentam em detalhe o mecanismo de tradução, nomeadamente, o processo de tradução de consultas espaciais SPARQL para consultas regulares e de regras SWRL espaciais para regras SWRL regulares. Ambos os processos utilizam instruções SQL para consultar a base de dados.

3.2.3. Consultas SPARQL espaciais

FILTER é utilizado como palavra-chave em consultas SPARQL para definir consultas espaciais. As funções com expressões regulares que correspondem a literais sem marcador de linguagem podem ser usadas para combinar as formas lexicais de outros literais usando a função de comparação de *strings*. Além disso, o SPARQL *FILTER* pode utilizar os operadores relacionais como "=", "> " ou " <" para comparar e restringir resultados. A ideia foi então utilizar o princípio do *FILTER* e fazer a sua extensão a fim de processar funções georrelacionais.

3.2.4. Aplicação do FILTER para o geoprocessamento

Conforme descrito na secção 1.3.2 deste capítulo, as operações de *geoprocessamento* retornam uma geometria que é resultado do processamento da geometria de um ou mais objetos. Recorrendo-se a um exemplo é demonstrada a sintaxe de filtro do processamento

geoespacial em SPARQL. Como pode ser visto no Código II-16 um novo filtro é introduzido através da palavra-chave SPATIAL_FILTER que ajuda o tradutor durante o processo de análise da consulta.

Exemplo: selecionar os edifícios localizados a uma distância inferior a 2km de um rio. A palavra-chave *FILTER* é substituída pela palavra-chave *SPATIAL_FILTER*. Código II-16 visualiza a solução SPARQL para devolver a resposta.

```
SELECT ?name1 ?name2
WHERE
{
?feat1 feat:name ?name1
?feat2 feat:name ?name2
?feat1 rdfs:type feat:River
?feat2 rdfs:type feat:Building
SPATIAL_FILTER [buffer (?x, 2 000,?feat1)]
SPATIAL_FILTER [intersection (?y,?x,?feat2)]
}
```

Código II-16:exemplo de geoprocessamento

O tradutor identifica o termo SPATIAL_FILTER durante o processo de tradução. Em seguida, determina a natureza das operações espaciais e executa-as. Neste caso particular, as operações espaciais *buffer* e intersecção são executadas e o resultado é inserido na base de conhecimento utilizando as classes *sa:sp_buffer* e *sa:sp_intersection.* A classe *sa:sp_buffer* possui indivíduos na zona do *buffer* de 2000 metros em volta do rio e a classe *sa:sp_intersection* possui indivíduos que são edifícios que intersectam esse *buffer*. Esses edifícios são relacionados com os rios correspondentes através da propriedade de objeto *sa:hasIntersection*. Assim que termina o processo de enriquecimento, o motor traduz as declarações GeosSPARQL para as declarações SPARQL padrão para executar a consulta. A declaração SPARQL traduzida é ilustrada no Código II-17.

```
SELECT ?name1 ?name2
WHERE
{
?feat1 feat:name ?name1
?feat2 feat:name ?name2
?feat1 rdfs:type feat:River
?feat2 rdfs:type feat:Building
?feat1 sa:hasBuffer ?x
```

```
?x rdfs:type sa:sp_buffer
?x sa:hasBufferDistance 2 000
?y rdfs:type sa:sp_Intersection
?y sa:hasIntersection ?x
?y sa:hasIntersection ?feat2
}
```

Código II-17: conversão para SPARQL padrão.

A Tabela II-7mostra a tradução de funções de geoprocessamento contidas em SPATIAL_FILTER no componente do triplo padrão de uma consulta SPARQL.

Funções	**Sintaxe SPARQL espacial**	**Tradução**
Buffer	SPATIAL_FILTER[buffer(?x,b,?y)] Resultado: inserido na base de conhecimento como indivíduos da classe sa:sp_Buffer.	?x rel:hasBuffer ?y ?y rdfs:type sa:sp_buffer ?y sa:hasBufferDistance 2 000
Union	SPATIAL_FILTER[union(?x,?y1,?y2)] Resultado: inserido na base de conhecimento como indivíduos da classe sa:sp_union	?x rdfs:type sa:sp_union ?x sa:hasUnion ?y1 ?x sa:hasUnion ?y2
Intersection	SPATIAL_FILTER[Intersection(?x,?y1,?y2)] Resultado: inserido na base de conhecimento como indivíduos da classe sa:sp_Intersection	?x rdfs:type sa:sp_Intersection ?x sa:hasIntersection ?y1 ?x sa:hasIntersection ?y2
Difference	SPATIAL_FILTER[difference(?x,?y1,?y2)] Resultado: inserido na base de conhecimento como indivíduos da classe sp_Difference	?x rdfs:type sa:sp_Difference ?x sa:hasDifference ?y1 ?x sa:hasDifference ?y2

Tabela II-7: Sintaxe SPARQL especial e sua tradução em SPARQL sintaxe

3.2.5. Aplicação do FILTER para georrelacionamento

O Código II-18 mostra como selecionar um conjunto de objetos ligados por uma relação espacial *Touch*. Os nomes dos objetos são selecionados de acordo com as duas primeiras

restrições na cláusula *Where*. O primeiro objeto é do tipo *feat:River* e o segundo do tipo *feat:Building*. O SPATIAL_FILTER seleciona objetos que se tocam espacialmente.

```
SELECT ?name1 ?name2
WHERE
{
?feat1 feat:name ?name1
?feat2 feat:name ?name2
?feat1 rdfs:type feat:River
?feat2 rdfs:type feat:Building
SPATIAL_FILTER [touches (?feat1,?feat2)]
}
```

Código II-18: Objetos ligados por uma relação espacial *Touch*.

O objetivo desta consulta é obter os pares de objetos do tipo *feat:River* e do tipo *feat:Building* que se tocam. A nova ligação é armazenada na Ontologia com ajuda da relação *sa:hasTouches* que é do tipo *sa:hasSpatialRelations*. O SPATIAL_FILTER é substituído pelo triplo "?feat1 *sa:touch* ? feat2 ". Assim, esta regra pode ser processada por um motor SPARQL padrão, conforme mostra o Código II-19.

```
SELECT ?name1 ?name2
WHERE
{
?feat1 feat:name ?name1
?feat2 feat:name ?name2
?feat1 rdfs:type feat:River
?feat2 rdfs:type feat:Building
?feat1 sa:touch ?feat2
}
```

Código II-19: Consulta SPARQL de georrelacionamento traduzida.

A Tabela II-8 mostra a tradução de funções georrelacionais contidas em SPATIAL_FILTER em triplos padronizados de uma consulta SPARQL.

Funções	**Sintaxe SPARQL espacial**	**Tradução**
Disjoint	SPATIAL_FILTER [Disjoint(?x, ?y)]	?x sa:hasDisjoints ?y
Touches	SPATIAL_FILTER [Touches(?x, ?y)]	?x sa:hasTouch ?y

Within	SPATIAL_FILTER [Within(?x, ?y)]	?x sa:hasWithin ?y
Overlaps	SPATIAL_FILTER [Overlaps(?x, ?y)]	?x sa:hasOverlaps ?y
Equals	SPATIAL_FILTER [Equals(?x, ?y)]	?x sa:hasEqual ?y
Crosses	SPATIAL_FILTER [Crosses(?x, ?y)]	?x sa:hasCrosses ?y
Intersects	SPATIAL_FILTER [Intersects(?x, ?y)]	?x sa:hasIntersects ?y
Contains	SPATIAL_FILTER [Contains(?x, ?y)]	?x sa:hasContains ?y

Tabela II-8: Funções georrelacionais contidas em SAPATIAL_FILTER

3.2.6. Otimização

O mecanismo de tradução é demorado para uma grande base de dados espacial. A fim de selecionar o contexto de execução, quatro opções podem ser dadas ao SPATIAL_FILTER.

- SPATIAL_FILTER_SELECT: Nenhuma operação espacial é realizada; a regra é traduzida, sem qualquer processamento espacial;
- SPATIAL_FILTER_PROCESS: operações espaciais são processadas apenas para conjunto de objetos que não têm essa relação. Se a relação já existir na base de conhecimento ela não é processada.
- SPATIAL_FILTER_UPDATE: operações espaciais são processadas apenas para conjunto de objetos que já possuem a relação, a fim de atualizar essas relações.
- SPATIAL_FILTER_ALL: Esta é a opção por padrão, que consiste em processar todas as relações a fim de analisar se a relação já existe na base de conhecimento ou se há necessidade de atualizá-lo.

O exemplo a seguir mostra uma seleção de objetos feita usando a opção SPATIAL_FILTER_UPDATE.

```
SELECT ?name1 ?name2
WHERE
{
?feat1 feat:name ?name1
?feat2 feat:name ?name2
?feat1 rdfs:type feat:River
?feat2 rdfs:type feat:Building
SPATIAL_FILTER [touches (?feat1,?feat2)]
SPATIAL_FILTER_UPDATE
}
```

Código II-20: exemplo de seleção dos objetos com relação Touch

3.2.7. Regras de inferência através de SWRL

No projeto de pesquisa do A.Karmacharya [13] foram definidos oito *built-ins* para SWRL. Esses *built-ins* refletem quatro funções espaciais de geoprocessamento e quatro funções de georrelacionamento que foram discutidos anteriormente.

Os *built-ins* para funções de georrelacionamento são apenas propriedades dos objetos e essas propriedades são utilizadas em colaboração com as funções espaciais no sistema de base de dados.

3.2.8. Geoprocessamento de elementos embutidos (built-ins)

Os *built-ins* refletem funções de geoprocessamento que retornam geometrias e são ajustados na Ontologia através da sequência: *feat:Feature, sa:hasSpatialRelations e sa:spatialOperation*.

Esta série de propriedades de classe é ilustrada na Tabela II-9. O passo inicial consiste em analisar os componentes internos para serem processados pelos motores de tradução. Primeiro os *built-ins* são identificados a partir da declaração e em simultâneo são também identificados os objetos em o que esses *built-ins* são aplicados. Depois, as instruções SQL com funções espaciais relevantes sobre os objetos são executadas a nível da base de dados. Os resultados são adicionados na base de conhecimento. Em seguida os *built-ins* são rescritos usando as *feat:Feature, sa:hasSpatialRelations e sa:spatialOperation* para gerar a declaração SWRL padrão que é executada através de mecanismos de inferência padrão. A Tabela II-9 mostra os *built-ins* de geoprocessamento.

Funções	**Classe**	**Propriedade do objeto**	**Propriedade do dado**	***Built-ins***
Buffer	sa:sp_Buffer	sa:hasBuffer	sa:hasBuffer Distance	Buffer(?x,b,?y)
Union	sa:sp_Union	sa:hasUnion	-	Union(?x,?y1,?y2)
Intersecti on	s:sp_Intersection	sa:hasIntesection	-	Intersection(?x,?y1, ?y2)
Differen ce	sa:sp_Difference	sa:hasDifference	-	Difference(?x,?y1,? y2)

Tabela II-9: *Built-ins* de geoprocessamento

A execução de todos os *built-ins* pode ser elaborada primeiro executando as operações espaciais e, em seguida, traduzindo as declarações com *built-ins* espaciais em declarações SWRL padrão. Por exemplo o Código II-21.

```
Feat:Feature(?x)^Buffer(?x, b, ?y)
```

Código II-21: exemplo de built-ins de geosprocessamento

Numa consulta contendo *built-in buffer,* primeiro é executada a instrução SQL com a função especial em cada objeto da classe que se pretende executar. Por exemplo se a declaração estiver relacionada com *bufferring* dos edifícios então, cada instância da classe *feat:edificio* será estendida (feita o *buffering*) através da execução de a instrução SQL. A instrução SQL com a função espacial *buffer* seria como no Código II-22.

```
SELECT Buffer (geom::Feature, bufferDistance)
```

Código II-22: instrução com a instrução buffer

O resultado da execução é adicionado a base de conhecimento. As linhas do resultado são geometrias que representam a extensão (*buffer)* de cada objeto. A classe *sa:sp_Buffer* é instanciada por objetos representando todas linhas e armazenando a geometria extendida, considerando a respetiva extensão (*buffer).* Em seguida as declarações com *built-in* espaciais são traduzidas em declarações SWRL padrão, como mostrado no Código II-23.

```
feat:Feature(?x)^sa:hasBuffer(?x,?y)^sa:sp_Buffer(?y)
^sa:hasBufferDistance(?y,b)
```

Código II-23: exemplo de *built-ins* traduzido em declarações SWRL padrão.

Os *built-ins* são convertidos na sequência *feat:Feature, sa:hasSpatialRelations* e *sa:spatialOperation* de declaração SWRL padrão.

Assim, a instrução converte o *built-ins* espacial em sequências *feat:Feature, sa:hasSpatialRelations e sa:spatialOperation* que são declarações SWRL padrão. A lista completa de execução de SQL, o enriquecimento do resultado e o processo de tradução da são ilustrados na Tabela II-10.

Built-ins	**Declarações SQL**	***Built-ins* traduzidos**	***Built-ins***
Swrlspatial:Buffer(?x,b?y)	SELECT Buffer(geom::Feature,BufferDistance) Resultado: Inserido na base de conhecimento como indivíduos da classe sa:sp_Buffer.	sa:hasBuffer(?x,?y) sa:sp_Buffer(?x,?y) sa:hasBufferDistance(?x,?y)	^Swrlspatial:Buffer(?x,b?y)
Swrlspatial:Union(?x,?y1,?y2)	SELECT Union(geom::Feature1,geom::Feature2) Resultado: Inserido na base de conhecimento como indivíduos da classe sa:sp_Union	sa:sp_Union(?x) sa:hasUnion(?x,?y1) sa:hasUnion(?x,?y2)	Swrlspatial:Union(?x,?y1,?y2)
Swrlspatial:Intersection(?x,?y1,?y2)	SELECT Intesection(geom::Feature1,geom::Featu	sa:sp_Intersection(?x) sa:hasIntersection(?x	Swrlspatial:Intersection(?x,?y1,?

	re2) Resultado: Inserido na base de conhecimento como indivíduos da classe sa:sp_Intersection	,?y1) sa:hasIntersection(?x,?y2)	y2)
Swrlspatial:Difference(?x,?y1,?y2)	SELECT Difference(geom::Feature1,geom::Feature2) Resultado: Inserido na base de conhecimento como indivíduos da classe sa:sp_Difference	sa:sp_Difference (?x) sa:hasDifference(?x,?y1) sa:hasDifference(?x,?y2)	Swrlspatial:Difference(?x,?y1,?y2)

Tabela II-10: Execução de instruções SQL de geoprocessamento de *built-ins* para o enriquecimento espacial.

Os componentes internos e a sua ligação com as propriedades do objeto são apresentados na Tabela II-11.

Funções	**Classe**	**Propriedade do objeto**	***Built-ins***
Disjoint	-	sa:hasDisjoint	Disjoint(?x,?y)
Touches	-	sa:hasTouch	Touch(?x,?y)
Within	-	sa:hasWithin	Within(?x,?y)
Overlaps	-	sa:hasOverlap	Overlaps(?x,?y)

Tabela II-11: *Built-ins* de georrelacionamento

A natureza do *built-ins* permite saber qual a operação espacial deve ser realizada ao nível da base de dados. Portanto as declarações devem ser analisadas para identificar os *built-ins* espaciais, depois a instrução SQL com a operação espacial relacionada é executada a nível de base de dados e as propriedades dos objetos na base de conhecimento são enriquecidas com base nos resultados obtidos. Os *built-ins* espaciais são convertidos em *feat:Feature* na

sequência *sa:hasSpatialRelations,* feat*:Feature* pelo mecanismo de tradução que agora é uma declaração padrão para que possa ser executado. A Tabela II-12 mostra as instruções SQL de *built-ins* de georrelacionamento.

Built-ins	**Declarações SQL**	***Built-ins* Traduzidos**
Swrlspatial:Disjoint(?x,?y)	SELECT Feature2 From spTable WHERE Disjoint(geom::Feature1, geom::Feature2)	*sa:hasDisjoint(?x,?y)*
Swrlspatial:Touches(?x,?y)	SELECT Feature2 From spTable WHERE Touch(geom::Feature1, geom::Feature2)	*sa:hasTouch(?x,?y)*
Swrlspatial:Within(?x,?y)	SELECT Feature2 From spTable WHERE Within(geom::Feature1, geom::Feature2)	*sa:hasWithin(?x,?y)*
Swrlspatial:Overlapst(?x,?y)	SELECT Feature2 From spTable WHERE Overlap(geom::Feature1, geom::Feature2)	*sa:hasOverlaps(?x,?y)*

Tabela II-12: Execução de instruções SQL de *built-ins* de georrelacionamento para o enriquecimento espacial

Por exemplo, considerar a função no Código II-24.

```
Feat:Feature(?x) ^ feat:Feature(?y)^ Touch(?x,?y)
```

Código II-24: exemplo de *Código* de georrelacionamento

Este exemplo determina se um objeto toca um outro. Geralmente as operações de georrelacionamento são binárias e retornam valores binários. Por exemplo, a operação *Touch* no comando do Código II-25, retorna verdadeiro ou falso determinando se a geometria de *Feature1* toca a geometria de *Feature2*.

```
SELECT Touch (geom::Feature1, geom::Feature2)
```

Código II-25: exemplo de função de georrelacionamento para obtenção do valor *booleano*

Mas se a mesma operação é executada como no exemplo do Código II-26 retorna a geometria de todos os *Feature2* que tocam num *Feature1*.

```
SELECT Feature2 FROM spTable WHERE Touch
(geom::Feature1,geom::Feature2)
```

Código II-26: exemplo de built-ins de georrelacionamento para filtro de linhas

A tabela *spTable* armazena as geometrias dos recursos que foram anotados espacialmente. Os resultados obtidos através da execução da instrução são enriquecidos pela propriedade *sa:hasTouch* do objeto especificado. Por fim, processa-se o *built-ins Touch (?x,?y)* na sequência *feat:Feature, sa:hasSpatialRelations* e *feat:Feature* para obter a declaração *SWRL* executada. A desagregação do *built-in Touch (? x,?y)* é dada como uma declaração SWRL padrão que pode ser novamente reutilizada pelo motor de inferência, como mostra o Código II-27.

```
feat:Feature(?x)^hasTouch(?x, ?y)^feat:Feature(?y)
```

Código II-27: desagregação do *built-ins* de Georrelacionamento.

4. Tecnologias utilizadas

Esta secção descreve as tecnologias utilizadas para o armazenamento de dados espaciais e dados semânticos, nomeadamente a base de dados *SQL-Server* e base de dados *Virtuoso*.

4.1. Tecnologias espaciais

Existem vários sistemas de gestão de bases de dados como o *Microsoft SQL-Server, IBM DB2, Oracle ou PostgreSQL,* que fornecem extensões para lidar com dados espaciais. O suporte de dados espaciais inclui os seguintes quatro itens:

- tipos espaciais nativos para representar as formas espaciais e objetos que podemos gerir em sistema de base de dados espaciais.
- operações espaciais, tais como testes de intersecção de dois objetos espaciais.

- acesso espacial ou métodos indexação para permitir processamento eficiente das operações espaciais sobre grandes volumes de dados espaciais.

- otimizações baseadas em custos para escolha de um plano de execução eficiente (frequentemente recorrendo a um índice espacial) para implementar as operações espaciais.

Para o presente trabalho foi utilizado o *SQL-Server* 2012 [21]. Esta plataforma fornece um mecanismo de armazenamento e recuperação seguro e fiável de dados, permitindo desta feita a criação e gestão de aplicações de dados de elevada disponibilidade e desempenho.

4.1.1. SQL-Server e dados espaciais

As principais características do *SQL-Server* 2012 são [22]:

- criar, implementar e gerir aplicações empresariais seguras, escaláveis e fiáveis;

- maximizar a produtividade das Tecnologias de Informação (TI) através da redução da complexidade do desenvolvimento e do suporte de aplicações de base de dados;

- controlar custos sem sacrifício do desempenho, da disponibilidade, da escalabilidade ou da segurança.

O SQL-Server suporta dados espaciais para gestão de dados de conhecimento geográfico. Em particular, possui dois tipos de elementos embutidos para representar dados geográficos bidimensionais num sistema de referência planar ou geodésico.

Os dados num sistema de referência planar são representados por um novo tipo de dado chamado *geometry* que pode representar os pontos, linhas, polígonos e combinações dos anteriores num plano bidimensional cartesiano.

Os dados num sistema de referência geodésico são representados por um novo tipo de dados chamados *geography*, que também representa pontos, linhas e polígonos num plano bidimensional definido sobre a superfície de uma geoide que se aproxima do formato da terra.

Ambos tipos suportam uma infinidade de operações espaciais, incluindo composição de objetos e predicados de relacionamento espacial. O conjunto de funcionalidades fornecidas pelos tipos espaciais obedece a padrões OGC [23].

A fim de aplicar de forma eficiente predicados da relação espacial, *SQL-Server* também oferece uma infraestrutura de indexação e implementa extensões para o processamento de consultas que permite ao otimizador tomar decisões baseadas em custos para escolher o plano de execução adequado se um ou mais índices estiverem presentes.

4.2. Tecnologias semânticas

Nesta secção são abordadas as diferentes formas e formatos de armazenamento de dados semânticos, considerando as vantagens de cada uma. Também são comparados os diferentes tipos de repositórios de triplos (*triplestores*).

4.2.1. Armazenamento de arquivos RDF

Para gravar arquivos RDF como uma sequência de bytes em disco é utilizado o termo técnico: serialização. As serializações são geralmente feitas em ficheiros de texto, que podem utilizar diferentes sintaxes para representar triplos. Um dos formatos mais utilizados, devido à sua maior legibilidade para o ser humano, é o *Turtle.*

O formato mais simples é chamado *N-Triplos*. Neste formato, todos os URIs são escritos na sua totalidade, sem abreviações, e dentro de parênteses angulares e as *strings* são escritas dentro de aspas. Cada triplo é escrito na sua própria linha e termina sempre com um ponto no final, como mostra o Código II-28.

```
<urn:isbn:6251587X> <http://creator> <http://card#i> .

<urn:isbn:006251587X> <http://title> "Weaving the Web" .

<http://card#i> <http:/title> "Director" .
```

Código II-28: Exemplo de declarações de triplos em formato N-Triplos.

A serialização mais antiga, RDF/XML, fazia parte da especificação original RDF em 1999.

O Código II-29 mostra triplos escritos no formato RDF/XML.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:v="http://www.w3.org/2006/vcard/">

<rdf:Description rdf:about="urn:isbn:006251587X">
<dc:title>Weaving the Web</dc:title>
<dc:creator rdf:resource="http://www.w3.org/People/Berners-
Lee/card#i"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.w3.org/People/Berners-
Lee/card#i">
<v:title>Director</v:title>
</rdf:Description>

</rdf:RDF>
```

Código II-29: Exemplo de declarações de triplos em RDF/XML

O formato RDF/XML nunca se tornou popular devido à complexidade e dificuldades de processamento que apresenta. Outro problema são as limitações impostas pelas regras de nomeação de elementos XML.

O formato de serialização N3 (abreviatura de "Notação 3"), proposto por Tim Berners-Lee, combina a simplicidade de *N-Triplos* com a capacidade RDF/XML de dividir URIs longos com prefixos. O Código II-30 mostra uma declaração RDF utilizando a anotação N3.

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<urn:isbn:0123735564> dc:creator
<http://www.topquadrant.com/people/dallemang/foaf.rdf#me> ,
<http://www.cs.umd.edu/~hendler/2003/foaf.rdf#jhendler> .
```

Código II-30 Declaração RDF com anotação N3

4.2.2. Armazenar RDF em bases de dados

Conforme já visto anteriormente, os triplos podem ser armazenados em ficheiros de texto. Porém quando o volume de dados é maior esta solução não é viável, visto que não possui mecanismos para indexar os dados. Neste caso, a melhor solução seria a utilização de

sistemas de gestão de bases de dados que permitisse indexar os dados e decidir que dados carregar em memória.

Os dados RDF podem ser armazenados numa base de dados relacional, como MySQL ou Oracle, por exemplo, contudo é sempre melhor optar por uma solução otimizada para armazenamento de triplos. Em geral, dá-se o nome de repositório de triplos (*triplestore*) a este tipo de soluções.

Ao avaliar-se as potencialidades de uma *triplestore*, apoiando-se nas questões típicas da gestão de base de dados, como tamanho, velocidade, disponibilidade e custo; há várias questões relacionadas com SPARQL a serem consideradas:

- se suporta a linguagem de consulta ou atualização SPARQL que os desenvolvedores de *triplestore* criam;
- com que facilidade a *triplestore* recebe a consulta SPARQL e em seguida devolve o resultado, interactivamente ou programaticamente;
- se a *triplestore* pode servir como um *endpoint* SPARQL;
- se a *triplestore* suporta o padrão SPARQL mais recente.

4.2.3. Repositórios de triplos com dados espaciais

O suporte para dados espaciais em *triplestores* é diversificado. Vários fornecedores suportam dados espaciais, mas nem todos os fornecedores suportam a mesma representação de dados ou partilham o mesmo suporte para consultas relacionais. Algumas *triplestores* usam as ontologias W3C, acima mencionadas, enquanto outras usam as suas próprias.

A *triplestore* utilizada neste trabalho é o *openLink Virtuoso* [24]. Apesar de esta *triplestore* ter capacidade para armazenar e manipular dados espaciais possui algumas limitações, abaixo indicadas. Neste trabalho, esta *triplestore* foi utilizada apenas como um repositório de dados semânticos.

OpenLink Virtuoso é um servidor hibrido de SQL-ORDBMS e de aplicações *web* fornece a gestão de dados RDF, XML e SQL num processo servidor. O acesso a *triplestore* está

disponível via SPARQL, SIMILE *Semantic Bank* API, ODBC, GRDDL, JDBC, ADO.NET, XMLA, WebDAV, e *Virtuoso*/PL (SQL *Stored Procedure Language*). O produto está disponível nas edições opensource e nas edições comerciais. Para este trabalho foi utilizada a edição *opensource*.

O OpenLink Virtuoso também tem suporte para o vocabulário básico W3C. Uma função SPARQL é fornecida para converter um pare de valores latitude e longitude num ponto geométrico. O tipo de dados literal espacial *virtrdf:Geometry* permite a indexação de pontos literais. Possui funções que permitem testar se um objeto contém ou intersecta outro. Através de uma combinação dessas relações e suas negações, a maior parte das relações *Egenhofer* podem ser testadas. No entanto, algumas relações, tais como testes para sobreposição, não são suportados em *Virtuoso*.

O *Parliament* [25] é um repositório de triplos de alto desempenho da *Raytheon BBN Technologies* e forneceu o primeiro índice geoespacial para web semântica em 2007. Este índice suporta dados na Ontologia GeoOWL e introduz ontologias para consultar dados espaciais via propriedades RDF que correspondem às relações espaciais *RCC* e relações simples de recursos *OGC*. No entanto, dados no vocabulário básico *W3C* não são suportados.

O *Ontotext OWLIM-SE6* [26] repositório de triplos pode indexar pontos de dados representados no Vocabulário *W3C Geo básico*. A única relação espacial que pode ser consultada é saber se um ponto está contido dentro de um círculo ou polígono. A capacidade de consulta para as relações entre geometrias de ordem superior, tais como linhas e polígonos não é suportado.

O *OpenSahara9* [27] fornece um serviço para a adição da indexação externa e capacidades de consulta geoespacial para qualquer repositório de triplos. A implementação é um *wrapper* para a base de dados *PostgreSQL* [28] com extensões espaciais *PostSIG* [29] , como tal, o *OpenSahara* suporta todas as geometrias e relações definidas no acesso de recursos simples OGC. Tipos de dados literais são introduzidos utilizando a linguagem de marcação de texto *Well-Known Text* (WKT), enquanto as relações espaciais que *PostSIG* suporta são implementadas como funções de filtro *SPARQL*.

III Modelação e Implementação do sistema

Neste capítulo são apresentados os requisitos, a modelação e a implementação do sistema que foi desenvolvido neste projeto para demonstração da possibilidade de integração de dados espaciais em sistemas de web semântica, utilizando a abordagem descrita no capítulo anterior na secção 3.2.

1. Problema

Conforme descrito no capítulo II secção 4.2.3 existem muitas *triplestores* que utilizam ontologias e predicados para tornarem possível a indexação e execução de consultas espaciais. Porém cada uma dessas soluções apresenta limitações no armazenamento e processamento dos dados espaciais, tal é o caso da base de dados *Virtuoso*, a *triplestore* utilizada neste trabalho, não suporta algumas relações, tais como relações de sobreposição, para além de exigir curvas de aprendizagem bastante esforçadas, visto que, apenas possui funções que permitem testar se um objeto contém ou intersecta o outro. Outras relações só podem ser testadas a partir das combinações e negações das funções "contém" e "intersecta". Por outro lado os dados RDF podem ser armazenados numa base de dados relacional mas estas bases de dados não estão otimizadas para armazenamento de triplos. Por isso no sistema proposto os dados são armazenados e processados nas suas bases de dados nativas, isto é, dados espaciais numa base de dados espacial e dados semânticos na base de dados semântica.

Todas as *triplestores* são ligeiramente diferentes, os dados espaciais que podem ser consultados numa *triplestore* podem não o ser em outra. Por isso a adoção de uma solução que segue um padrão para armazenamento e processamento RDF pode ser a melhor escolha.

Para o desenvolvimento do sistema proposto era possível a adoção de uma das duas soluções propostas e descritas antes no capítulo anterior, a saber: o padrão GeoSPARQL e a proposta do projeto de pesquisa, descrito antes.

Para a escolha de uma das soluções, foram tidas em conta algumas considerações, como: o nível atual de adoção por soluções já existentes; o seu nível de especificação; o seu grau de dificuldade ou simplicidade na implementação; e as suas principais diferenças ou semelhanças.

Ambas as soluções apresentam semelhanças ao nível:

- da utilização do padrão OGC. As regras são formuladas utilizando termos padrão e não termos orientados ao domínio, o que permite ter representações geoespaciais lógicas, consistentes e comprensíveis na web semântica;

  da necessidade do processamento de consultas SPARQL espaciais e regras SWRL espaciais, no qual depois da interpretação das instruções, a fim de analisar as declarações espaciais, é necessária a rescrita das consultas SPARQL espaciais para consultas SPARQL padrão.

Para o desenvolvimento de um sistema, a proposta do projeto de pesquisa apresenta vantagens, visto que, descreve ao pormenor o processo de tradução da consulta, nomeadamente a consulta espacial gerada, a rescrita da consulta SPARQL espacial para consulta SPARQL padrão e apresenta mecanismos de otimização da consulta SPARQL espacial, conforme descrito no capítulo II secção 3.2.6, o que permite criar uma implementação mais correta.

Quanto ao nível de adoção do GeoSPARQL, a Parliament é a única *triplestore* que o utiliza. O que pode significar que o padrão pode ainda estar em fase de estabilização (imaturo).

2. Requisitos

No presente trabalho pretendia-se a criação de um sistema capaz de receber, processar e oferecer resultados a pesquisas, envolvendo dados semânticos e espaciais. Os diferentes tipos de dados deveriam residir em repositórios diferentes, mais adequados a cada um dos tipos. A receção de pesquisas e a devolução de resultados deveria ser feita de forma integrada, escondendo completamente todo o trabalho necessário ao processamento diferenciado dos dois tipos de dados.

2.1. Casos de uso

A Figura III-1 representa os requisitos funcionais do sistema na forma de casos de uso. O grau de urgência com que os casos de uso devem ser implementados é igual, visto que, o funcionamento do sistema só faz sentido se todos estiverem a funcionar.

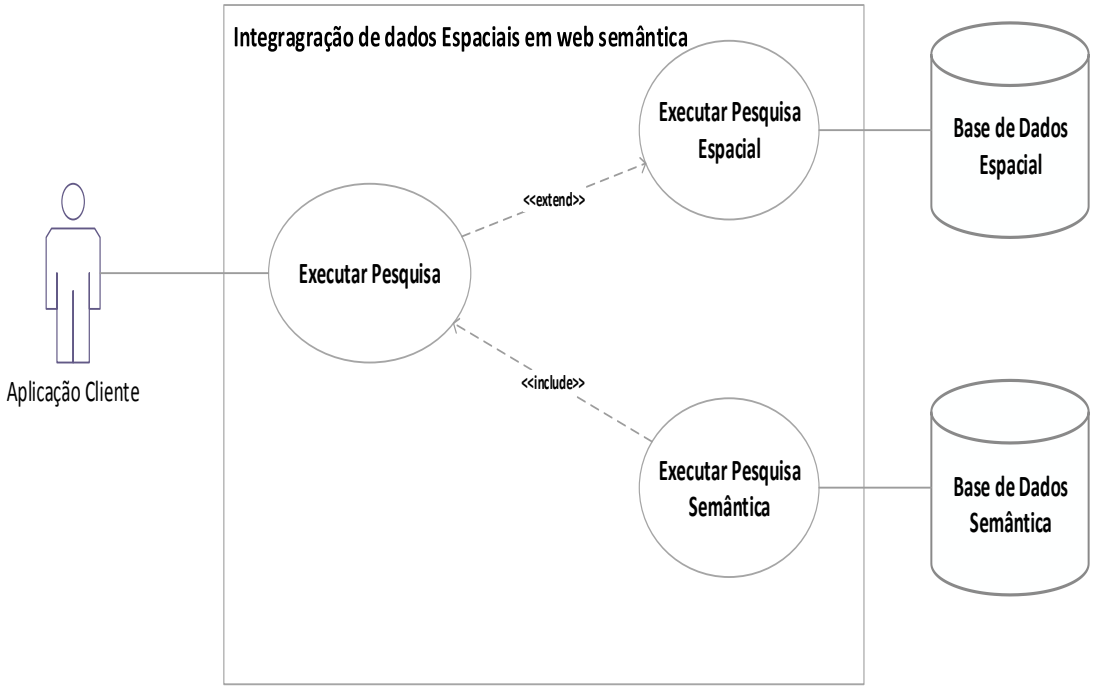


Figura III-1:Diagrama de casos de uso

Descrição de casos de uso.

Executar Pesquisa

Atores: Aplicação cliente.

Tipo: Funcionalidade.

Pré-condição: Deve existir um mecanismo de interpretação dos dados para que as pesquisas da aplicação cliente sejam percebidas pelo serviço, e que permita a este gerar pesquisas corretas para os repositórios semântico e espacial.

Descrição: "Executar Pesquisa" é o caso de uso principal. Este pressupõe operações que envolvem ambos os repositórios semânticos e espacial ou apenas o semântico. O resultado

deste caso de uso é uma resposta única à consulta, integrando dados semânticos e espaciais, caso necessário.

Pós-condição: Resultado da consulta enviado ao cliente.

Executar Pesquisa Semântica

Tipo: Funcionalidade

Pré-condição: Os dados e variáveis fornecidos como argumentos devem ser válidos, ou seja devem constituir predicados válidos no repositório semântico.

Descrição: Este caso de uso é invocado pelo caso de uso "Executar Consulta" e retorna o resultado de acordo com os dados de pesquisa fornecidos pela aplicação cliente.

Pós-condição: Resultado da pesquisa semântica integrado na resposta.

Executar Pesquisa Espacial

Tipo: Funcionalidade.

Pré-condição: Os dados e variáveis fornecidos como argumentos pela aplicação cliente devem ser válidos no repositório espacial.

Descrição: O caso de uso em causa é invocado pelo caso de uso "Executar Pesquisa" e devolve um resultado de acordo com os dados de pesquisa fornecidos pela aplicação cliente.

Pós-condição: Resultado da pesquisa integrado na resposta.

2.2. Diagrama de domínio

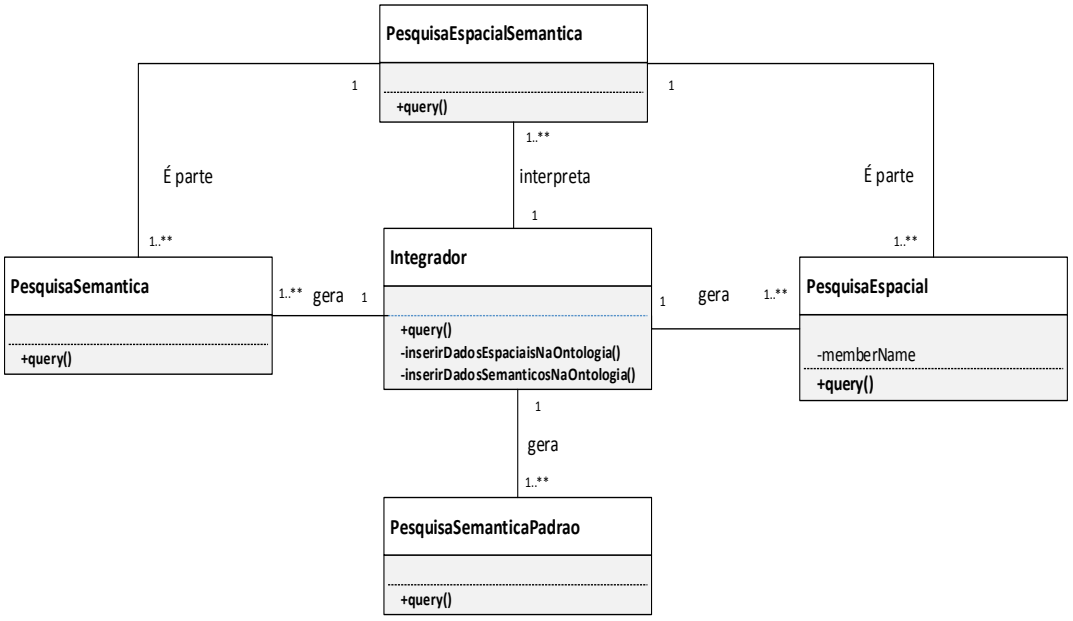


Figura III-2: Diagrama de domínio.

A Figura III-2 representa o diagrama de domínio, contendo as entidades principais constituintes do sistema, as quais se passam a descrever:

PesquisaEspacialSemantica – entidade responsável por receber a pesquisa enviada pela aplicação cliente.

> **Método *query()*** recebe e envia a pesquisa a entidade integrador.

Integrador – entidade responsável por integrar as pesquisas semânticas e espaciais num único repositório e devolver a resposta à aplicação cliente.

> **Método *query()***– interpreta a pesquisa da aplicação cliente, envia dados à pesquisa espacial e semântica e integra os respetivos resultados. Este método é ainda responsável por enviar dados à pesquisa espacial semântica padrão e retornar a resposta desta à aplicação cliente.

PesquisaEspacial – entidade responsável por proceder à pesquisa no repositório espacial, esta pesquisa faz parte da pesquisa iniciada pela entidade **PesquisaEspacialSemantica.**

> **Método *query()*** - gera a pesquisa espacial e retorna a resposta ao integrador.

PesquisaSemantica – entidade responsável por realizar a pesquisa no repositório semântico, esta pesquisa, também, faz parte da pesquisa iniciada pela pesquisa **PesquisaEspacialSemantica**.

Método *query()* - gera a pesquisa semântica e devolve a resposta ao integrador.

PesquisaSemanticaPadrao – entidade responsável por proceder à pesquisa no repositório do integrador.

Método *query()* – gera a pesquisa que é executada sobre o repositório do integrador.

3. Desenho do sistema e implementação

De seguida são abordados os processos de desenho do sistema, identificando-se com elevado detalhe o modo como os requisitos devem ser satisfeitos do ponto de vista técnico.

3.1. Arquitetura do sistema

A Figura III-3 permite visualizar a arquitetura do sistema proposto.

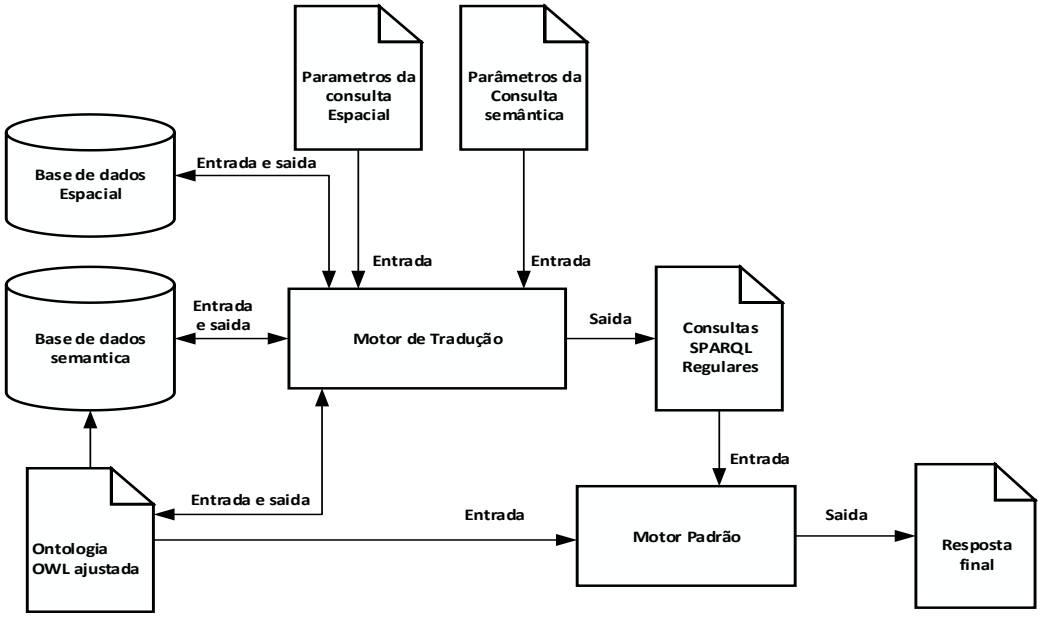


Figura III-3: Arquitetura do sistema proposto baseado A.Karmacharya.

O motor de tradução recebe dois tipos de parâmetros. Um conjunto de parâmetros para consulta espacial e outro para consulta semântica. Os parâmetros para consulta espacial são interpretados pelo motor de tradução a fim de apurar que funções e operações espaciais invocar na base de dados espacial. O resultado desta consulta é convertido em dados semânticos e inserido num repositório semântico temporário. A consulta semântica é executada na base de dados semântica e o seu resultado também é inserido no repositório semântico temporário. É executada uma consulta SPARQL regular através do motor SPARQL padrão e por fim a resposta é enviada à aplicação cliente.

3.2. Diagrama de pacotes

A Figura III-4 mostra os pacotes identificados para criação do sistema.

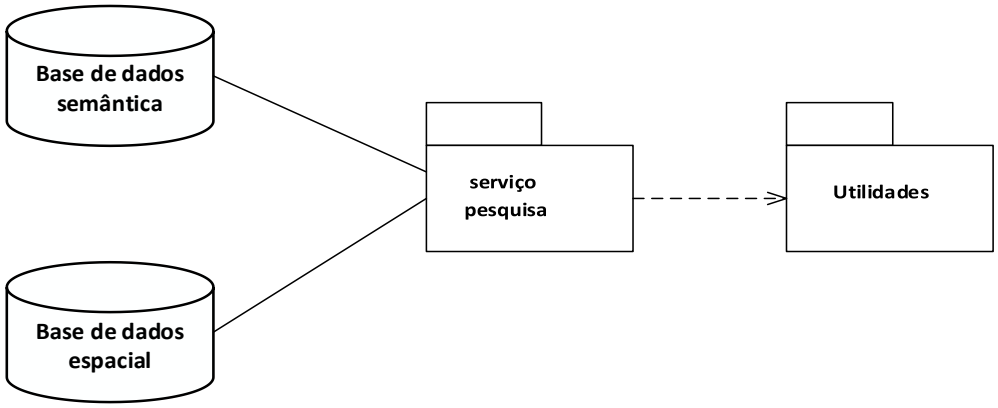


Figura III-4:Diagrama de pacotes.

Pacote serviço de pesquisa - contém classes desenvolvidas para tornar o serviço funcional, isto é, fazer pesquisas e devolver a resposta à aplicação cliente.

Pacote de utilidades - possui classes utilitárias que são utilizadas pela aplicação cliente.

Base de dados semântica - é um repositório de dados que armazena dados semânticos.

Base de dados espacial - é um repositório de dados que armazena dados espaciais.

3.3. Diagrama de sequências

O diagrama de sequência na Figura III-5Figura IV-1, mostra a ordem cronológica da interação entre objetos para realização da consulta solicitada pela aplicação cliente.

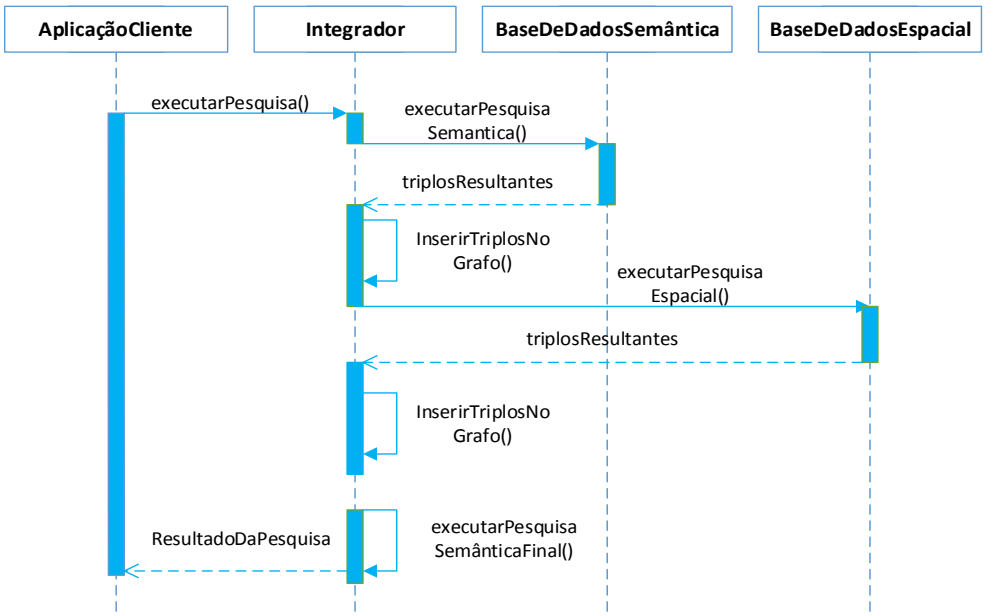


Figura III-5:Diagrama de sequências

Para a execução da pesquisa da aplicação cliente as operações são desencadeadas na seguinte ordem:

Método *executarPesquisa()* – invoca o método ***executarpesquisasemantica()*** e passa lhe os nomes dos tipos de dados e atributos necessários para responder a pergunta da aplicação cliente.

Método *executaPesquisaSemantica()* – gera uma pesquisa semântica, executa-a e devolve a resposta da pesquisa ao integrador na forma de triplos.

Método *inserirTriplosNoGrafo()* – insere os triplos gerados no grafo.

Método *executaPesquisaEspacial()* – gera uma pesquisa espacial, executa-a e retorna a uma resposta na forma de triplos ao integrador.

Método *executaPesquisasemanticafinal()* – estando os resultados das pesquisas semântica e espacial contidos no grafo. O método pesquisa semântica final faz a pesquisa semântica sobre o grafo e retorna a resposta a aplicação cliente.

3.4. Diagrama de classes

A Figura III-6 representa o diagrama de classes da fase de desenho em que as classes dos objetos são detalhadas, isto é, são apresentados os métodos e atributos para o acesso a base de dados semântica e base de dados espacial e a estrutura de troca de mensagens entre os objetos.

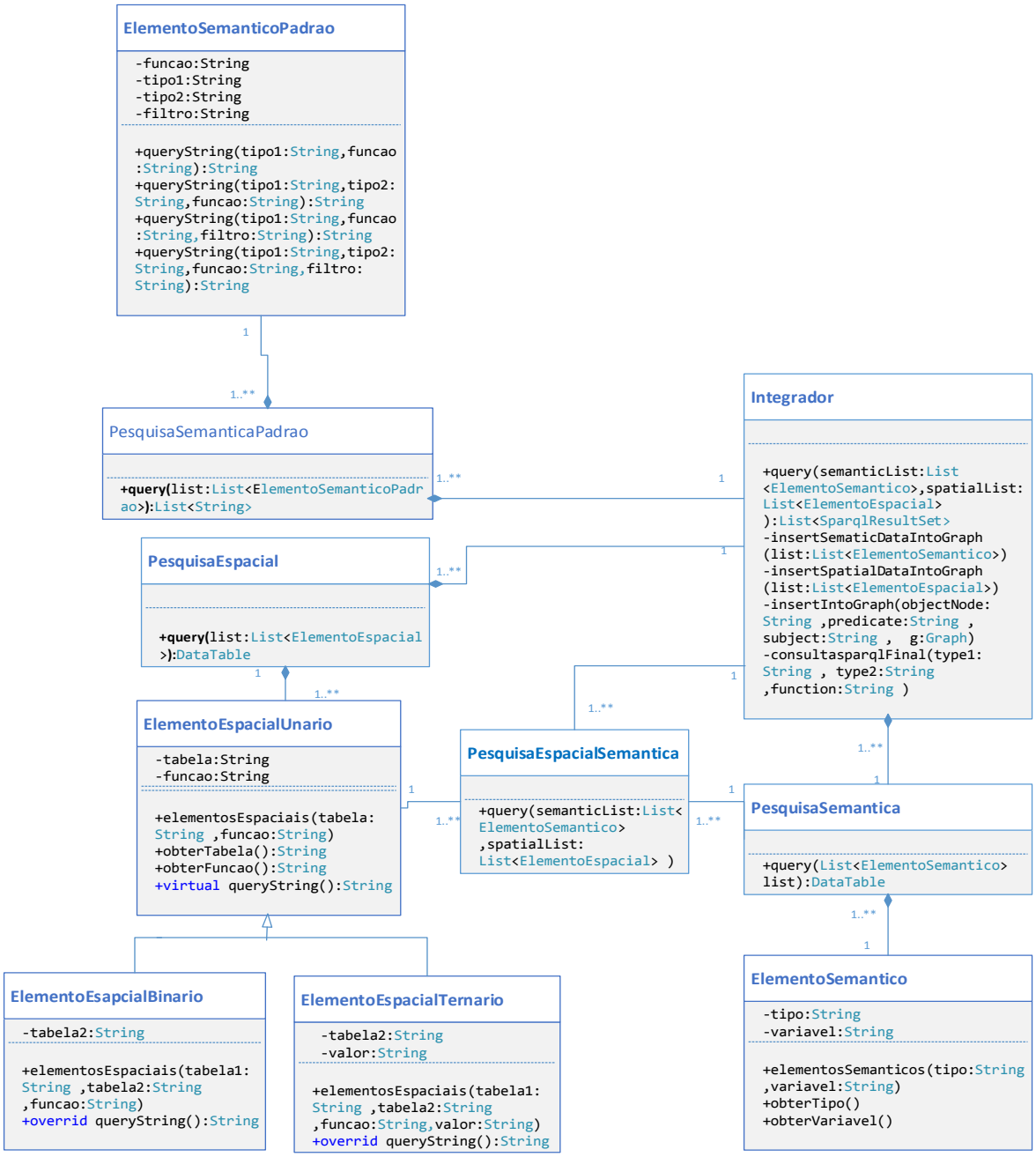


Figura III-6:Diagrama de classes.

Classe PesquisaEspacialSemantica - responsável por enviar perguntas da aplicação clientes ao integrador e receber as respetivas respostas.

Método *query()* - recebe duas listas, uma contendo dados semânticos e outra dados espaciais.

Integrador – é a classe encarregue de integrar pesquisa semântica e espacial num único repositório (grafo) e devolver a resposta à aplicação cliente.

> **Método *query()*** - recebe do método ***query()*** da classe **PesquisaEspacial Semantica** duas listas, uma com elementos semânticos e outra com elementos espaciais que permitem gerar Pesquisas semânticas e espaciais respetivamente. Os resultados de ambas pesquisas são inseridos num grafo (repositório) criado nesta classe.

Classe PesquisaSemantica – esta classe realiza a pesquisa no repositório semântico.

> **Método *query()*** - recebe do método ***query()*** da classe **Integrador,** uma lista de dados semânticos, gera uma pesquisa semântica e devolve como resposta, uma tabela contendo triplos constituídos por código, tipo e atributo do objeto

Classe ElementoSemantico – é responsável por criar dados semânticos utilizados pela classe **PesquisaSemantica** para gerar uma pesquisa de acordo com os argumentos fornecidos pela aplicação cliente.

Classe PesquisaEspacial - faz a pesquisa espacial no repositório espacial em conformidade com os argumentos passados pelo integrador.

> **Método *query()* –** recebe do método ***query()*** da classe **Integrador,** uma lista de dados espaciais, gera uma pesquisa espacial e devolve como resposta, uma tabela contendo triplos.

Os **elementos espaciais** apresentam três versões para acomodar três casos de aplicação de operações e funções espaciais.

ElementoEspacialUnario – gera pesquisas que permitem obter informação de medição de um dado geométrico. O Código III-1 mostra um exemplo de funções espaciais suportadas:

```
(ST_Area/ST_Length(geom).
```

Código III-1: exemplo da função espacial suportada pelos métodos da classe ElementoEspacialUnário.

> **Método *queryString()*** - foi declarado com modificador virtual o que indica que pode ser sobrescrito na classe derivada.

ElementoEspacialBinario - esta classe, cria pesquisas com predicados espaciais que determinam relacionamentos topológicos entre duas geometrias. O Código III-2 mostra Exemplo de funções suportadas nesta classe.

```
ST_Disjoints((geom1),(geom2));
ST_Equals((geom1),(geom2));
ST_Touches((geom1),(geom2));
ST_Within((geom1),(geom2));
ST_Overlaps((geom1),(geom2));
```

Código III-2: exemplo de funções suportadas pela ElementoEspacialBinário.

ElementoEspacialTernario – gera pesquisas que determinam relacionamentos topológicos entre um objeto espacial normal e um outro transformado. O exemplo de funções suportadas é mostrado no Código III-3.

```
(ST_Disjoints/ST_Equals/ST_Touches/ST_Within/ST_Overlaps((ST_
Buffer(geom)), (geom2) ).)
```

Código III-3: exemplos de funções suportadas pela classe ElementoEspacialTernário.

As classes derivadas **ElementoEspacialBinaria e ElementoEspacialTernario** subescrevem o método ***queryString()***, permitindo desta forma a utilização do comportamento do polimorfismo para se devolver a consulta certa que é deduzida a partir dos dados de entrada da classe.

PesquisaSemanticaPadrao – classe responsável por executar a pesquisa final, cujo resultado é enviado como resposta à aplicação cliente. Esta pesquisa é executada sobre o grafo criado na classe **Integrador.** O grafo contém resultados das pesquisas semânticas e espaciais. Cada pesquisa da lista é gerada em função dos dados de entrada que são passados ao método ***queryString().*** Na classe **ElementoSemânticoPadrão** é feito o *overload* do método ***queryString()*** para permitir que o resultado seja retornado de acordo com os parâmetros passados pela aplicação cliente.

3.5. Implementação do sistema proposto

Para o desenvolvimento do sistema proposto foram utilizadas as seguintes tecnologias:

A base de dados *SQL-Server* como base de dados espacial, *OpenLink Virtuoso* para base de dados semântica e C# como linguagem de programação.

A linguagem de programação C# utiliza bibliotecas que são oferecidas pelo *dotNetRDF* que é uma API (*Application Programming Interface*) poderosa e flexível para trabalhar com RDF e SPARQL em ambiente *dotNet*. Esta API fornece um conjunto de bibliotecas e ferramentas *dotNet*, *open source* e escritas em C #, para trabalharem com RDF, SPARQL e Web Semântica na plataforma *dotNet*. O objetivo principal do *dotNetRDF* é fornecer uma forma eficaz para operar com quantidades razoáveis do RDF em *dotNet*.

O *dotNetRDF* oferece suporte para várias *triplestores*, tal é o caso do *OpenLink Virtuoso*. Esta para além do suporte do *dotNetRDF* foi desenvolvida para permitir acesso via *Ado.net,* o que permite que as linguagens de programação *dotNet* tenham múltiplas formas de acesso a *triplestore openLink Virtuoso*.

Na base de dados espacial as tabelas devem ser constituídas apenas pelos dados geométricos que representam os objetos e os códigos que os identificam.

Na base de dados semântica deve ser criada uma estrutura hierárquica dos objetos manipulados pelo sistema, nomeadamente atributos*, redfs:type* e operações espaciais e cada uma dessas subclasses pode ser subdivida para armazenar os objetos em classes mais apropriadas, conforme mostra a Figura III-7.

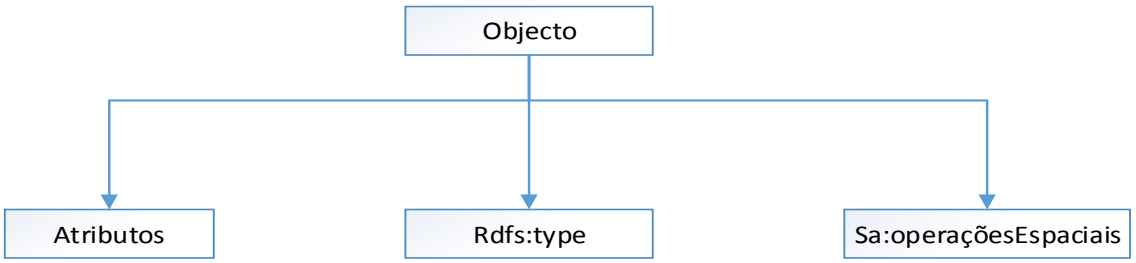


Figura III-7 Estrutura hierárquica básica do objeto

Todas as classes e os métodos descritos na secção 3.4 são implementados utilizando a linguagem de programação C#.

Este trabalho é uma implementação da descrição do projeto de investigação do A.Karmacharya [13]. Sobre a arquitectura do projeto de investigação é criada uma adaptação que consiste em dividir a pesquisa SPARQL espacial em duas, isto é, em pesquisa semântica e pesquisa espacial antes da sua execução. É também criado um repositório semântico temporário para evitar a sobre do sistema, eliminando os resultados das pesquisas espaciais quando estes já não forem necessários.

Todas as pesquisas são geradas ao nível da aplicação e executadas nos seus sistema de armazenamento e processamento de dados, isto é, as pesquisas semânticas são executadas na base de dados semântica e as pesquisas espaciais na base de dados espacial, os resultados destas são enviados na forma de triplos para o repositório semântico temporário, onde são integrados com outros dados semânticos necessário para a execução da pesquisa do cliente. A pesquisa semântica final é gerada na aplicação de acordo com os parâmetros passados pela aplicação cliente e executada no repositório temporário semântico e a resposta é enviada a aplicação cliente.

IV Caso de estudo

Para demonstrar as potencialidades do sistema proposto, desenvolveu-se um caso de estudo para a caracterização de zonas territoriais no Município de Aveiro. Considerou-se uma base de dados semântica que é constituída por códigos, nomes, tipos e outros atributos que descrevem objetos. Esta informação é combinada com dados geográficos sobre os mesmos objetos. Os objetos podem ser edifícios, estradas, ferrovias e hidrovias. A combinação de dados semânticos e espaciais permite definir regras de inferência para caracterizar as zonas usando conceitos como zona comercial, zona residencial, área rural, área urbana, zona acessível ou zona de difícil acesso.

1. Fontes de dados

Os dados usados para a implementação deste caso de estudo foram obtidos a partir da página web do Instituto Nacional de Estatística (INE) [30] e da página web do *openstreetmap* [31].

Os dados do INE incluem dados definitivos do Censos do ano 2011 da população residente, população presente, famílias, alojamentos e edifícios. Os dados estão organizados numa estrutura hierárquica de unidades territoriais: *Nuts* I, II, III, município, freguesia, secção e subsecção estatísticas.

As *NUTS* (Nomenclaturas de Unidades Territoriais - para fins Estatísticos) designam as sub-regiões estatísticas em que se divide o território dos países da União Europeia, incluindo o território português [30].

- *Nuts* 1 - constituído por três unidades, correspondentes ao território do continente e às Regiões Autónomas dos Açores e da Madeira.
- *Nuts 2* - constituído por sete unidades, das quais cinco no continente e os territórios das Regiões Autónomas dos Açores e da Madeira. As sub-regiões em *NUTS 2* são:

Região Norte, Região Centro, Região Lisboa, Região Alentejo, Região Algarve, Região Autónoma dos Açores e Região Autónoma da Madeira;

- *Nuts 3* - constituído por 30 unidades, das quais 28 no continente e duas correspondentes às Regiões Autónomas dos Açores e da Madeira. Por exemplo Sub-região Baixo Vouga, Alto Alentejo, Alentejo Central, Baixo Alentejo, Algarve etc. No caso deste trabalho considerou-se a Sub-região Baixo Vouga.

Um Município é uma autarquia local, criado para atender as necessidades estatísticas locais, classificado como unidade administrativa local 1. Em Portugal existem 308 municípios. No caso deste trabalho foram usados dados do município de Aveiro.

Uma Freguesia é a menor divisão administrativa. Criada para atender as necessidades estatísticas locais, é por isso classificada de unidade administrativa local 2. Em Portugal existem 3091 freguesias.

Uma Secção Estatística é uma unidade territorial correspondente a uma área contínua da Freguesia, com cerca de 300 alojamentos destinados à habitação.

Uma Subsecção Estatística é uma unidade territorial que identifica a mais pequena área homogénea de construção ou não, existente dentro da secção estatística. Corresponde ao quarteirão nas áreas urbanas, ao lugar ou parte do lugar nas áreas rurais ou a áreas residuais que podem ou não conter unidades estatísticas.

Um Lugar é um aglomerado populacional com 10 ou mais alojamentos destinados à habitação de pessoas e com uma designação própria, independentemente de pertencer a uma ou mais freguesias. A aplicação deste conceito nem sempre é fácil devido a duas razões:

- existem aglomerados com uma designação própria mas que têm menos de 10 alojamentos e por isso não deveram ser considerados como lugar.

- há situações em que existe a tendência de considerar como lugar áreas que são conhecidas por um nome específico, quando na realidade são partes (bairros, zonas ou mesmo ruas) de um conjunto mais alargado e uniforme que constitui, de facto, um lugar.

Estes dados estão disponíveis nos formatos CSV e *shapefile*.

O ficheiro CSV agrega dados alfanuméricos sobre estatísticas do Censos 2011 da população residente, população presente, famílias, alojamentos e edifícios organizados numa estrutura hierárquica das unidades territoriais mencionada anteriormente.

O *shapefile* é um formato de armazenamento de dados vetoriais nativo da ESRI (*Environmental Systems Research Institute* – fornecedor internacional de software para sistemas de informação geográfica) [32], que contém a posição, o formato e os atributos dos elementos geográficos. Podem ser utilizados em aplicações *Desktop*, tas como o *ArcSIG for Desktop e o ArcSIG Explorer Desktop.*

O *shapefile* utilizado no caso de estudo é constituído por dados *geoespaciais* das subsecções estatísticas das freguesias. Os ficheiros que constituem o *shapefile* possuem as seguintes extensões: DBF; PRG; SBN; SBX; SHP, SHX, XML. Para o caso de estudo apenas foi utilizado o ficheiro com a extensão *SHP*, que contém dados espaciais do município de Aveiro.

A importação dos dados em formato *shapefile* para a base de dados *SQL-Server* é feita através da aplicação ***Shapefile Uploader for SQL-Server* 2008.** Durante a importação dos dados, a opção para criar índices espaciais não deve ser selecionada, visto que, essa tentativa provoca um erro de importação que culmina com a não criação da tabela na base de dados. Porém esses índices devem ser criados posteriormente usando *o SQL-Server* para melhorar a performance na execução das operações espaciais

A Figura IV-1 representa o modelo de dados do ficheiro em formato *shapefile* que é importado para base de dados *Sql-Server.*

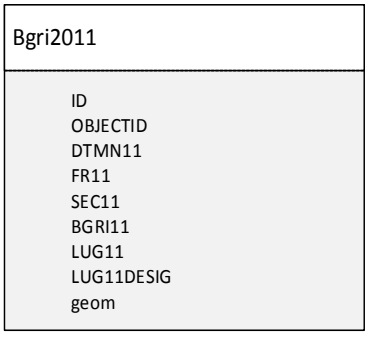


Figura IV-1:Modelo de dados do *shapefile* do INE.

A Tabela IV-1 descreve os atributos do modelo de dados da Figura IV-1

Atributos	**Descrição**
Id	É chave primária da tabela e representa a ordem em que o objeto foi inserido
objectID	Código da subsecção
DTMN11	Código do Município
FR11	Código da freguesia
SEC11	Código da secção
BGRI11	Código completo da secção
LUG11	Código do lugar
LUG11DESIG	Designação do lugar
geom	Dado geométrico que representa o objeto

Tabela IV-1:Descrição do modelo de dados do shapefile do INE.

O *openStreetMap* disponibiliza dados sobre vários tipos de objetos, mas para o caso de estudo foram utilizados os seguintes: edifícios, estradas, ferrovias e hidrovias. Os dados apresentam-se na forma de tabelas de base de dados relacionais. Essas tabelas são constituídas por um campo geográfico que representa o objeto e outros atributos que o descrevem, como mostra o modelo de dados na Figura IV-2.

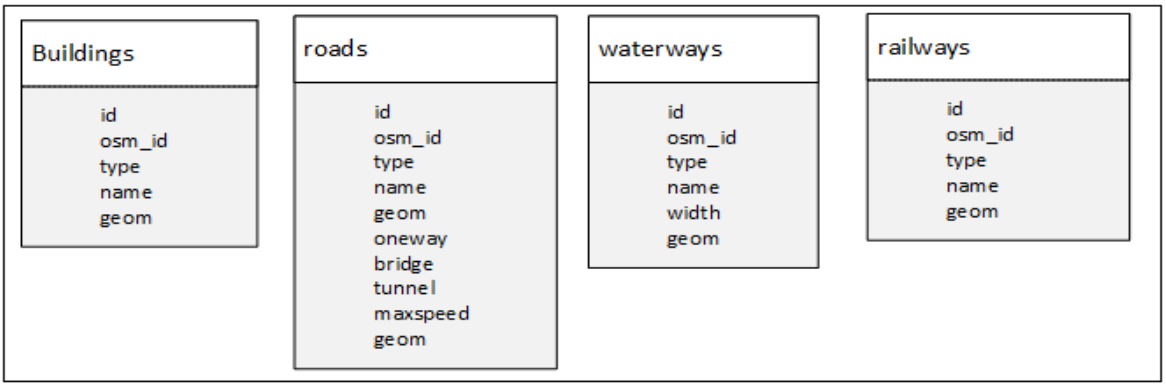


Figura IV-2:Modelo de dados do *shapefile* do *openStreetMap*.

O modelo de dados da Figura IV-2 é constituído por quatro tabelas que possuem atributos que descrevem os objetos conforme a Tabela IV-2:

Atributos	**Descrição**
Id	É chave primária da tabela, indica a ordem de inserção do objeto
osm_id	Identifica o objeto
name	Indica o nome do objeto
geom	Dado geométrico que representa o objeto
width	Comprimento da hidrovia
Oneway	Define o número de estradas
Bridge	O número de pontes que a estrada possui
Tunnel	Número de túneis existentes na estrada
Maxspeed	Determina a velocidade máxima permitida na estrada
Type	Determina o tipo do objeto

Tabela IV-2:Descrição do modelo de dados do *openstreetMap*.

2. Construção do repositório de dados

A partir dos dados importados do *openstreetMap* e do INE, obteveram-se as tabelas edifícios, estradas, ferrovias e hidrovias com os atributos descritos anteriormente na Tabela IV-1 e Tabela IV-2, e a tabela freguesias tem como atributos o código, nome e os dados geométricos das freguesias.

As tabelas do *openstreetMap* contêm dados de todo Portugal. No trabalho apenas se consideraram dados do município de Aveiro. O processo de criação do repositório semântico e espacial seguiu os seguintes passos:

- criar um dado geométrico que representa o município de Aveiro. Isso é feito utilizando a função espacial *ST_Union* do *Sql-server*. Neste caso são unidos todos os dados geométricos que representam freguesias do município de Aveiro e formam um único dado que representa o município;

- selecionar dados dos edifícios, estradas, hidrovias e ferrovias que pertencem ao município de Aveiro. Com o dado geométrico que representa o município de Aveiro, utilizou-se a operação espacial *ST_Contains* para selecionar todos edifícios contidos no município de Aveiro, e a operação *ST_Intersects* para selecionar as estradas, hidrovias e ferrovias que intersectam o município;

- construir um repositório semântico e outro espacial.

2.1. Base de dados semântica

Para a criação da base de dados semântica implementou-se um processo de conversão de dados relacionais em triplos. Neste caso, todos os atributos das tabelas, exceto o *geom,* foram convertidos em: predicados, os códigos (*osm_id*) em sujeito e o valor em objeto. Por exemplo a tabela edifícios (*Buildings*) é constituída pelos seguintes campos: *osm_id, type, name e geom*. Neste caso os campos *type, name e osm_id* serão convertidos em dados semânticos. O campo *osm_id* identifica os objetos em ambas base de dados, o que permite a integração de dados de uma base de dados e da outra. Assim sendo a base de dados espacial será constituída apenas pelos campos *osm_id e geom*. Mais detalhes desse processo são descritos nas secções subsequentes.

Para a criação dos triplos que constituem o repositório semântico foram seguidos os seguintes passos:

- criar prefixos para que a escrita de triplo fosse abreviada e simplificada. A abreviatura substitui uma referência *URI* completa e é formada por um nome qualificado *XML* (também designado por *Qname*). *Qname* é constituído por um prefixo associado a um *namespace URI*, seguido por dois pontos ":" e um nome. *Namespece* são usados para identificar unicamente os elementos. Por exemplo se o *Qname nse* é associado ao *namespace URI http://www.ua.pt/nuts/entidades*, então o *Qname nse:school* é a abreviatura simplificada para a referencia URI *http://www.ua.pt/nuts/entidades/school*. Em seguida é apresentada a lista de prefixos utilizados no caso de estudo.
    - Prefixo *rdf: namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#;*
    - Prefixo *rdfs: namespace URI: http://www.w3.org/2000/01/rdf-schema#;*
    - Prefixo *nse: namespace URI: http://www.ua.pt/nuts/entidades;*
    - Prefixo *nsi: namespace URI: http://www.ua.pt/nuts/individuos;*
    - Prefixo *nsp: namespace URI: http://www.ua.pt/nuts/predicados.*
- criar uma estrutura hierárquica dos objetos manipulados pelo sistema. Como pode ser visto na Figura IV-3, temos três subclasses, nomeadamente atributos*, redfs:type* e operações espaciais e cada uma dessas subclasses pode ser subdivida para armazenar os objetos em classes mais apropriadas. A classe *rdf:type* pode representar hidrovia, edifício, estrada ou ferrovia.

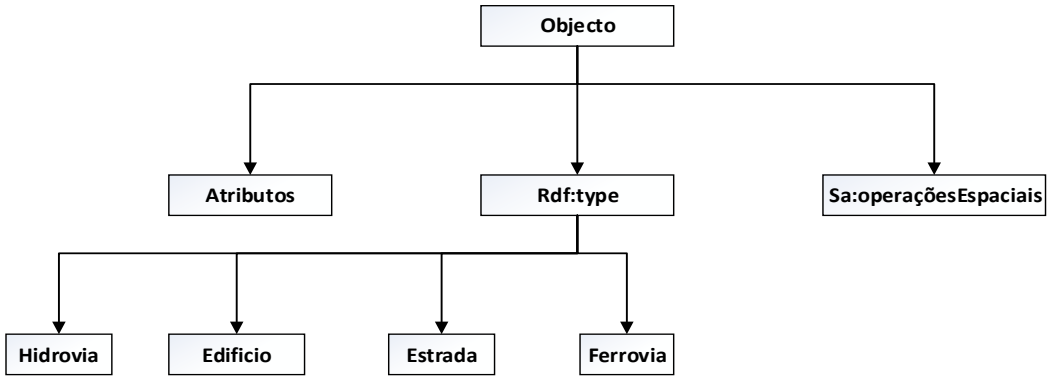


Figura IV-3:Estrutura hierárquica do objeto.

As subclasses do tipo hidrovia, edifício, estrada e ferrovia por sua vez subdividem-se em outras.

A subclasse edifício subdivide-se em subclasses do tipo: casa, escola, hospital, restaurante, hotel, supermercado, estação de comboio, etc., conforme mostra a Figura IV-4.

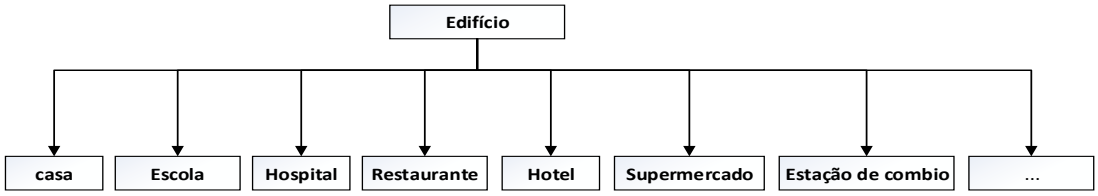


Figura IV-4:Estrutura hierárquica da classe edifício.

Utilizando a estrutura hierárquica da Figura IV-3 e Figura IV-4, seguiu-se o processo de criação de triplos (dados semânticos). Neste caso, o atributo (*name*) do objeto edifício foi convertido em predicado, o código (*osm_id*) em sujeito e o valor convertido em objeto. Neste caso, os edifícios podem ser do tipo casa, escola, hospital, restaurante, hotel, supermercado estação do comboio etc. Estes tipos por sua vez são subtipos do objeto edifício.

Por exemplo, uma escola é classificada como sendo do tipo *school (nsi:136668920 rdf:type nse:school)*, onde *school* é um subtipo do tipo *building (nse:school rdf:subTypeOf nse:building)* e possui um nome *(nsi:136668920 nsp:name "Escola Secundaria Dr. Mario Sacramento").*

A classe hidrovia é composta por subclasses do tipo: rio, canal, barragem, cachoeira, poço, reservatório, dreno e muito mais outras subclasses. Segundo Figura IV-5.

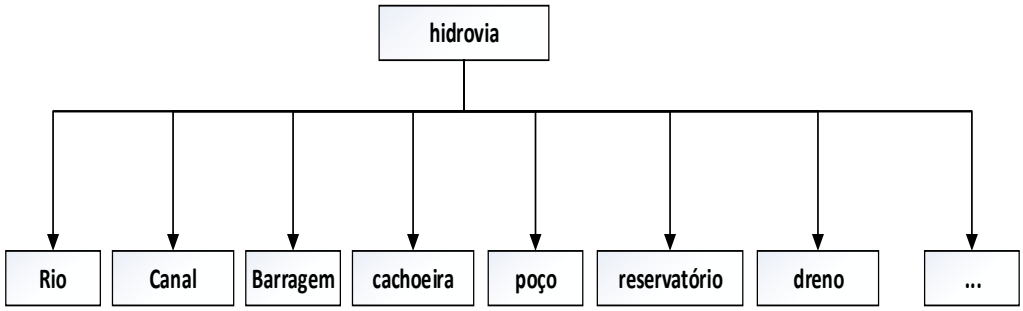


Figura IV-5:Estrutura hierárquica da classe

Seguindo as estruturas hierárquicas da Figura IV-3 e Figura IV-5, são criados triplos para as hidrovias onde o código (osm_id) é um sujeito, o atributo *name* é um predicado e o conteúdo é um objeto. Os objetos do tipo rio, canal, barragem, cachoeira, poço, reservatório, dreno etc. são subtipos do tipo hidrovia.

Por exemplo, uma hidrovia pode ser do tipo canal *(*nsi:11280889 *rdf:type nse:canal)*, onde canal é um subtipo do tipo *waterway (nse:canal rdf:subTypeOf nse: waterway)* e possui um nome *(nsi:11280889 nsp:nome "Ria de Aveiro").*

A Figura IV-6 mostra a estrutura hierárquica da classe estrada.

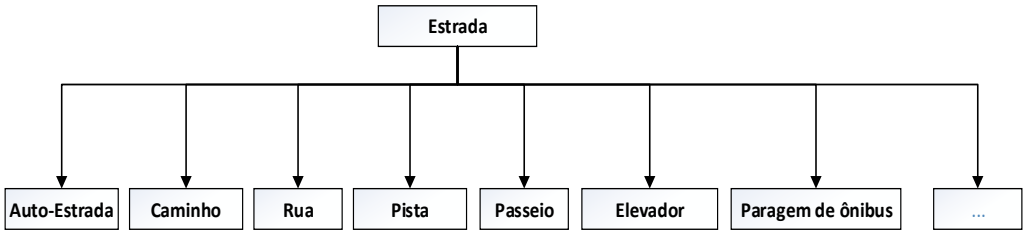


Figura IV-6:Estrutura hierárquica da classe estrada.

As subclasses autoestrada, caminho, rua, pista, passeio, elevador e paragem de autocarro, etc. pertencem à subclasse estrada. Por isso são subtipos do tipo estrada. Os predicados extraídos a partir dos atributos são *name, oneway, bridge, tunnel e maxspeed*. Os códigos (*osm_id*) são convertidos em sujeitos e o valor em objetos.

Por exemplo, uma estrada pode ser do tipo *secondary (*nsi:38628967 *rdf:type nse:secondary)*, onde *secondary* é um subtipo do tipo *road (nse:secondary rdf:subTypeOf nse: road)*, possui um nome *(*nsi: 38628967 *nsp:name "Avenida da Universidade"),* uma via em cada sentido *(*nsi: 38628967 *nsp: oneway "1"),* uma velocidade máxima *(*nsi: 38628967 *nsp:maxspeed "50*"), e não possui ponte *(*nsi: 38628967 *nsp: bridge "0")* nem túnel *(*nsi: 38628967 *nsp: tunnel "0").*

A ferrovia é uma subclasse que é constituída pelas seguintes subclasses: estação, plataforma, trilho, metro, cruzamento, mesa de transferência, prato giratório, etc. A Figura IV-7 mostra a estrutura hierárquica da classe ferrovia.

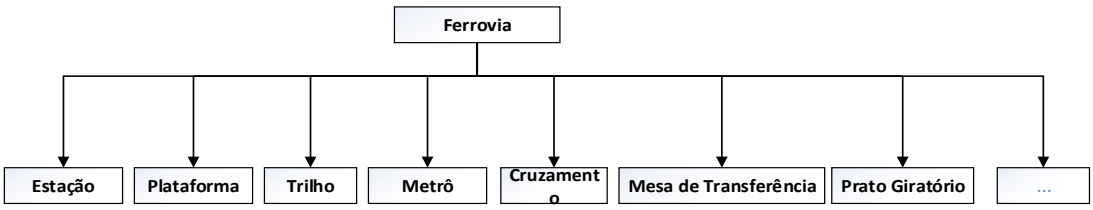


Figura IV-7:Estrutura hierárquica da classe ferrovia

Tendo em conta a estrutura hierárquica da Figura IV-7 definiram-se triplos da seguinte forma. As subclasses desta estrutura constituem os subtipos do tipo ferrovia. O nome é o único predicado identificado. Os códigos (*osm_id*) constituem os sujeitos e o conteúdo dos objetos.

Por exemplo, todos objetos do tipo *rail* (*nsi:152191015 rdf:type nse:rail*) são subtipos do tipo *railway (nse:rail rdf:subTypeOf nse: railway)* e possuem um nome *(nsi:* 152191015 *nsp:name "estação de aveiro").*

Os triplos inseridos na base de dados semânticos foram criados através da execução de consultas *SQL* na base de dados *Sql-Server*. Os resultados são exportados para um ficheiro de texto e por último importados para a base de dados semântica.

Por exemplo a consulta do Código IV-1 permite obter o código e o tipo do edifício e criar um triplo que relaciona esses dois dados. A *string* "*nsi*" e "*nse*" são prefixos que

substituem as referências URI "http://www.ua.pt/nuts/individuos" e "http://www.ua.pt/nuts/entidades" respetivamente. Ambos são sucedidos por dois pontos ":" e pelos nomes dos atributos que permitem obter os códigos (*osm_id*) e os tipos de edifícios (*type)*. A *string* "*rdf:type*" relaciona o código (*osm_id*) do edifício e o seu tipo (*type*). Na Figura IV-8 temos como resultado uma linha escrita da seguinte forma: "*nsi:136668920 rdf:type nse:school"* onde "*136668920"* é o código do edifício e *school* representa o tipo de edificio.

```
select 'nsi:'+osm_id+' rdf:type '+'nse:'+ type from building
```

Código IV-1 consulta sql para a classificação do edifícios

A Figura IV-8 mostra uma coleção de triplos gerados pela consulta *sql* para a classificação dos edifícios.

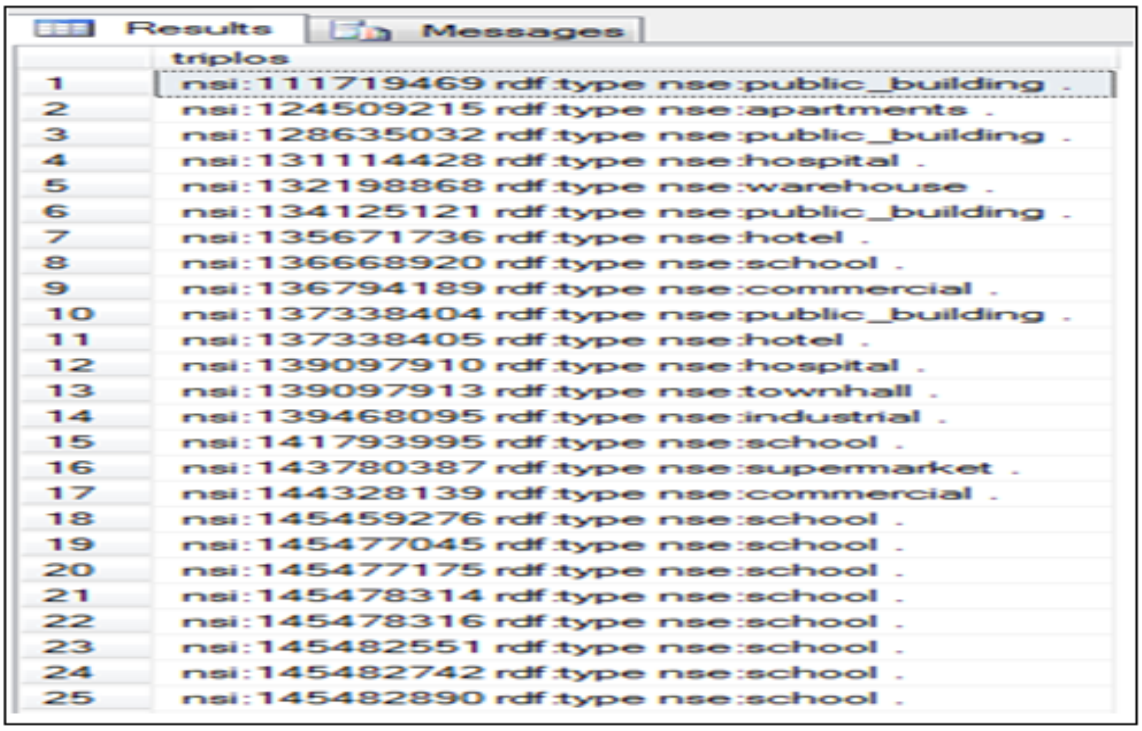


Figura IV-8:Consulta gerada para a classificação dos edifícios

Depois de se anexar o resultado da consulta aos prefixos, o ficheiro do texto é guardado e importado para a base de dados *Virtuoso*. A definição dos prefixos deve ser feita segundo a seguinte sintaxe, escrita da palavra reservada "*prefix*" precedida pelo simbolo *"@"*, escrita

da abreviatura (por exemplo "*nsi*") sucedido de dois pontos ":" e por fim a escrita da referência URI a abreviar (por exemplo http://www.ua.pt/nuts/individuos).

A Figura IV-9 mostra o ficheiro do texto preparado para ser importado para a base de dados semântica. Em que as primeiras três linhas correspondem a criação dos prefixos e outras a criação de triplos.

```
@prefix nse:  <http://www.ua.pt/nuts/entidades/>
@prefix nsi:  <http://www.ua.pt/nuts/individuos/>
@prefix nsp:  <http://www.ua.pt/nuts/predicados/>

nsi:111719469 rdf:type nse:public_building .
nsi:124509215 rdf:type nse:apartments .
nsi:128635032 rdf:type nse:public_building .
nsi:131114428 rdf:type nse:hospital .
nsi:132198868 rdf:type nse:warehouse .
nsi:134125121 rdf:type nse:public_building .
nsi:135671736 rdf:type nse:hotel .
nsi:136668920 rdf:type nse:school .
nsi:136794189 rdf:type nse:commercial .
nsi:137338404 rdf:type nse:public_building .
nsi:137338405 rdf:type nse:hotel .
nsi:139097910 rdf:type nse:hospital .
nsi:139097913 rdf:type nse:townhall .
nsi:139468095 rdf:type nse:industrial .
nsi:141793995 rdf:type nse:school .
nsi:143780387 rdf:type nse:supermarket .
nsi:144328139 rdf:type nse:commercial .
nsi:145459276 rdf:type nse:school .
nsi:145477045 rdf:type nse:school .
nsi:145477175 rdf:type nse:school .
nsi:145478314 rdf:type nse:school .
nsi:145478316 rdf:type nse:school .
nsi:145482551 rdf:type nse:school .
nsi:145482742 rdf:type nse:school .
nsi:145482890 rdf:type nse:school .
nsi:145482891 rdf:type nse:school .
```

Figura IV-9:Ficheiro de texto gerado a partir do resultado da consulta

A Figura IV-10 mostra a interface gráfica para a inserção dos triplos na base de dados *Virtuoso* a partir dos dados contidos no ficheiro do texto. Foi selecionado o ficheiro do texto contendo triplos sobre edifícios e escolhido o grafo http://www.ua.pt/nuts para alojar os triplos.

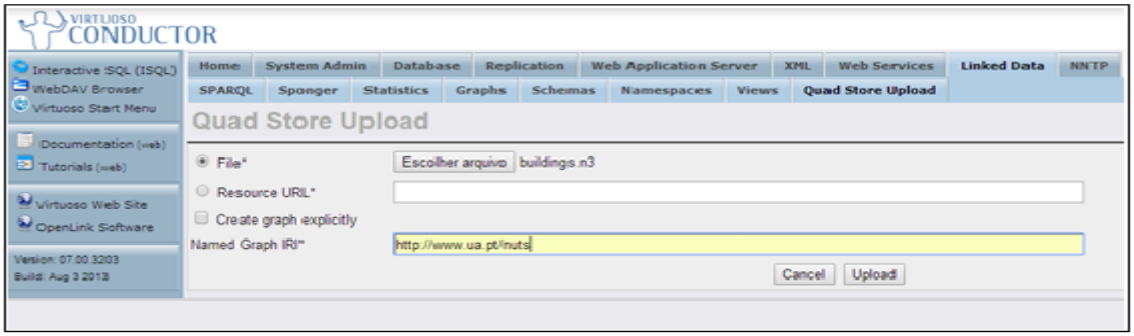


Figura IV-10: Carregamento do ficheiro do texto contendo triplos

De seguida carrega-se (*upload*) o ficheiro. Feito isso, as consultas *SPARQL* sobre os dados contidos no ficheiro podem ser executadas na base de dados *Virtuoso*.

2.2. Base de dados espacial

A base de dados espacial armazena apenas os códigos (*osm_id*) e a geometria do objeto. A Figura IV-11 mostra o modelo de dados da base de dados espacial final.

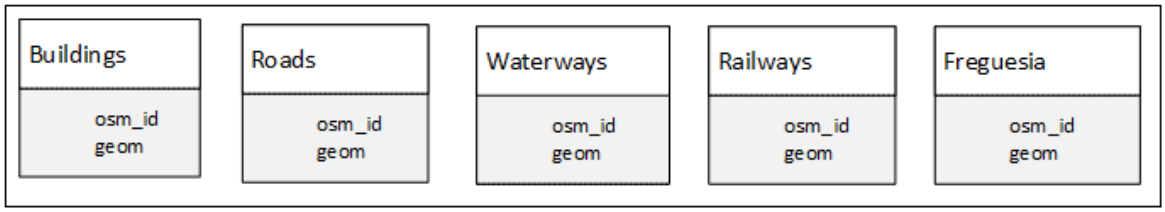


Figura IV-11:Modelo de dados da base de dados espacial final

O modelo apresentado na Figura IV-11 permite executar apenas consultas que envolvam operações espaciais sobre os objetos.

2.3. Representação objeto espacial com semântica

O sistema proposto neste trabalho permite executar vários tipos de consultas:

- consultas sem nenhuma operação espacial. Neste caso, a pesquisa não exige informação espacial para obter a sua resposta. A resposta é devolvida diretamente da base de dados semântica sem ser necessário criar um repositório temporário, por exemplo: obter o número de residentes em cada freguesia;

- consultas envolvendo operações espaciais. O primeiro passo consiste em verificar se as relações espaciais já existem no repositório temporário. Se existirem, a pesquisa é feita sobre os triplos já existentes, senão, é executada uma consulta espacial e as novas relações geradas são inseridas no repositório semântico temporário na forma de triplos. Por fim é feita a pesquisa sobre os novos dados.

  Por exemplo, para responder à pergunta "Quais são os edifícios que pertencem a cada uma das freguesias?" serão gerados triplos que relacionam freguesias com os

edifícios, isto é, será criada uma coleção de triplos que determina quais os edifícios em cada freguesia. No triplo *(nsi:010506 nsp:Contem nsi:*136668920*)*, o código 010506 identifica uma freguesia cujo nome é Glória *(nsi:010565 nsp:name "Gloria";),* e o código 136668920 identifica um edifício do tipo escola cujo nome é Escola Secundaria Dr. Mário Sacramento. Esta informação já existe na base de dados semântica, conforme os triplos seguintes: *(nsi:136668920 rdf:type nse:school; nse:school subTypeOf nse:building; nsi:136668920 nsp:name "Escola Secundaria Dr. Mario Sacramento")*. Feito isto, já pode ser executada uma consulta *SPARQL* para devolver a resposta esperada.

- se as relações espaciais já existirem no repositório temporário podem ser simplesmente atualizadas;
- a opção executada por omissão consiste em processar todas relações no sentido de verificar se as relações já existem ou não, ou se devem ser atualizadas.

2.4. Regras de inferência e perguntas e respostas

Nesta secção são definidas regras de inferência para caracterizar as zonas usando conceitos como zona comercial, zona residencial, área rural, área urbana, zona acessível ou zona de difícil acesso. São ainda exemplificadas perguntas que podem ser feitas ao sistema e visualizadas as respetivas respostas.

Regras de Inferência

O caso de estudo consiste em caracterizar as zonas geográficas de acordo com o tipo de povoamento, tipo de atividade praticada e a acessibilidade da zona. Estes conceitos serão definidos e usados para criar regras de inferência para extrair informação sobre as freguesias

Tipo de povoamento

A definição do tipo de povoamento é feita em função da densidade demográfica de cada freguesia ou lugar:

- Freguesias urbanas - freguesias que possuem densidade populacional superior a 500 h/km², onde o "h" significa habitantes, ou com população residente superior ou igual a 5 000 habitantes.

- Freguesias semiurbanas - freguesias não urbanas que possuem densidade populacional superior a 100 h/km² e inferior ou igual a 500 h/km², ou com população residente superior ou igual a 2 000 habitantes e inferior a 5 000 habitantes

- Freguesias rurais - as restantes.

Tipo atividade

- Zonas comerciais: zonas que tenham, em média, 20 ou mais edifícios/km² destinados a atividades do comércio ou serviços;

- Zonas residenciais: caso não seja zona comercial;

Tipo de acesso

- Zonas acessíveis: zonas que possuem em média mais de 10 estradas ou 5 paragens de autocarro e uma estação de comboios;

- Zonas de difícil acesso: caso não seja zona acessível.

Perguntas e respostas

A secção a seguir mostra as perguntas que podem ser feitas ao sistema e como são retornadas e visualizadas as respostas.

1. Listar todas freguesias que sejam:

    a) Rurais;

    b) Urbanas;

    c) Comerciais;

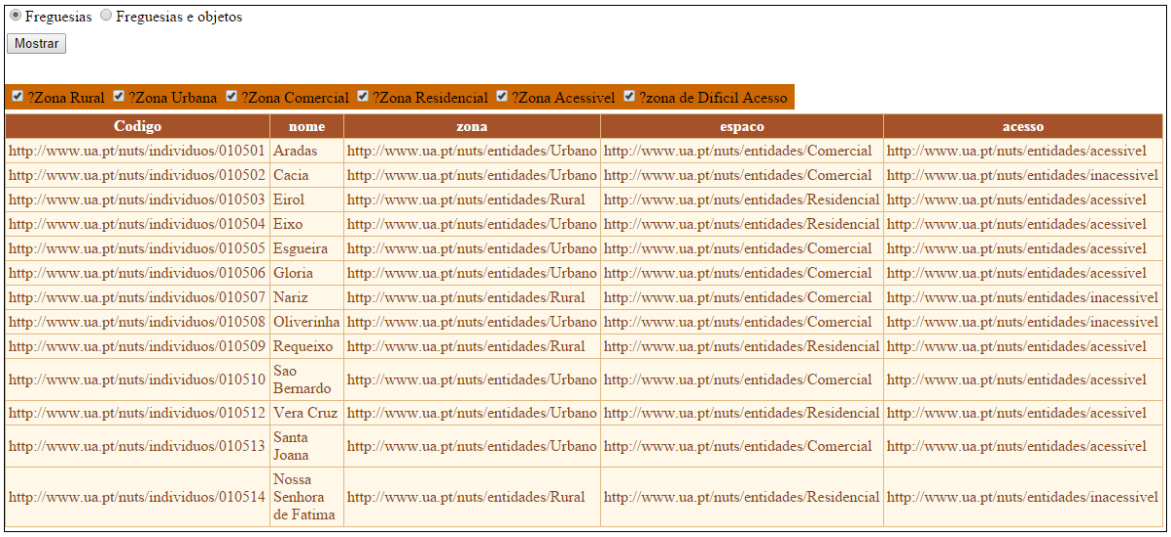
    d) Residenciais;

    e) Acessíveis;

f) De difícil acesso.

A aplicação apresenta duas opções (*freguesias* ou *freguesias e objetos*). A opção *freguesia* permite listar freguesias nas condições das classificações escolhidas, as freguesias podem ser classificadas em em zona rural, urbana, comercial, residencial, acessível ou de difícil acesso.

A Figura IV-12 mostra resultado de uma pesquisa em que são listadas todas as freguesias e as suas respetivas classificações. O resultado é constituído pelo código e nome da freguesia, tipo de povoamento (rural ou urbano), tipo de atividade (comercial ou residencial) e tipo de acesso (acessível ou de difícil acesso).



Freguesias Freguesias e objetos

Mostrar

?Zona Rural ?Zona Urbana ?Zona Comercial ?Zona Residencial ?Zona Acessivel ?zona de Dificil Acesso

Codigo	nome	zona	espaco	acesso
http://www.ua.pt/nuts/individuos/010501	Aradas	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010502	Cacia	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/inacessivel
http://www.ua.pt/nuts/individuos/010503	Eirol	http://www.ua.pt/nuts/entidades/Rural	http://www.ua.pt/nuts/entidades/Residencial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010504	Eixo	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Residencial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010505	Esgueira	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010507	Nariz	http://www.ua.pt/nuts/entidades/Rural	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/inacessivel
http://www.ua.pt/nuts/individuos/010508	Oliverinha	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/inacessivel
http://www.ua.pt/nuts/individuos/010509	Requeixo	http://www.ua.pt/nuts/entidades/Rural	http://www.ua.pt/nuts/entidades/Residencial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010510	Sao Bernardo	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010512	Vera Cruz	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Residencial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010513	Santa Joana	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010514	Nossa Senhora de Fatima	http://www.ua.pt/nuts/entidades/Rural	http://www.ua.pt/nuts/entidades/Residencial	http://www.ua.pt/nuts/entidades/inacessivel

Figura IV-12:Resultado da pergunta 1.

O utilizador pode alterar as opções da pergunta conforme as suas necessidades, por exemplo, listar as freguesias obedecendo às condições *b*, *c* e *e* da pergunta, isto é, freguesias que sejam urbanas, comerciais e acessíveis. A Figura IV-13 mostra o resultado da pesquisa selecionada.

Freguesias Freguesias e objetos

Mostrar

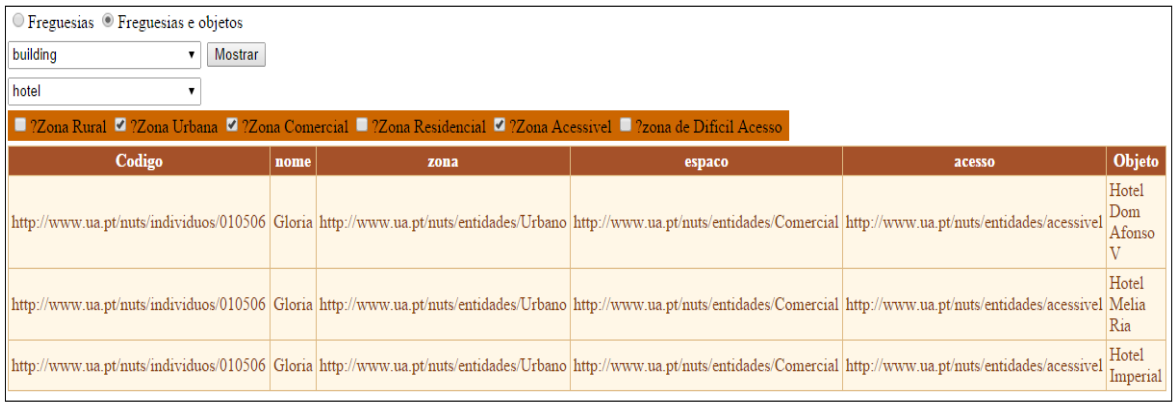
?Zona Rural ?Zona Urbana ?Zona Comercial ?Zona Residencial ?Zona Acessivel ?zona de Dificil Acesso

Codigo	nome	zona	espaco	acesso
http://www.ua.pt/nuts/individuos/010501	Aradas	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010505	Esgueira	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010510	Sao Bernardo	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel
http://www.ua.pt/nuts/individuos/010513	Santa Joana	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel

Figura IV-13:Resposta da pergunta obedecendo as condições a, c, e e.

1.1. Listar hotéis localizados em zonas urbanas, comerciais e acessíveis.

Como os hotéis são edifícios é necessário selecionar a opção *building* (edifícios) e em seguida listar os *Hotel* (hotéis). No grupo de opções seguintes pode-se escolher as características das zonas que se pretendem. Neste caso concreto estão selecionadas as zonas urbanas, comerciais e acessíveis. A Figura IV-14 mostra o resultado da pesquisa segundo as opções selecionadas.



Freguesias Freguesias e objetos

building Mostrar

hotel

?Zona Rural ?Zona Urbana ?Zona Comercial ?Zona Residencial ?Zona Acessivel ?zona de Dificil Acesso

Codigo	nome	zona	espaco	acesso	Objeto
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel	Hotel Dom Afonso V
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel	Hotel Melia Ria
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/Urbano	http://www.ua.pt/nuts/entidades/Comercial	http://www.ua.pt/nuts/entidades/acessivel	Hotel Imperial

Figura IV-14:Resultado da pergunta 1.1

2. Listar os seguintes dados estatísticos para cada Freguesia pertencente ao Município de Aveiro.

   a) A área total de cada freguesia, dada em km^2;

   b) A quantidade de edifícios existentes;

c) A área total ocupada por edifícios dada em km^2;

d) A percentagem da área ocupada por edifícios (área da freguesia/área dos edificios);

e) Número de residentes por freguesia;

f) A densidade populacional (número de pessoas/km^2).

A Figura IV-15 mostra a resposta gerada quando são selecionadas todas perguntas (opções).

? AreaFreguesia ? NumeroEdificios ? AreaEdificios ? Percentagem ? NumResidentes ? DensidadePopulacional Estatisticas

idFreguesia	nome	AreaFreguesia	numeroEdificios	areaEdificios	percentagem	NumResidentes	DensidadePopulacional
http://www.ua.pt/nuts/individuos/010507	Nariz	9,90867	3	0,002360177	0,023819311774435923	1356	137
http://www.ua.pt/nuts/individuos/010501	Aradas	9,49926	60	0,094823214	0,998216850575729056	9521	1002
http://www.ua.pt/nuts/individuos/010508	Oliverinha	12,8416	13	0,035462015	0,276149506292050834	4633	361
http://www.ua.pt/nuts/individuos/010506	Gloria	7,28926	157	0,2757591823	3,783088849896971709	12561	1723
http://www.ua.pt/nuts/individuos/010514	Nossa Senhora de Fatima	13,4398	4	0,01496886	0,11137710382594979	1844	137
http://www.ua.pt/nuts/individuos/010505	Esgueira	18,2594	377	0,3528257912	1,932296741404427308	12772	699
http://www.ua.pt/nuts/individuos/010502	Cacia	38,0909	163	0,3030518305	0,795601654200872124	7059	185
http://www.ua.pt/nuts/individuos/010511	Sao Jacinto	14,737	109	0,1268207141	0,86055991110809527	896	61
http://www.ua.pt/nuts/individuos/010504	Eixo	17,782	3	0,002205262	0,012401653357327634	5284	297
http://www.ua.pt/nuts/individuos/010512	Vera Cruz	40,9751	410	0,1392838443	0,339923134537804666	9271	226
http://www.ua.pt/nuts/individuos/010503	Eirol	6,0736	4	0,0018863717	0,031058543532665964	724	119
http://www.ua.pt/nuts/individuos/010510	Sao Bernardo	4,18799	115	0,1043920909	2,492653776632704471	4715	1126
http://www.ua.pt/nuts/individuos/010509	Requeixo	11,0212	8	0,003657911	0,033189770623888505	1150	104
http://www.ua.pt/nuts/individuos/010513	Santa Joana	6,22174	94	0,0913167476	1,467704333514418796	7756	1247

Figura IV-15:Resposta da pergunta 2

O método implementado para responder a esta pergunta compreende os seguintes passos.

Primeiro verifica-se na base de dados semântica a existência de triplos que relacionam os códigos dos edifícios e freguesias com as suas áreas. Caso não existam, na base de dados espacial são seguidos os seguintes passos:

- calcular a área de cada freguesia e relacionar os códigos das freguesias com as respetivas áreas (utilizando a função *ST_Area()*);
- calcular as áreas dos edifícios na base de dados espacial, utilizando a função espacial *ST_Area()*;

- criar triplos que relacionam os códigos dos edifícios com as suas áreas (código possui área). Por exemplo, o triplo *(nsi: 135671736 nsp:Areaa "70")* significa que o edifício com código *135671736* possui uma área de $70m^2$.

- devolver triplos que relacionam as freguesias com os edifícios. *(nsi:*010512 *nsp:contains nsi: 135671736)*

- enviar a coleção de triplos para base de dados semântica

Na base de dados semântica

- inserir os triplos criados na base de dados espacial no repositório temporário;

- executar uma consulta *SPARQL* na base de dados semântica que devolve uma coleção de triplos de freguesias e edifícios, constituídos por código e o tipo de objeto (código tipo objeto), código do objeto e o seu nome (código nome do objeto) e inserir a coleção no grafo temporário. De salientar que o *Virtuoso* não duplica os triplos, isto é, tem um mecanismo de otimização que permite verificar a existência do triplo antes de inseri-lo.

  Exemplos de triplos.

  *(nsi:*010512 *rdf:type nse:freguesia)*

  *(nsi:*010512 *nsi:name "Vera Cruz")*

  (nsi:135671736 rdf:type nse:hotel)

  (nse:hotel nsi:subTypeOf nse:building)

- executar uma consulta *SPARQL* sobre o grafo temporário de acordo com as perguntas escolhidas, como mostra a Figura IV-15.

Descrição das opções apresentadas pela janela da Figura IV-15.

- *AreaFreguesia* – área ocupada por uma determinada freguesia;

- *NúmeroEdifícios* – número total de edifícios existentes em cada freguesia;
- *AreaEdifícios* – área total ocupada por edifício dentro de uma freguesia;
- Percentagem – é a percentagem da área ocupada por edifícios dentro da área total de uma freguesia (Área de Edifícios / Área Freguesia);
- *NumResidentes* – é o número total de pessoas que residem numa determinada freguesia;
- *DensidadePopulacional* – número de h/km^2 (Número de residentes/ Área Freguesia). Este valor permite determinar se a freguesia é rural ou não.

Nos resultados são também exibidos os códigos e nomes das freguesias.

2.1. Listar para uma determinada freguesia o número de edifícios e a área que ocupam em cada tipo de edifício (determinado de acordo com a sua funcionalidade).

A aplicação disponibiliza uma lista de freguesias, onde o utilizador pode selecionar a freguesia que pretende para visualizar os seus dados.

Freguesia

Gloria

idFreguesia	nome	type	numEdificios	areaEdificios
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/museum	1	0,0044747
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/house	2	0,00526803
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/roof	2	0,000736055
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/pois	45	0,1492536339
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/retail	1	0,0071306
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/theatre	1	0,00123498
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/parking	1	0,00552958
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/school	34	0,1115491769
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/car_wash	1	0,000302029
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/place_of_worship	1	0,000494953
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/building	157	0,2757591823
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/chapel	1	0,00025717
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/townhall	1	0,000560252
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/restaurant	2	0,000905927
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/hotel	3	0,003746215
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/bank	1	0,000361237
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/retirement_home	1	0,00182411
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/courthouse	1	0,00166586
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/hospital	1	0,0154912
http://www.ua.pt/nuts/individuos/010506	Gloria	http://www.ua.pt/nuts/entidades/police	1	0,00066266

Figura IV-16:Resultado da pergunta 2.1

A tabela de resposta é constituída pelo código e nome da freguesia, uma referência URI que representa o tipo de edifício, o número de edifícios e a área ocupada por esses edifícios.

Para responder esta pergunta nas bases de dados espacial e semântica são seguidos os passos descritos na resposta da pergunta 2. A diferença é no retorno da resposta, em que no repositório temporário é executada uma consulta que calcula o número e a área ocupada por cada tipo de edifício.

3. Listar edifícios, hidrovias, estradas, pertencentes a uma determinada freguesia em função dos seus tipos. Por exemplo, pode se pretender uma lista de todos edifícios que sejam escolas, hotéis, hospitais etc., ou todas hidrovias que sejam rios, canais etc.

   Na Figura IV-17 a opção "*Freguesia*" permite selecionar a freguesia, neste caso selecionou-se *Oliverinha* e a opção "*Tipo Objeto*" permite selecionar o tipo do objeto, no caso foi selecionado o objeto do tipo *building.* A opção "*Todos*" permite listar todos os objetos que pertençam ao tipo selecionado e a opção "*selecionados"* gera uma *checkedList* onde se pode escolher os subtipos do tipo do objeto selecionado, Foram escolhidos os seguintes subtipos do tipo *building: church* (Igreja)*, hospital* (hospital) *e school* (escola). Pressionando o botão "*Mostrar*" é gerada uma lista com informação da freguesia e dos objetos do tipo escolhido, como se pode ver na Figura IV-18.

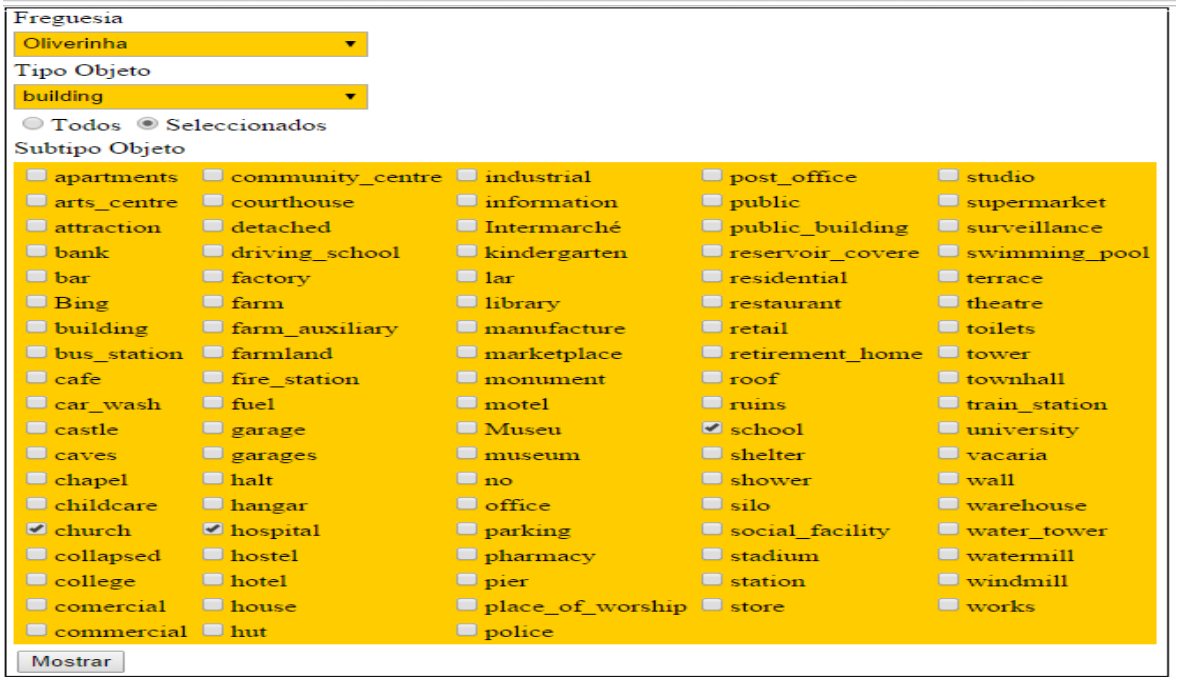


Figura IV-17:Opções da pergunta 3

A Figura IV-18 mostra todas igrejas, hospitais e escolas localizadas na freguesia Oliverinha.

Para responder esta pergunta nas bases de dados espacial e semântica são seguidos os passos descritos na resposta da pergunta 2. A diferença é que neste caso na base de dados espacial apenas são criados triplos que relacionam as freguesias com os edifícios *(nsi:*010512 *nsp:contains nsi:135671736)* e no retorno da resposta, no repositório temporário é executada uma consulta que devolve os tipos dos objetos selecionados (neste caso "*church*", "*hospital*", "*school*") que estejam contidos na freguesia selecionada.

idfreguesia	freguesia	idElement	nome
http://www.ua.pt/nuts/individuos/010508	Oliverinha	http://www.ua.pt/nuts/individuos/159259875	Igreja de Oliveirinha
http://www.ua.pt/nuts/individuos/010508	Oliverinha	http://www.ua.pt/nuts/individuos/238257116	Centro de Saude de Oliveirinha
http://www.ua.pt/nuts/individuos/010508	Oliverinha	http://www.ua.pt/nuts/individuos/238256992	Escola EB1 de Costa do Valado
http://www.ua.pt/nuts/individuos/010508	Oliverinha	http://www.ua.pt/nuts/individuos/238256993	Escola EB2/3 Castro Matoso
http://www.ua.pt/nuts/individuos/010508	Oliverinha	http://www.ua.pt/nuts/individuos/238363709	Escola EB1 de Quintas
http://www.ua.pt/nuts/individuos/010508	Oliverinha	http://www.ua.pt/nuts/individuos/244233232	Escola EB1 de Oliveirinha

Figura IV-18:Resposta da pergunta 3.

4. Listar edifícios, ou hidrovias ou estradas que tenham nos seus arredores outros edifícios, ou hidrovias ou estradas.

Na Figura IV-19 a opção "*Tipo Objeto 1*" permite escolher o tipo do objeto neste caso escolheu-se a opção *building* (edifícios) para permitir que a opção *"SubTipo Objeto 1"* seja preenchida por subtipos do tipo selecionado (neste caso, o tipo *building*) onde se deve selecionar o subtipo (no caso, selecionou-se o subtipo *hotel*). A opção "*Valor Buffer*"permite especificar o raio de extenção que será abrangido tomando o objeto do subtipo selecionado (*hotel*) como o ponto central. Na opção "*Tipo Objeto 2*" é selecionada o tipo de objeto (*waterway*) que se pretende verificar a sua abrangência no raio definido. A *checkeList* é preenchida por subtipos *(Canal, dam, drain, river, stream*) do tipo de objetos selecionado na opção "*Tipo Objeto 2"* (*waterway*).

O resultado é constituído por código (referência URI) e o nome do hetel, código (referência URI) e o nome do canal.

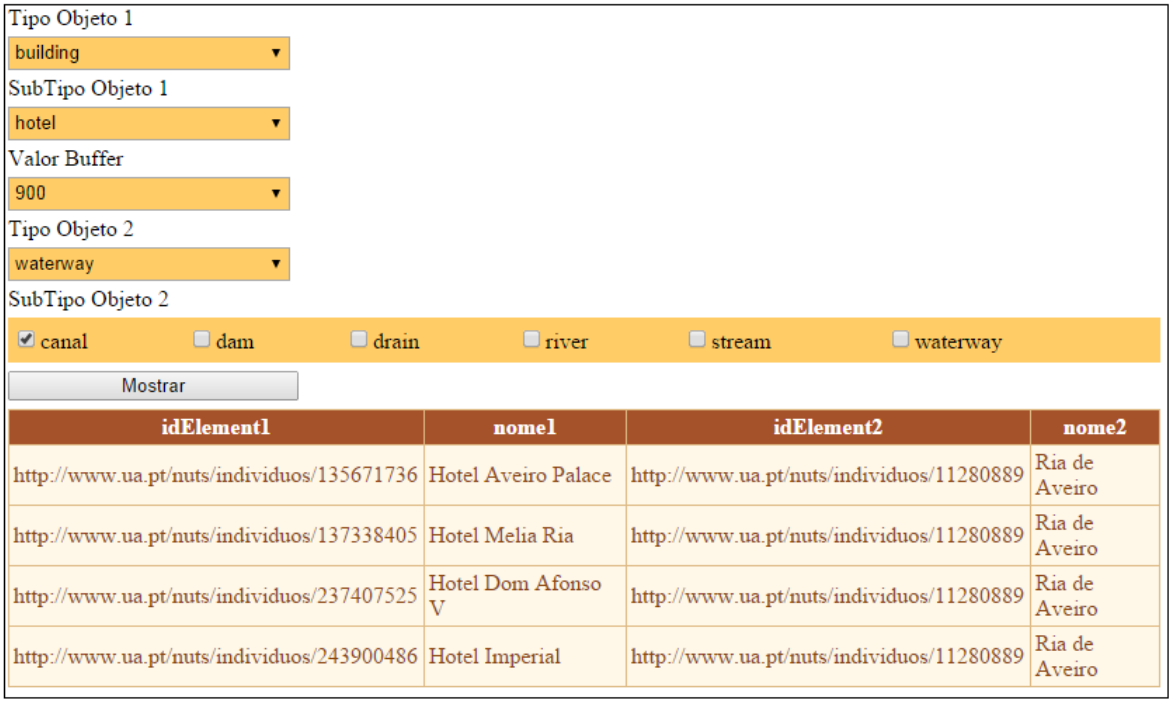


Figura IV-19:Resposta da pergunta 4

O método implementado para responder a estas perguntas compreende os seguintes passos.

Na base de dados espacial.

- gerar um *buffer,* com o valor especificado na opção "*Valor Buffer*", sobre todos os objetos do tipo selecionado na opção "*Tipo Objeto 1*" (no exemplo, o tipo *building*).
- selecionar todos os objetos escolhidos na opção "*Tipo Objeto 2*" (no exemplo, *waterway*) que sejam abrangidos pelo *buffer* de cada objeto pertencente ao tipo escolhido na opção "*Tipo Objeto 1*" (*building*) e criar um triplo que relaciona ambos tipos de objetos. Por exemplo (*nsi:54812658 nsp:buildbufferwaterway900 nsi:4856704).*
- enviar a coleção de triplos para base de dados semântica.

Na base de dados semântica

- inserir os triplos criados na base de dados espacial no repositório temporário;
- executar na base de dados semântica uma consulta que para ambos tipos (*building* e *waterway*) devolva uma coleção de triplos constituídos por

código e o tipo de objeto (código tipo objeto), código do objeto e o seu nome (código nome do objeto).

Exemplos de triplos.

(nsi:135671736 rdf:type nse:hotel)

(nsi:135671736 nsp:nome "Hotel Avenida")

(nse:hotel nsi:subTypeOf nse:building)

(nsi:135671736 rdf:type nse:canal)

(nsi:11280889 nsp:nome "Ria de Aveiro")

(nse:canal nsi:subTypeOf nse:waterway)

- inserir a coleção de triplos no grafo temporário;
- executar sobre o repositório temporáio uma consulta que devolva o código (referência URI) e o nome do objeto do 'subtipo' escolhido na opção (*subtipo Objeto 1*) e o código (referência URI) e o nome do objeto do tipo selecionado na opção "*subtipo Objeto 2*" que tenha sido abrangido pelo *buffer* do 1º objeto.

V Conclusão e trabalho futuro

1. Conclusão

O principal objetivo deste projeto de dissertação foi o desenvolvimento de um sistema que permitisse a integração de dados espaciais com dados semânticos. Permitindo desta forma que aplicações clientes possam enviar suas pesquisas espaciais e semânticas ou simplesmente semânticas e obter respostas. Este objetivo foi materializado, visto que, o sistema integra dados espaciais e dados semânticos armazenados em repositórios diferentes e permite pesquisas. Por exemplo responde a perguntas relacionadas com a localização dos edifícios, estradas, hidrovias e ferrovias. Para a caracterização das zonas são obtidas informações espaciais através da utilização de funções e operações espaciais, esta informação é inserida num repositório semântico onde são aplicadas as regras de inferência que classificam as zonas. A resposta é enviada ao cliente partindo da base de dados semântica.

Para além dos objetivos satisfeitos. Durante o desenvolvimento deste trabalho foram adquiridos vários conhecimentos: foi estudada a manipulação de dados espaciais, em que apesar de se ter o domínio da linguagem SQL, a experiência de manipulação de dados espaciais era inexistente e houve o primeiro contato com a base de dados *Virtuoso*.

Os conhecimentos já existentes como trabalhar com a linguagem de programação C# e sistema de gastão de base de dados SQL-Server foram melhorados. Todos os conhecimentos envolvendo os processos para a integração de dados espaciais em web semântica foram adquiridos ao longo da pesquisa e desenvolvimento do sistema.

Por fim, também permitiu ganhar experiência na organização das várias fases da implementação de um projeto desta dimensão.

2. Análise critica

No presente trabalho não foram utilizados muitas das classes e propriedades dos objetos espaciais propostos pelo padão GeosSPARQL. Essas propriedades permitem descrever dados espaciais e associar os recursos com as suas geometrias por forma a permitir o armazenamento e recuperação eficiente de dados. Considerando a propriedade *geo:hasGeometry que* é utilizada para ligar recursos com uma geometria que representa a sua extensão espacial. Um recurso pode ter várias geometrias associadas. Por exemplo o recurso universidade de Aveiro pode estar associada através desta propriedade com as geometrias que representam os departamentos de Mecânica, departamento de eletrónica, telecomunicações e Informática etc., pelo que fica simples e compreensível escrever uma consulta SPARQL espacial para devolver todas geometrias do recurso universidade de Aveiro.

Os dados utilizados para o caso do estudo apresentam limitações. Estes em alguns casos não possuem a descrição dos objetos. Por exemplo sobre um objeto pode se saber que a geometria representa um edifício mas sem se saber o tipo de edifício, isto é, se é uma escola, hotel ou hospital. Este fenómeno faz com que as respostas não sejam consistentes.

3. Trabalho futuro

Adaptação do sistema para uma realidade do país de proveniência do autor (Moçambique). Em que, por exemplo tendo em conta a localização das residência dos estudantes e das suas escolas, o sistema permitisse produzir estatísticas sobre a distância que os estudantes percorrem para chegarem as suas escolas ou produzir estatísticas sobre a distância que as pessoas percorrem para chegarem às suas unidades sanitárias. Tendo um sistema que responde desta forma pode ajudar às estruturas governamentais competentes a alocar os recursos do estado em lugares estratégicos.

Criação de mais classes e propriedades propostas pelo GeoSPARQL para permitir o armazenamento, recuperação e partilha eficiente de dados espaciais.

Desenvolvimento de um caso estudo em que a partir dos resultados que indicam os objetos seja possível chegar a localização do mesmo no *Google Maps* ou equivalente.

Referências

[1] C. Dempsey, "What is SIG?", *SIG Lounge*, 10-set-2010. .

[2] A. Karmacharya, "Introduction of a spatial layer in the Semantic Web framework: a proposition through the Web platform ArchaeoKM", Université de Bourgogne, 2011.

[3] M. J. de Smith, M. F. Goodchild, e P. A. Longley, "Geospatial analysis", *Matador*, 2009.

[4] G. Câmara, A. M. Monteiro, S. D. Fucks, e M. S. Carvalho, "Spatial analysis and SIG: a primer", *Natl. Inst. Space Res. Braz.*, 2009.

[5] R. Battle e D. Kolas, "Enabling the geospatial semantic web with Parliament and GeoSPARQL", *Semantic Web*, vol. 3, nº 4, p. 355–370, 2012.

[6] "WGS84 - World Geographic System". [Online]. Available at: http://www.isa.utl.pt/dm/sig/sig20002001/TemaSGR/wgs84.html. [Acessado: 09-set-2014].

[7] "epsg > EPSG home". [Online]. Available at: http://www.epsg.org/. [Acessado: 09-set-2014].

[8] "Standards - W3C". [Online]. Available at: http://www.w3.org/standards/. [Acessado: 09-set-2014].

[9] B. DuCharme, *Learning SPARQL*. O'Reilly Media, Inc., 2013.

[10] F. Manola, E. Miller, B. McBride, e others, "RDF primer", *W3C Recomm.*, vol. 10, nº 1–107, p. 6, 2004.

[11] D. Beckett e B. McBride, "RDF/XML syntax specification (revised)", *W3C Recomm.*, vol. 10, 2004.

[12] T. Berners-Lee, J. Hendler, O. Lassila, e others, "The semantic web", *Sci. Am.*, vol. 284, nº 5, p. 28–37, 2001.

[13] A. Karmacharya, C. Cruz, F. Boochs, e F. Marzani, “Integration of spatial processing and knowledge processing through the semantic web stack”, in *GeoSpatial Semantics*, Springer, 2011, p. 200–216.

[14] “OGC Standards | OGC”. [Online]. Available at: http://www.opengeospatial.org/standards. [Acessado: 05-set-2014].

[15] “GeoSPARQL - A Geographic Query Language for RDF Data | OGC”. [Online]. Available at: http://www.opengeospatial.org/standards/geosparql. [Acessado: 09-set-2014].

[16] “The OGC requests comment on the candidate Geographic Information - Well Known Text for coordinate reference systems standard | OGC”. [Online]. Available at: http://www.opengeospatial.org/standards/requests/112. [Acessado: 04-out-2014].

[17] “Geography Markup Language | OGC”. [Online]. Available at: http://www.opengeospatial.org/standards/gml. [Acessado: 09-set-2014].

[18] “Simple Feature Access - Part 1: Common Architecture | OGC”. [Online]. Available at: http://www.opengeospatial.org/standards/sfa. [Acessado: 09-set-2014].

[19] M. J. Egenhofer, J. Sharma, e D. M. Mark, “A critical comparison of the 4-intersection and 9-intersection models for spatial relations: formal analysis”, in *AUTOCARTO-CONFERENCE-*, 1993, p. 1–1.

[20] J. Renz, “Maximal tractable fragments of the region connection calculus: A complete analysis”, in *IJCAI*, 1999, p. 448–455.

[21] “Beginning Microsoft SQL Server 2012 Programming”. [Online]. Available at: http://it-ebooks.info/book/738/. [Acessado: 04-out-2014].

[22] “Microsoft SQL Server 2012, como comprar e quais os preços em Lisboa, portugal”. [Online]. Available at: http://www.softinmotion.pt/produtos/servidores/sql-server.aspx. [Acessado: 09-set-2014].

[23] “Web Map Service | OGC”. [Online]. Available at: http://www.opengeospatial.org/standards/wms. [Acessado: 09-set-2014].

[24] “OpenLink Virtuoso Universal Server”, *OpenLink Virtuoso Universal Server*. [Online]. Available at: http://virtuoso.openlinksw.com/. [Acessado: 09-set-2014].

[25] “Parliament High-Performance Triple Store”. [Online]. Available at: http://parliament.semwebcentral.org/. [Acessado: 09-set-2014].

[26] J. Rincón Cinca, “XML y web semántica: Bases de datos en el contexto de la web semántica”, 2012.

[27] “Open Sahara - OpenDataNederland.org”. [Online]. Available at: http://opendatanederland.org/nl/open-data-tool/open-sahara. [Acessado: 09-set-2014].

[28] “PostgreSQL: The world’s most advanced open source database”. [Online]. Available at: http://www.postgresql.org/. [Acessado: 09-set-2014].

[29] “PostSIG — Spatial and Geographic Objects for PostgreSQL”. [Online]. Available at: http://postSIG.net/. [Acessado: 09-set-2014].

[30] “Página de download de informação geográfica”. [Online]. Available at: http://mapas.ine.pt/download/index2011.phtml. [Acessado: 04-set-2014].

[31] [Online]. Available at: http://download.geofabrik.de/europe/portugal.html. [Acessado: 09-set-2014].

[32] “Esri Software Products | A Complete SIG Mapping Platform”. [Online]. Available at: http://www.esri.com/products. [Acessado: 04-out-2014].

Glossário

ArcSIG é um sistema de informação geográfica (SIG) para trabalhar com mapas e informação geográfica. Permite criar e utilizar mapas, compilar dados geográficos, analisar informações mapeadas, compartilhar e descobrir informação geográfica e gerir informação geográfica numa base de dados.

DSS - *decision support system* é um sistema de informação baseado no computador que suporta a tomada de decisões sobre negócios ou questões organizacionais. Estes sistemas suportam os níveis de gestão, operação e planificação de uma organização. Estes sistemas podem ser completamente automatizados, humano ou combinação de ambos.

Egenhofer's 9-intersection model é um modelo topológico e um padrão para descrever relações espaciais de duas regiões (duas geometrias em duas dimensões), na geometria, topologia geoespacial e domínios relacionados com a análise espacial.

EPSG - *European Petroleum Survey Group* fundado em 1986 e absorvido por OGP em 2005, era uma organização científica com relacionamento com a indústria de petróleo europeu composto por especialistas que trabalham em geodesia aplicada, pesquisa e cartografia relacionada com a exploração de petróleo.

ESRI (*Environmental Systems Research Institute*) - é uma empresa Americana, fundada em 1969, especializada na produção de sistemas de informação geográfica e outras soluções na área de informações geográficas.

GML (*Geography Markup Language*) - declara recursos geográficos utilizando anotações XML. Pode ser utilizado como linguagem de modelação para sistemas geográficos ou como formato aberto de troca de informação geográfica.

IBM DB2 - é um sistema de gestão de base de dados relacionais produzidos pela IBM. Funcionam em servidores baseados em sistemas *Unix, Windows* ou *Linux*.

INE (Instituto Nacional de Estatística) – organismo responsável por produzir e divulgar informação estatística oficial, promovendo a coordenação, o desenvolvimento e a divulgação da atividade estatística em Portugal.

MySQL – Sistema de gestão de base dados que pode ser usado em várias plataformas e atualmente é propriedade da *Oracle Corporation*. Apresenta uma boa compatibilidade com diversas linguagens, tais como *Java* ou *Python*, bom desempenho e de fácil utilização, suportando a linguagem SQL. Suporta também *triggers*, *cursors* e *stored procedures*, por exemplo.

OGC (*Open Geospatial Consortium*) – é uma organização voluntária internacional de consenso, fundada em 1994, por mais de 400 organizações comerciais e governamentais, sem fins lucrativos em todo o mundo que colaboram num processo de consenso incentivando o desenvolvimento e implementação de padrões abertos para conteúdo geoespacial e de serviços, processamento e partilha de dados.

OGC Simple Feature é um padrão da OGC e ISO que especifica o modelo comum de armazenamento e acesso de dados geográficos bidimensionais (ponto, linha, polígono, múltiplos pontos, múltiplas linhas, etc.)

OpenStreetMap (OSM) é um projeto de mapeamento colaborativo para criar um mapa livre e editável do mundo, inspirado em páginas como a Wikipédia.

Oracle – é um sistema de gestão de base de dados, inventado nos anos 70, editado pela sociedade do mesmo nome (*Oracle coorparation*) líder mundial das bases de dados. A Oracle também criou a linguagem de programação PL/SQL, utilizada no processamento de transações.

PostgreSQL – O PostgreSQL é um sistema de gestão de base dados relacional. É um sistema *open-source*, que teve início em 1985, estando atualmente na versão 9. Permite diversas opções aos seus utilizadores, tais como consultas complexas, uso de chaves estrangeiras, *triggers,* integridade nas transições, suporte a ambientes concorrentes, estrutura para suportar dados georreferenciados, entre outros.

RCC (*Region Connection Calculus***)** – serve para representação espacial qualitativa e racional. RCC descreve regiões de uma forma abstrata (no espaço Euclidiano ou topológico) através das suas possíveis relações.

SQL-Server – desenvolvido pela Microsoft, o SQL-Server é um sistema de gestão de base dados cuja criação foi iniciada em 1988. Possui diversas edições, cada uma com diferentes características, de forma a adaptar-se a diferentes pessoas ou diferentes sistemas. Possui suporte para *triggers* e *stored procedures,* por exemplo.

WGS 84 (*World Geodetic System*) é uma norma usada em cartografia de origem geocêntrica utilizado pelo GNSS do DoD, e pelo sistema de posicionamento global - (GPS), definida em 1984. É composta por um sistema de coordenadas para a terra, uma superfície de referência esferoidal padrão (a base ou elipsoide de referência) para dados de altitude, e uma superfície gravitacional equipotencial (o geoide) que define o nível médio do mar.

WKT (*Well-Known Text*) é uma linguagem de marcação de texto para representar objetos de geometria vetorial em mapas e transformações entre sistemas de referência espacial.