Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Processamento Analítico Espacial e Exploratório Integrando Dados Estruturados e Semiestruturados

Daniel Farias Batista Leite

Campina Grande, Paraíba, Brasil

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Processamento Analítico Espacial e Exploratório Integrando Dados Estruturados e Semiestruturados

Daniel Farias Batista Leite

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande – Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Metodologia e Técnicas de Computação

Cláudio de Souza Baptista, Ph.D.
(Orientador)

Campina Grande, Paraíba, Brasil

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

L533p Leite, Daniel Farias Batista.

Processamento analítico espacial e exploratório integrando dados estruturados e semiestruturados / Daniel Farias Batista Leite. – Campina Grande, 2016.

112f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2016.

"Orientação: Prof. Ph.D. Claúdio de Souza Baptista".

Referências.

1. Tecnologia da Computação. 2. *Business Intelligence* (Análise de Dados). 3. *Exploratory* SOLAP (Ferramentas Exploratórias). 4. Web Semântica. I. Baptista, Claúdio de Souza. II. Título.

CDU 004(043)

Resumo

Tecnologias de *Business Intelligence* (BI) têm sido utilizadas com sucesso para fins de análise de dados. Tradicionalmente, essa análise é realizada em um contexto restrito e bem controlado, onde as fontes de dados são estruturadas, periodicamente carregadas, estáticas e totalmente materializadas. Atualmente, há uma diversidade de dados nos mais diversos formatos, a exemplo de RDF (*Resource Description Framework*), um formato semiestruturado, semanticamente rico e externo à infraestrutura de BI. Embora tal formato seja enriquecido semanticamente, e muitas vezes possua um componente espacial, realizar a análise é um desafio. Nessa perspectiva, uma nova categoria de ferramentas analíticas vem surgindo. As ferramentas exploratórias (*Exploratory OLAP*), como são conhecidas, se caracterizam pela descoberta, aquisição e integração de dados externos em ambientes comuns de análise. Do nosso conhecimento, até a presente data, existem apenas duas ferramentas exploratórias propostas na literatura e elas apresentam duas grandes limitações: exploram apenas fontes de dados estruturadas; e não há exploração do componente espacial dos dados integrados. São ferramentas exploratórias OLAP, e não ferramentas exploratórias SOLAP. Baseando-se nessas ferramentas, este trabalho propõe uma abordagem exploratória SOLAP que integra dados semiestruturados espaciais semânticos com fontes de dados estruturados espaciais tradicionais. Um sistema, denominado ExpSOLAP, que dá suporte a consultas SOLAP *on-line* sob as duas fontes de dados foi desenvolvido. Por fim, o sistema ExpSOLAP é avaliado através de um exemplo prático, no contexto da base de dados obtida no *Linked Movie Data Base*, utilizando RDF e banco de dados relacional. Foram formuladas consultas que validaram a análise convencional e espacial na exploração de ambas fontes de dados.

Palavras chaves: *Business Intelligence*, *Exploratory* SOLAP, Web Semântica, Ontologia, *Linked Data*.

Abstract

Business Intelligence (BI) technologies have been successfully applied for data analysis purposes. Traditionally, such analysis is performed in well-controlled and restricted context, where data sources are structured, periodically loaded, static and fully materialized. Nowadays, there is a plenty of data in different formats such as the Resource Description Framework (RDF), a semi-structured and semantically rich format external to the BI infrastructure. Although such data formats are enriched by semantics and contains a spatial data component, performing data analysis is challenging. As a result, the Exploratory OLAP field has emerged for discovery, acquisition, integration and query such data, aiming at performing a complete and effective analysis on both internal and external data. To the best of our knowledge, there are only two exploratory tools proposed in the literature and they have two major limitations due to only structured data sources can be explored and there is no exploration of the spatial component of the integrated data. While they are exploratory OLAP tools, they are not exploratory SOLAP tools. Based on these tools, this work proposes an Exploratory SOLAP approach that integrates semantic spatial semi-structured data with traditional spatial structured data sources. A system named ExpSOLAP, which supports online SOLAP queries on both data sources, was developed. Finally, a case study was carried out in order to evaluate the ExpSOLAP system based on a dataset originating from the Linked Movie Data Base and using RDF and relational datasets. The formulated queries enabled to validate the conventional and spatial analysis from both data sources.

Keywords: Business Intelligence, Exploratory SOLAP, Semantic Web, Ontology, Linked Data.

Agradecimentos

Primeiramente a Deus por me proporcionar mais essa conquista, me provendo de sabedoria e forças para concluir este trabalho de dissertação.

Aos meus pais, Alexandre e Rosana, pelo exemplo, princípios éticos e morais, pela força, suporte, apoio, motivação e dedicação ao longo de toda a minha vida. As minhas irmãs, Debora, Rachel, Suzana e Izabel, pela companhia durante todos esses anos.

À minha família.

Agradeço à minha namorada, Elisa Diniz, pelo apoio, pela presença, pelo carinho e compreensão, sobretudo nos momentos de maior dificuldade.

Ao meu orientador e amigo, Cláudio de Souza Baptista, pelo incentivo, discussões, brigas e, principalmente, pela paciência e confiança dedicada, desde a graduação, à minha formação acadêmica.

Ao meu amigo e parceiro de projetos, Júlio Henrique Rocha.

À Hugo Feitosa de Figueiredo e Tiago Eduardo da Silva, pela amizade, sensatez, conselhos e palavras sábias.

À José Amilton Moura Acioli Filho, pelo companheirismo e colaboração na realização deste trabalho.

À Universidade Federal de Campina Grande (UFCG) e ao Laboratório de Sistemas de Informação (LSI) por toda minha formação acadêmica, onde pude crescer profissionalmente e pessoalmente. Agradeço a todos integrantes do LSI, em especial, Maxwell Guimarães de Oliveira, Yuri Lacerda, Ana Gabrielle, Tiago Leite, André Luiz Alves e Davi Serrano.

Agradeço aos grupos de amigos das horas de descontração: Equipe Nasa, Galera de Dona Inês e Ap-22.

A todas as pessoas que participaram e ajudaram, diretamente ou indiretamente, nessa conquista e que não tiveram o nome citado nestes agradecimentos.

À CAPES e ao CNPq pelo apoio financeiro.

Sumário

Lista de Abreviaturas e Siglas

BI	*Business Intelligence*
DW	*Data Warehouse*
ETC	Extração, Transformação e Carga
ETQ	*Extract, Transform and Query*
JDBC	*Java Data Base Connectivity*
OLAP	*On-line Analytical Processing*
RDF	*Resource Description Framework*
RDFS	RDF *Schema*
SDW	*Spatial* DW
SIG	Sistema de Informação Geográfica
SOLAP	*Spatial OLAP*
SPARQL	*SPARQL Protocol and RDF Query Language*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Location*
URN	*Uniform Resource Names*
VQL	*Visual Query Language*
W3C	*World Wide Web Consortium*

Lista de Figuras

Lista de Quadros

Lista de Código Fonte

Capítulo 1 – Introdução

Neste capítulo é introduzida a problemática envolvida no trabalho proposto nesta dissertação. Na seção 1.1, é contextualizada a área na qual este trabalho de dissertação está inserido, bem como definido o problema a ser explorado. Na seção 1.2, são descritas as questões de pesquisa consideradas no escopo da pesquisa desta dissertação. Na seção 1.3, são elencados os objetivos, gerais e específicos, do trabalho proposto nesta dissertação. Na seção 1.4, são destacadas as contribuições deste trabalho. Por fim, na seção 1.5, é apresentada a estrutura restante desta dissertação.

1.1 Contextualização e Problemática

Nos últimos anos, devido à necessidade constante de aprimorar o processo de tomada de decisão, diferentes sistemas de suporte à decisão, denominados sistemas de *Business Intelligence* (BI), vêm sendo desenvolvidos para atender aos gestores das empresas.

As ferramentas de BI têm por intuito oferecer novas funcionalidades aos seus clientes, além de apoiar o processo de tomada de decisão. Para isso, capturam dados de *Data Warehouses* (DW), transformando-os em informações úteis, permitindo também a análise histórica, corrente e preditiva das operações de negócio de uma empresa. Dentre as ferramentas de BI mais utilizadas estão as denominadas *On-line Analytical Processing* (OLAP), que proporcionam a rápida exploração e análise dos dados através de tabelas, gráficos, relatórios dinâmicos, *dashboards* e *scoreboards* (Percival e Singh, 2012).

Outrossim, grande parte dos dados vem incorporada de um componente espacial, seja endereço, nome de um Ponto de Interesse (POI) ou coordenadas geográficas. Franklin (1992) estima que a proporção dos dados desta modalidade atinja 80%. Essa peculiaridade aguçou o interesse das organizações em explorar tais dados, com o objetivo de obter novas informações que ajudem os gestores a tomarem as melhores decisões levando em consideração o componente espacial. Dessa forma, surgiu uma nova classe de aplicação conhecida como *Spatial* OLAP (SOLAP), capaz de realizar análises como uma ferramenta OLAP, enriquecendo esta análise com a exploração da dimensão espacial, apresentando os resultados por meio de mapas (Rivest et al., 2005). Desde então, várias pesquisas vêm sendo desenvolvidas, abordando os mais diversos problemas, a exemplo de consultas espaciais em ambiente multidimensional (Silva et al., 2010) e, principalmente, a integração dos dados espaciais em ferramentas OLAP, originando as soluções *JMap Spatial OLAP* (Rivest et al.,

2005), *GeoWOlap* (Bimonte, Tchounikine, & Miquel, 2007), *SOVAT* (Scotch e Parmanto, 2005) e *Framework GeoBI* (Silva, 2013).

Tradicionalmente, as ferramentas OLAP/SOLAP foram projetadas para realizar análises em um contexto restrito e bem controlado. As fontes de dados são internas (corporativas), estruturadas, materializadas e periodicamente carregadas, sendo submetidas a um processo de Extração, Transformação e Carga (ETC). Não obstante, há uma considerável quantidade de dados, ricos em conteúdo, em um mundo externo que, sendo incorporadas às ferramentas analíticas tendem a aperfeiçoar o processo de tomada de decisão.

Os dados externos estão disponíveis, em sua grande maioria, na web. Essa disponibilidade de conteúdo se deu em decorrência da rápida evolução da rede, motivada pela interatividade entre os serviços oferecidos e pela colaboração entre seus usuários. Por possuir essas características, a web foi classificada como Web de Documentos ou Web 2.0.

Apesar da ampla diversidade de dados, estes não estão interligados e também há uma heterogeneidade entre os modelos, o que dificulta a integração dos mesmos. Existem ligações entre páginas, mas não entre dados. Nessa perspectiva, a Web Semântica (também conhecida como Web de Dados ou Web 3.0) foi proposta como uma extensão da Web de Documentos, com o intuito de enriquecer, semanticamente, a informação. Para tanto, adiciona significado bem definido aos dados, possibilitando que tanto pessoas quanto máquinas possam reutilizá-los (Berners-Lee et al., 2001). Além disso, os dados são disponibilizados em um formato padrão, gerenciável e passam a ser interligados. A essa coleção de conjuntos de dados interligados na web dá-se o nome de *Linked Data* (Bizer et al., 2009).

Os dados externos obtidos na web revelam-se valiosos porque trazem consigo novas informações, como, por exemplo, preferências e opiniões de usuários, além de também incluir, em alguns casos, a geo-localização. Destarte, é essencial a incorporação desses dados, de forma integrada aos dados corporativos, para a extração de conhecimentos relevantes que deem suporte ao processo de tomada de decisão.

Entretanto, a incorporação desses dados em ferramentas analíticas não é uma tarefa trivial e está cercada de desafios, a exemplo da diferença entre esquemas e o provisionamento dos dados – facilidade para acesso e integração dos dados. Nesse contexto destacam-se os dados disponíveis na Web Semântica. Os dados têm grande disponibilidade, estão interligados a várias outras fontes de dados (*Linked Data*) e são gerados em uma velocidade alta, muitas vezes na forma de fluxo contínuo (*data streaming*), estando disponíveis, em sua maioria, em formato semiestruturado, compostos por um conjunto de triplas (<*sujeito, predicado,*

objeto>), através do arcabouço RDF – *Resource Description Framework* (Lassila e Swick, 1999).

Esse cenário, ao mesmo tempo em que reforça a oportunidade e importância do uso de dados semiestruturados no processo de tomada de decisão (Inmon et al., 2008), corrobora para um grande desafio: enriquecimento de dados internos estruturados por meio de fontes externas semiestruturadas e, consequentemente, integração desses dados em ferramentas analíticas. A integração entre fontes de dados heterogêneas é um problema atual e vem sendo estudada por vários autores, como por exemplo Kettouch et al. (2015), Kimoto et al. (2015) e Brito et al. (2014).

Elucidando a importância da integração entre fontes de dados heterogêneas, um gestor de uma corporação pode detectar em sua base operacional interna que não há vendas de um determinado produto para o estado da Paraíba e, através da incorporação de dados externos (*Twitter, Facebook*) ser capaz de constatar o porquê dessa rejeição, fomentando a elaboração de estratégias direcionadas para o aumento de vendas desse produto em tal localidade. De forma semelhante, um governante, ao visualizar dados internos da conjuntura socioeconômica do país, tem a possibilidade de integrar dados externos relativos ao orçamento federal. Estes últimos estão disponíveis, em RDF, no portal nacional de dados abertos[1]. Com tais fontes de dados correlacionadas, o governante pode extrair conhecimento que auxilie no planejamento de novas políticas socioeconômicas.

Nessa perspectiva, uma nova categoria de ferramentas analíticas vem surgindo. As ferramentas exploratórias (do inglês *Exploratory* OLAP), como são conhecidas, caracterizam-se pela descoberta, aquisição e integração de dados externos em ambientes comuns de análise (Abelló et al., 2015). Os autores ainda propõem uma categorização das ferramentas exploratórias baseada em cinco critérios: materialização, transformações, frequência de integração, estruturação e extensibilidade.

Diante da pesquisa de anterioridade realizada na literatura, foram encontrados apenas dois trabalhos no âmbito de ferramentas exploratórias: Ibragimov et al. (2015) e Furtado et al. (2015). Tais trabalhos propõem ferramentas exploratórias, mas apresentam duas grandes limitações. Em primeiro lugar, as ferramentas propostas não exploram fontes de dados heterogêneas. Uma segunda limitação está relacionada a não exploração do componente espacial dos dados integrados. São ferramentas exploratórias OLAP, e não ferramentas exploratórias SOLAP.

[1] http://dados.gov.br/dataset/orcamento-federal

1.2 Questões de Pesquisa

As seguintes questões de pesquisa foram consideradas e testadas no escopo da pesquisa descrita nesta dissertação:

- Q1: considerar a dimensionalidade dos dados (convencionais, espaciais, temporais e espaço-temporal) como um novo critério de categorização de ferramentas exploratórias, estendendo os critérios propostos por Abelló et al. (2015), aprimora a classificação das ferramentas exploratórias?
- Q2: existe possibilidade de desenvolver uma ferramenta exploratória SOLAP, integrando fontes de dados estruturadas com fontes de dados semiestruturadas?

1.3 Objetivos

Considerando-se o contexto da necessidade de tornar o processo de tomada de decisão mais completo e eficaz, a partir da integração de dados internos com fontes externas, a pesquisa aqui realizada tem como objetivo propor um ambiente *Exploratory* SOLAP, denominado ExpSOLAP. Esse ambiente possibilita a exploração visual e espacial de várias fontes de dados nos mais variados tipos de estruturação e formatos.

Para atingir esse objetivo geral, foram traçados os seguintes objetivos específicos:

- Utilizar uma linguagem de consulta visual para facilitar o processo de análise dos dados;
- Possibilitar a criação de gráficos, relatórios e mapas dos dados estruturados e semiestruturados;
- Explorar de forma integrada fontes de dados estruturados e semiestruturados semânticos, através do uso de ontologia representada no formato RDF; e
- Explorar as características espaciais dos dados.

1.4 Contribuições

A principal contribuição deste trabalho está em definir uma solução *Exploratory* SOLAP capaz de acessar e integrar diferentes fontes de dados, estruturadas e semiestruturadas, incorporando dados convencionais e espaciais. Do nosso conhecimento, até a presente data, não foi encontrada nenhuma solução para *Exploratory* SOLAP na literatura.

Ademais, considerando o universo de *Exploratory* OLAP, esta dissertação contribui com a possibilidade de explorar fontes de dados heterogêneas, até então não endereçadas nos trabalhos relacionados.

Por fim, propõe-se uma nova categorização para ferramentas exploratórias, estendendo a categorização inicialmente proposta por Abelló et al. (2015) com a adição de um sexto critério: dimensionalidade.

1.5 Estrutura da Dissertação

O restante deste trabalho está organizado da seguinte forma:

- No *Capítulo 2,* é apresentada uma fundamentação teórica abordando os conceitos que conduzem ao entendimento do tema em estudo;
- No *Capítulo 3*, são apresentados os trabalhos já realizados a respeito da temática em estudo ou que estejam de alguma forma a esta relacionados. Os principais trabalhos científicos são descritos, permitindo a realização de uma avaliação comparativa que expõe pontos fontes e fracos de cada solução;
- No *Capítulo 4*, é apresentada a solução ExpSOLAP, descrevendo detalhadamente sua arquitetura e todos os seus componentes;
- No *Capítulo 5*, é endereçado um exemplo prático no contexto de Filmes com o intuito de avaliar e validar a solução proposta; e
- No *Capítulo 6*, são apresentadas as considerações finais deste trabalho e as discussões sobre propostas para trabalhos futuros.

Capítulo 2 – Referencial Teórico

Neste capítulo são apresentados os pressupostos teóricos que fundamentam esta dissertação. Para tanto, o mesmo está organizado em quatro seções. Na seção 2.1, são abordadas as tecnologias da Web Semântica, apresentando-se seus conceitos, linguagens e padrões. Os temas relacionados à área de *Business Intelligence,* a exemplo de *Data Warehouse, OLAP, Spatial* DW, SOLAP e Linguagem de Especificação Visual, são descritos na seção 2.2. Na seção 2.3, dá-se ênfase às ferramentas analíticas exploratórias, expondo-se as suas características e categorizações. Por fim, na seção 2.4, são elencadas as considerações finais.

2.1 Web Semântica

A Web Semântica foi originalmente idealizada por Berners-Lee (2000) e, posteriormente, promovida pelo consórcio *World Wide Web* (W3C). A proposta de Berners-Lee era criar uma infraestrutura na Web de tal forma que as informações não fossem apenas entendidas por humanos, mas também para serem processadas e compreendidas por máquinas.

Em um trabalho posterior, Berners-Lee et al. (2001) descrevem uma aplicação que demonstra os benefícios de sua proposta e definem conceitos, padrões e tecnologias chaves para a Web Semântica, definindo a sua arquitetura, a qual é organizada em camadas, em uma estrutura similar a uma pirâmide (Figura 1).

A camada inferior da arquitetura, na qual se situam o Unicode e URI (*Uniform Resource Identifier*), tem por objetivo facilitar o intercâmbio de dados e fornecer meios para identificação única dos objetos na Web Semântica. O Unicode é um padrão no qual todos os caracteres, independente de codificação ou linguagem, sejam representados computacionalmente. O URI é um identificador único de recursos na internet, sendo composto de URL (*Uniform Resource Location*) e de URN (*Uniform Resource Names*). O primeiro identifica a localização (endereço) do recurso, enquanto o segundo identifica o nome do recurso.

A camada XML contém as definições de esquema e de *namespaces* que poderão ser utilizadas nas camadas superiores, a exemplo da camada RDF e RDF *Schema* (RDFS). Os *namespaces* são um conjunto de símbolos e abreviaturas utilizados para organizar objetos, de modo que esses possam ser utilizados referenciando o nome abreviado. Esta última camada é composta por dados em formatos RDF que permitem a representação de conceitos, regras e a definição de vocabulários utilizando marcações XML. A camada Ontologia suporta a

evolução do vocabulário RDFS, contemplando especificações formais e explícitas de conceitos expressos na camada inferior. A camada de Assinatura Digital provê a incorporação de mecanismos de segurança que garantem a confiabilidade da informação.

A camada de Lógica permite a escrita de regras enquanto a camada de Prova executa as regras e avalia, em conjunto com os mecanismos da camada de Confiança, se a regra é válida ou não. Essas três camadas superiores ainda estão sendo pesquisadas pelo W3C e não há uma padronização definitiva.

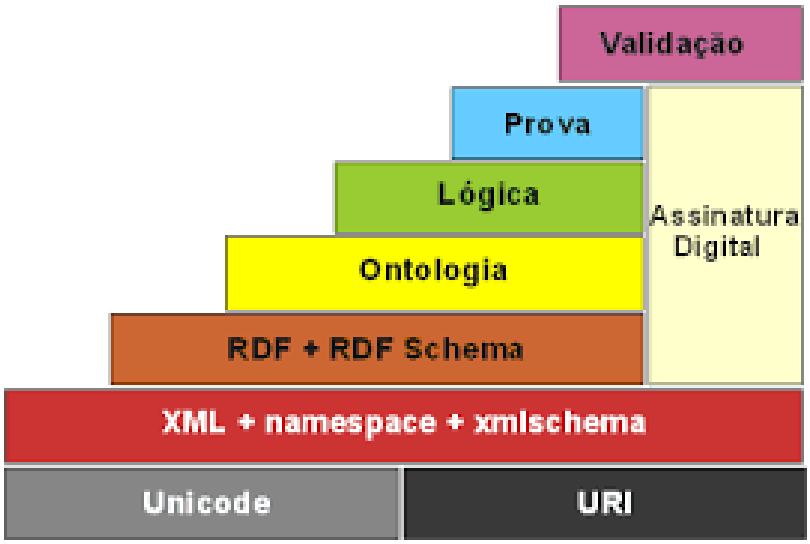

Figura 1: Camadas da Web Semântica. Fonte: http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_001-04.pdf

2.1.1 Ontologias

Segundo Noy e McGuinness (2001), uma ontologia de domínio define um vocabulário comum para o compartilhamento de informações de um domínio. Esse vocabulário inclui definições a respeito de conceitos básicos e relacionamentos presentes no domínio. De modo geral, as ontologias são modelos de domínio que possuem duas características especiais: (1) são expressas utilizando linguagens formais que possuem uma semântica bem definida; e (2) são construídas utilizando significados previamente acordados entre os membros de uma determinada comunidade.

Com base nessas peculiaridades, há dois formatos existentes, ambos reconhecidos pela W3C, para o desenvolvimento de ontologias: RDF e OWL (*Ontology Web Language*).

2.1.1.1 RDF e RDFS

O RDF é um arcabouço baseado em XML, que permite a representação de conceitos básicos e seus relacionamentos. A W3C define RDF como modelo padrão para intercâmbio de dados na

Web Semântica, visto que esse formato apresenta características que facilitam a fusão de dados mesmo com a heterogeneidade de esquemas, além de ser flexível e extensível para representar informações.

Um arquivo RDF é composto por várias declarações RDF. Uma declaração RDF, por sua vez, é composta por uma tripla: <*sujeito, predicado, objeto*>. O sujeito e objeto são recursos, enquanto o predicado é uma propriedade utilizada para descrever o recurso e seu valor. Os três componentes da tripla são representados por conceitos abstratos, representados por URIs, mas podem ser simplificados com o uso de prefixos (*namespaces*). Os objetos também podem ser literais, ou seja, tipos primitivos, a exemplo de *String*, inteiro, booleano etc.

Outra forma de representar as declarações em RDF é através de um grafo, no qual o sujeitos e objetos são os nós e os predicados são as arestas ligando ambos. No Código Fonte 1 é ilustrado uma representação de declaração RDF em tripla, enquanto na Figura 2 é ilustrada a mesma declaração em formato de grafo.

Código Fonte 1: Exemplo de representação de uma declaração RDF em tripla

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:caracteristica="http://www.lsi.ufcg.edu.br/caracteristica#">
  <rdf:Description
     rdf:about="http://www.lsi.ufcg.edu.br/roupa#camisa">
     <caracteristica:tamanho>G</caracteristica:tamanho>
     <caracteristica:cor
        rdf:resource="http://www.lsi.ufcg.edu.br/cor#azul"/>
  </rdf:Description>
</rdf:RDF>
```

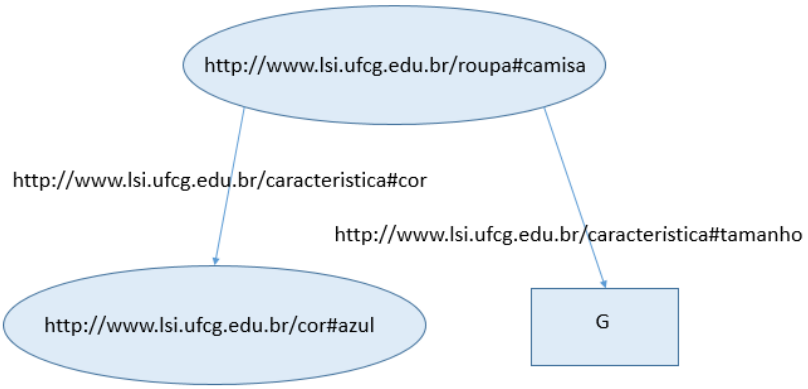

Figura 2: Representação de uma declaração RDF e grafo. Adaptado de https://jena.apache.org/tutorials/rdf_api_pt.html

O RDF *Schema* serve para descrever as propriedades dos atributos e classes que foram modelados através do RDF. Além disso, permite que sejam especificados relacionamentos hierárquicos entre as propriedades e as classes (W3C, 2004).

2.1.2 SPARQL

***S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage* (SPARQL) é uma linguagem de consulta, recomendada pela W3C, específica para pesquisar dados armazenados em formato RDF. De acordo com a gramática de SPARQL (W3C, 2008), há quatro tipos de consultas: *Select Query, Construct Query, Describe Query* e *Ask Query.* Cada tipo de consulta tem uma estrutura única, com cláusulas próprias. No entanto, duas cláusulas estão presentes em todas as modalidades de consultas, as cláusulas *Prefix Declarations* (declaração de prefixos) e *Where.*

A cláusula de declaração de prefixos é utilizada para abreviar URIs e contém uma lista de todos os URIs (prefixos) que são necessários à consulta, permitindo a importação de conceitos e o uso destes. Por exemplo, ao declarar o prefixo *foaf: <http://xmlns.com/foaf/0.1/>*, é possível utilizar conceitos do vocabulário, a exemplo *foaf:Person* como predicado de alguma tripla. A cláusula *Where* contém critérios de pesquisa para restringir o retorno, sendo constituída por padrões de grafo (*Query Patterns* e *Group Graph Pattern*), ou seja, triplas. A seguir, são descritos os objetivos específicos de cada modalidade de consulta.

- ***Select***: retorna todas, ou um subconjunto, de triplas de acordo com os padrões presentes na cláusula *Where*;
- ***Construct***: retorna um grafo RDF construído por variáveis dos padrões presentes na cláusula *Where*;
- ***Ask***: retorna um valor booleano que indica se o padrão da consulta corresponde ou não com a base de dados consultada;
- ***Describe***: retorna um grafo RDF que descreve todos os recursos (sujeitos e objetos) encontrados na base de dados consultada.

No âmbito deste trabalho de dissertação, apenas consultas do tipo *Select* foram consideradas, visto que as ontologias exploradas serão trabalhadas apenas através de operações de consulta, não sendo necessário realizar operações de remoção, atualização ou inserção dos dados. Além das cláusulas *Prefix Declaration* e *Where*, o tipo de consulta *Select* é constituída por outras três cláusulas, são elas:

- ***Select Clause*** **(Cláusula de Resultado)**: composta por variáveis que identificam e selecionam quais informações, e como, serão retornadas na consulta.
- ***Dataset Clause*** **(Definição da base a ser consultada**): é indicada para afirmar quais arquivos, ou grafos RDF, serão consultados, equivalendo à cláusula “FROM” na

linguagem SQL. É uma cláusula opcional, podendo ser omitida se a fonte de dados for importada na execução da consulta através da declaração de prefixos.

- ***Solution Modifiers*** **(Modificadores)**: responsável por rearranjar os resultados, seja por meio de *slicing* (selecionar uma porção dos resultados), *ordering* (ordenar os resultados) ou *group by* (agrupar os resultados com base em alguma função de agregação).

O algoritmo exibido no Código Fonte 2 ilustra um exemplo de consulta SPARQL do tipo *Select* para o seguinte questionamento: "Quais os nomes de todas as pessoas cadastradas na base de dados *foaf* ?". Para formulação dessa consulta, a fonte de dados *foaf* foi importada na cláusula *Dataset* (linha 4). Na cláusula de resultados, há declaração apenas de uma variável, ?nome, que é a informação que se deseja retornar. Já na cláusula *Where,* é criado um padrão que corresponde a declarações (triplas) da propriedade "nome das pessoas", justificando a utilização do predicado *foaf:name* (linha 6). A utilização da notação simplificada como predicado foi possível a partir da importação do vocabulário *foaf* na cláusula de prefixos. Por fim, na cláusula de modificadores, os resultados são ordenados alfabeticamente (linha 8).

Código Fonte 2: Exemplo de consulta SPARQL

```
1    PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2
3    SELECT ?nome
4    FROM <http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf>
5    WHERE {
6       ?person foaf:name ? nome .
7    }
8    ORDER BY ?nome
```

A base de dados semântica a ser explorada pode estar armazenada nativamente, com arquivos RDF locais, ou na web (nuvem). No primeiro caso, existem *frameworks* que permitem a manipulação desses arquivos semanticamente, de forma similar a manipulação de arquivos utilizando a linguagem Java (pacote *java.io*), através de códigos escritos em linguagens de programação. Os *frameworks* têm funcionamento semelhante a um JDBC, a exemplo do *Framework Apache Jena*[2]. Todavia, se a base semântica estiver armazenada na web, há serviços *online*, denominados *SPARQL Endpoint,* que possibilitam realizar consultas utilizando protocolo HTTP.

[2] https://jena.apache.org/

2.1.3 *Linked Data*

Linked Data é a prática de conectar dados na Web, utilizando RDF para criar declarações sobre propriedades ou recursos da rede, bem como para construir declarações sobre os relacionamentos existentes entre os recursos (Watson, 2009). É o conjunto das melhores práticas para publicar e conectar dados na Web (Bizer et al., 2009). Berners-Lee (2006) introduziu algumas dessas práticas através de quatro regras que, quando utilizadas, são capazes de interconectar os dados da Web. As regras são:

- utilizar URIs como nomes de coisas;
- utilizar URIs HTTP para que as pessoas possam encontrar esses nomes;
- fornecer através de URI informações úteis, utilizando padrões (RDF, SPARQL); e
- incluir links para outras URIs, desta forma, mais informações poderão ser descobertas.

2.2 *Business Intelligence*

O termo *Business Intelligence* refere-se a um conjunto de conceitos, metodologias, processos e tecnologias que visam transformar dados brutos em informações úteis e, assim, oferecer suporte à tomada de decisão (Chaudhuri et al., 2011). Nesta seção são descritos alguns conceitos e tecnologias que norteiam a área de BI.

2.2.1 *Data Warehouse* (DW)

Data Warehouse, ou Armazém de Dados, é um banco multidimensional voltado para análise estratégica e tomada de decisão. Inmon (2008) define como uma coleção de dados orientados por temas, integrados, variantes no tempo e não voláteis. Por manipular uma considerável coleção de dados, o DW precisa ser otimizado para facilitar análises complexas e aumentar o desempenho do banco de dados em consultas com milhares de ocorrências. Por esse motivo, o DW é projetado em torno de temas (*subjects*) de interesse das corporações, o que diminui o domínio de consulta.

A característica do DW ser integrado deve-se ao fato de o mesmo capturar dados oriundos de diferentes bases operacionais, armazenados em diversos formatos (*.xls*, *.txt*, pequenos DW - *Data-Marts*, etc.). É variante no tempo pois armazena dados históricos, não mantendo apenas os dados mais recentes; e é não volátil porque não permite operações de remoção (*deletes*) nem atualizações (*updates*), apenas carga de dados e consultas.

A carga de dados de um DW é realizada através de um processo de ETC, o qual tem início com a extração de dados de várias fontes operacionais. Em seguida, os dados são

transformados conforme o esquema utilizado no DW, aplicando-se correções aos dados de má qualidade, para, então, serem adicionados ao DW. A frequência de realização do processo de ETC depende de cada contexto, podendo ser realizada a cada modificação nas fontes operacionais ou em *batches* periódicos.

Os conceitos de fatos, medidas, dimensões e membros são utilizados para modelar DWs. Os temas de interesse são denominados de Fatos e estão associados às medidas - atributos numéricos que se deseja analisar. As dimensões caracterizam os fatos e fornecem uma visão em diferentes perspectivas de um fato. Uma instância da dimensão é denominada de membro; e uma combinação dos membros das dimensões identifica, de forma unívoca, um fato. Exemplificando dentro do contexto comercial, o evento mais importante é uma venda, por isso seria um fato. As medidas de análise poderiam ser o valor e a quantidade de itens vendidos, por exemplo, enquanto que as dimensões seriam o tempo (data da venda), produto e sua categoria e a unidade onde foi vendido o item, permitindo-se uma visualização do fato sob qualquer dessas perspectivas.

Um modelo multidimensional pode ser representado de duas formas, em um banco de dados relacional, por meio do esquema estrela ou do esquema floco de neve. O esquema estrela é caracterizado por ter uma tabela de fatos ligada a várias tabelas de dimensões, onde a primeira armazena os eventos ocorridos e as chaves para as características correspondentes nas tabelas de dimensões. Os relacionamentos entre as tabelas são evidenciados através de um relacionamento 1:N, no qual um registro na tabela de dimensão pode estar ligado a vários registros na tabela de fatos, porém um elemento da tabela de fatos só pode estar ligado a um valor na tabela da dimensão. O esquema estrela é desnormalizado, com armazenamento de dados redundantes, oferecendo um melhor desempenho. O esquema floco de neve, por outro lado, normaliza as dimensões, tornando o esquema mais complexo e menos eficiente (Kimball et al., 2013).

2.2.2 *On-line Analytical Processing* (OLAP)

De acordo com Rivest et al. (2005), para explorar os dados de forma interativa, as soluções de BI mais utilizadas são as ferramentas OLAP. Tais ferramentas suportam a natureza interativa do processo analítico, porque permitem que o usuário explore e navegue por meio de diferentes temas (dimensões), em diferentes níveis de detalhe, e, rapidamente, visualize os fatos ou dados intersectando as dimensões, qualquer que seja o nível de agregação.

As ferramentas OLAP implementam técnicas de análise para explorar modelos multidimensionais denominados cubos. Jensen et al. (2010) definem a estrutura de cubo como

sendo uma estrutura multidimensional utilizada para analisar os dados de forma eficiente e responsável por manter as informações agregadas e sumarizadas. O armazenamento dos dados em estruturada de cubo pode ser realiza seguindo três abordagens distintas, quais sejam: metodologia relacional (ROLAP – *Relational* OLAP), multidimensional (MOLAP – *Multidimensional* OLAP) ou híbrida (HOLAP), a qual combina os métodos ROLAP e MOLAP.

Os cubos multidimensionais organizam as informações em fatos, dimensões, hierarquias, níveis e medidas. As medidas são propriedades que o usuário deseja analisar e estão associadas a um fato. Uma função de agregação é utilizada para combinar vários valores de uma medida em um único valor.

Uma dimensão pode ser organizada em hierarquias de níveis, cada um dos quais representa um nível de detalhe dos dados, a exemplo da hierarquia Cidade -> Estado -> All. O nível All, presente em todas as hierarquias, representa todos os membros da dimensão. Os demais níveis compõem dados mais detalhados. No caso do exemplo, Cidade é o nível mais detalhado.

Outrossim, os operadores OLAP – disponíveis nas ferramentas OLAP, possibilitam que usuários naveguem e explorem os dados em diferentes níveis de detalhes e perspectivas. Dentre as operações OLAP, destacamos *roll-up, drill-down, slice, dice* e *pivot*. Os dois primeiros operadores permitem a navegação entre os níveis da hierarquia, diminuindo (*roll-up*) ou aumentando (*drill-down*) o nível de detalhe. Os operadores *slice* e *dice* são filtros que possibilitam, respectivamente, selecionar um subconjunto dos resultados ou aplicar restrições aos membros de dimensão. O operador *pivot* permite alterar a perspectiva de visualização do cubo. Quando ocorre um pivoteamento as dimensões são rotacionadas, ou seja, elas mudam de eixo.

Uma arquitetura comum de ferramentas OLAP (Figura 3) é compreendida em três camadas: a) camada de dados, que consiste nos dados a serem analisados, geralmente armazenados em um DW; b) o servidor OLAP, também denominado fonte de dados multidimensionais ou servidor de cubos, onde estruturas denominadas cubos são mantidas; e c) camada de aplicação ou cliente OLAP, que acessa os cubos de dados disponibilizados pelo servidor OLAP e permite que os usuários finais analisem os dados utilizando diferentes métodos de visualização e operadores OLAP.

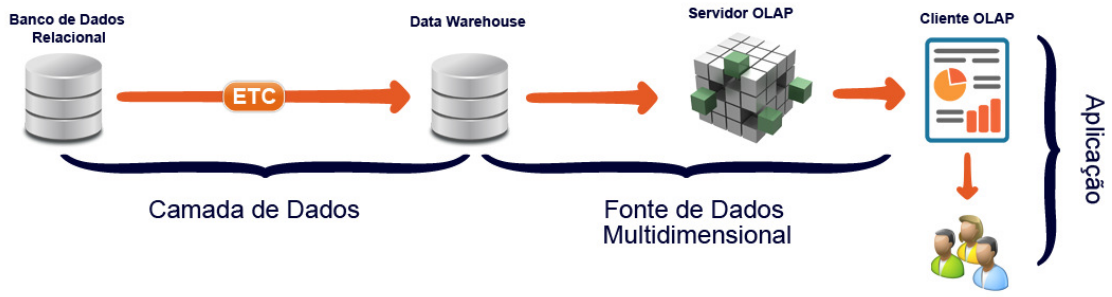

Figura 3: Arquitetura comum de sistemas OLAP

Para consultar informações diretamente no cubo multidimensional, pode-se utilizar a linguagem multidimensional de alto desempenho denominada *Multidimensional Expression* – MDX (MICROSOFT, 2014). A linguagem MDX fornece uma sintaxe rica e poderosa para recuperação e manipulação de dados multidimensionais. De forma similar à SQL, a sintaxe da consulta MDX é composta pelas cláusulas SELECT, FROM e WHERE. A cláusula SELECT deve incluir as variáveis para visualização dos eixos (eixo 0 – colunas, eixo 1 – linhas, etc). Na cláusula FROM nomeia-se a fonte de dados para consulta MDX, enquanto na cláusula WHERE descreve-se o eixo de filtros em uma consulta MDX. O algoritmo do Código Fonte 3 ilustra um exemplo de consulta utilizando a linguagem MDX.

Código Fonte 3: Exemplo de Consulta MDX. Fonte: https://msdn.microsoft.com/pt-br/library/ms144785%28v=sql.120%29.aspx

```
1    SELECT
2       {[Measures].[Sales Amount], [Measures].[Tax Amount] } ON COLUMNS,
3       {[Date].[Fiscal].[Fiscal     Year].&[2002],     [Date].[Fiscal].[Fiscal
     Year].&[2003] } ON ROWS
4    FROM [Adventure Works]
5    WHERE [Sales Territory].[Southwest] )
```

Por outro lado, o intercâmbio de informações entre as ferramentas OLAP (Cliente OLAP) e servidores multidimensionais se dá através do protocolo XMLA – XML *for Analysis* (MICROSOFT, 2014*b*). O XMLA é baseado em outros padrões, a exemplo do XML, SOAP e HTTP; e encapsula, dentre outras informações, a consulta MDX para execução. O algoritmo exibido no Código Fonte 4 ilustra um exemplo de comunicação entre servidor multidimensional e ferramenta OLAP.

2.2.3 *Spatial* DW e *Spatial* OLAP

A extensão do DW convencional para dar suporte aos dados espaciais denomina-se *Spatial DW* (SDW), a qual, além de manter as características tradicionais de um DW, permite a integração de dados georreferenciados através de medidas e dimensões espaciais.

Bédard et al. (2001) classificam a dimensão espacial em três tipos: não geométrica, geométrica ou mista. A dimensão não geométrica utiliza referências espaciais nominais, ou seja, nome de cidades; e por isso é suportado pelos DWs convencionais. A dimensão espacial geométrica é composta, em todos os níveis hierárquicos e em todos os membros, de forma geométricas (polígonos, linhas, pontos) que são georreferenciadas, permitindo uma análise espacial com visualização em mapas. Na dimensão espacial mista, uma hierarquia pode conter níveis geométricos e não geométricos, ou seja, apenas um subconjunto dos níveis hierárquicos é representado por geometrias georreferenciadas.

Código Fonte 4: Exemplo de comunicação via XMLA. Fonte: https://msdn.microsoft.com/en-us/library/ms186691.aspx

```
<soap:Envelope>
   <soap:Body>
      <Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
        <Command>
          <Statement>
            SELECT [Measures].MEMBERS ON COLUMNS FROM [Adventure Works]
          </Statement>
        <Command>
        <Properties>
          <PropertyList>
            <DataSourceInfo>Provider=MSOALP;Data Source=local;/>
            <Catalog>Adventure Works DW Multidimensional</Catalog>
            <Format>Multidimensional</Format>
            <AxisFormat>ClusterFormat</AxisFormat>
          </PropertyList>
        </Properties>
      </Execute>
   </soap:Body>
</soap:Envelope>
```

De forma semelhante, Bédard et al. (2001) propõem três tipos de medidas espaciais. A primeira consiste de uma geometria ou um conjunto de geometrias obtidas pela combinação de múltiplas dimensões espaciais, ou seja, um conjunto de coordenadas computadas por meio de uma operação espacial como merge, união ou intersecção. A segunda resulta de uma métrica espacial ou de um operador topológico, como área ou distância, por exemplo. A última é um conjunto de apontadores para as geometrias armazenadas em outra estrutura ou tecnologia.

A integração de dimensões e medidas espaciais traz grandes benefícios para o processo de tomada de decisão e permite a descoberta de novas informações que, antes, não eram acessíveis, devido à ausência desse tipo de dados, como, por exemplo, a seguinte consulta:

“Qual é o valor total de vendas dos produtos alimentícios nas cidades circunvizinhas de Campina Grande?”.

Com a presença do SDW, surgiu um novo tipo de ferramenta denominada SOLAP. De acordo com Malinowski (2014), o grande diferencial das ferramentas SOLAP é a possibilidade de fornecer aos tomadores de decisão a capacidade de análise de dados espaciais e convencionais em um só ambiente, sem a necessidade de dominar os conceitos geográficos necessários para a manipulação espacial em SIG (Sistemas de Informação Geográfica).

A arquitetura de uma ferramenta SOLAP se assemelha à arquitetura de um sistema OLAP, sendo também composta pelas mesmas três camadas, diferenciando pelo suporte a dados espaciais. Na camada de dados, os dados espaciais podem ser mantidos em um SDW ou em outro ambiente, onde apontadores irão relacionar os dados convencionais aos dados espaciais. Assim, a integração entre os dados espaciais e as ferramentas OLAP pode ser feita por meio de duas abordagens: a) federada, em que os dados espaciais não se encontram no mesmo ambiente dos dados convencionais; ou b) integrada, em que ambos os tipos de dados se encontram no mesmo ambiente (e.g. SDW). O servidor SOLAP é responsável por manter e disponibilizar os cubos espaciais. Por fim, o cliente SOLAP permite a visualização dos dados em tabelas, gráficos e mapas além de possibilitar a navegação nas hierarquias espaciais através dos operadores *spatial roll-up* e *spatial drill-down* e criação de filtros com operadores espaciais.

2.2.4 Linguagem de Especificação Visual (*Visual Query Language – VQL*)

Apesar da linguagem MDX apresentar um alto desempenho ao realizar consultas em cubos multidimensionais, ela apresenta um certo nível de dificuldade para usuários expressarem consultas. As consultas formuladas e os resultados são verborrágicos, de modo que somente usuários com conhecimento da linguagem e do esquema dos dados têm habilidade para formular consultas na linguagem MDX.

Essa metodologia não é adequada para sistemas analíticos, visto que estes prezam pela exploração dos dados de forma interativa. Ademais, a maioria de seus usuários, geralmente, são pessoas que não detêm conhecimento técnico algum para formular consultas utilizando a linguagem MDX. Nesse sentido, grande parte desses sistemas dispõe de uma Linguagem de Especificação Visual (*VQL – Visual Query Language*) que permite que os usuários formulem consultas e extraiam informações de forma amigável e intuitiva.

Basicamente as VQL oferecem componentes gráficos que possibilitam aos usuários formalizarem consultas através desses componentes. Além disso, organizam os dados a serem analisados em tabelas. Isso significa que o usuário deve: 1) selecionar a porção dos dados que deseja analisar; 2) organizar os dados em uma tabela e 3) criar gráficos ou outras representações visuais dos dados a partir da tabela.

2.3 Ferramentas Exploratórias

Segundo Abelló et al. (2015), a principal diferença entre ferramentas exploratórias e ferramentas analíticas tradicionais (OLAP/SOLAP) é a questão da exploração. Exploração de novas fontes de dados, de novas formas de estruturação de dados, de novas formas de integrar dados em conjunto ou, ainda, de novas formas de consultar dados. Os mesmos autores ainda propõem cinco critérios para definir o nível exploratório das ferramentas, quais sejam:

- **Materialização**: relacionado ao nível de materialização dos dados integrados, podendo ser: totalmente materializado, parcialmente materializado, resultado ou virtual. Os níveis desse critério não são aditivos. Em outras palavras, caso uma ferramenta seja classificada como resultado, não significa que ela também incorpore materialização parcial ou total.
    - **Totalmente Materializado**: todos os dados são materializados, por exemplo, em um DW;
    - **Parcialmente Materializado**: pelo menos uma fonte de dados integrada não é materializada;
    - **Resultado**: metodologia que inicialmente extrai dados sob demanda, computa os resultados, materializando apenas estes para futuras requisições; ou
    - **Virtual:** extrai os dados em tempo de execução e integra-os *on-the-fly*.
- **Transformação dos dados**: nível de transformações aplicadas aos dados durante o processo de integração. Essas transformações podem ser leves (*lighweight*) – quando envolvem operações simples e rápidas de serem realizadas, a exemplo de agregações e renomeação de atributos ou complexas (*complex*) – quando envolvem várias transformações, a exemplo de limpeza significativa dos dados, computação de medidas agregadas e dimensões de mudança lenta (*SCD – Slowly Changing Dimensions*). Os níveis do critério não são aditivos.
- **Frequência de Integração**: frequência em que o processo de integração dos dados é realizado. Os níveis desse critério são aditivos, ou seja, ao realizar frequência de

integração em *micro-batches*, a ferramenta exploratória oferece mecanismos de integrar dados de forma periódica.

  - **Periódico:** *batches* com uma quantidade considerável de dados, executados periodicamente (diário, semanal);
  - ***Micro-batches*:** executado com maior frequência (a cada 30 minutos, por exemplo), em quantidades menores de dados;
  - **Sob demanda:** integração de dados a partir de requisição ou solicitação do usuário;
  - **Tempo real:** atualização dos dados frequentemente, quase em tempo real; ou
  - **Contínuo:** integração realizada a cada segundo, com milhares de ocorrências de dados integradas.

- **Estruturação dos dados**: relacionado ao tipo de estruturação das fontes de dados exploradas, podendo ser estruturadas, semiestruturadas ou não estruturadas. Os níveis desse critério são aditivos.
- **Extensibilidade**: refere-se à facilidade de integração de novas fontes de dados à ferramenta exploratória, podendo ser estático, adaptação ou dinâmico. Tais níveis não são aditivos.
  - **Estático**: a integração de novas fontes não é trivial. A ferramenta já está configurada para acessar uma determinada fonte de dados.
  - **Adaptação**: a integração de novas fontes de dados pressupõe um procedimento para mapear os esquemas de tais fontes;
  - **Dinâmico**: a integração é automática, de modo que a ferramenta exploratória, sem intervenção manual, reconhece o esquema da nova fonte de dados e promove a integração.

Os critérios também estão ilustrados em formato de um pentágono (Figura 4). O pentágono mais interno representa as ferramentas analíticas tradicionais, que tem as fontes de dados estruturadas, totalmente materializadas em um DW após transformações complexas e periódicas. A adição de novas fontes de dados nessas ferramentas não é trivial, sendo necessário adaptar todo o processo de ETC. Por outro lado, os pentágonos mais externos representam ferramentas exploratórias. Frise-se que não há ferramenta totalmente exploratória, de modo que se uma ferramenta apresentar avanços em pelo menos um dos critérios, já pode ser considerada exploratória.

Figura 4: Categorização de ferramentas analíticas. Traduzida de Abelló et al. (2015)

Em relação à arquitetura, as ferramentas exploratórias apresentam uma maior complexidade quando comparadas às ferramentas analíticas tradicionais. Isto porque, além de agregar todo o fluxo de uma arquitetura tradicional (dados extraídos, submetidos a um processo de ETC, armazenados em um DW para serem explorados em uma estrutura de cubo multidimensional), a arquitetura de ferramentas exploratórias contém um fluxo adicional responsável pela exploração de dados externos. Este fluxo é particular pois não possui uma área intermediária para armazenamento dos dados, os quais são extraídos, submetidos a transformações *on-the-fly*, para só então ficarem disponíveis para consulta. Daí este processo ser denominado por Abelló et al. (2015) de ETQ – *Extract, Transformation and Query*. Na Figura 5 é ilustrado um exemplo de arquitetura para ferramentas exploratórias.

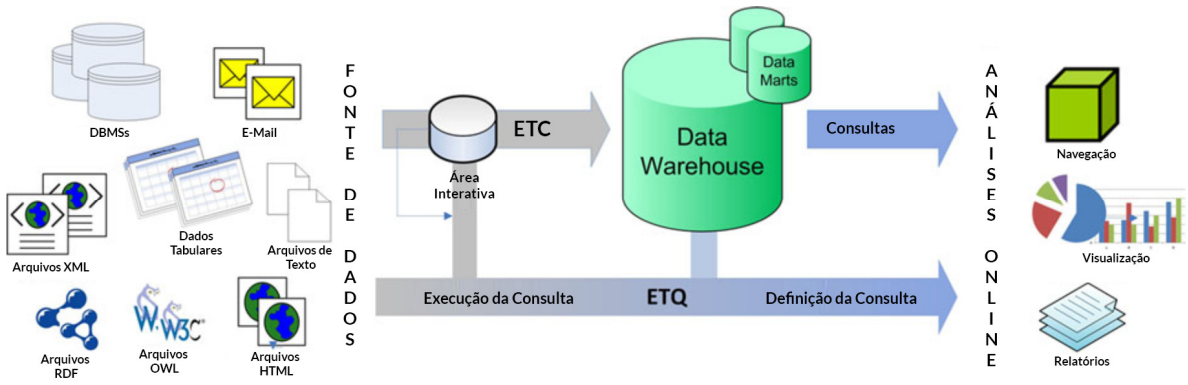

Figura 5: Exemplo de arquitetura para ferramentas exploratórias. Traduzida de Abelló et al. (2015)

2.4 Considerações Finais

Este capítulo apresentou a fundamentação teórica que embasa este trabalho. Foram apresentadas as principais características e conceitos das áreas de Web Semântica, *Business Intelligence* e de ferramentas exploratórias.

No que tange às ferramentas exploratórias, foram discutidos os critérios exploratórios e um fluxo de arquitetura padrão, sendo possível destacar a relevância e a complexidade da integração de dados externos às ferramentas analíticas.

No capítulo seguinte, serão apresentados trabalhos relacionados à pesquisa realizada nesta dissertação.

Capítulo 3 – Trabalhos Relacionados

Neste capítulo, é apresentado um levantamento bibliográfico dos principais trabalhos relacionados a esta pesquisa. A fim de facilitar a descrição dos mesmos e posicionar a solução proposta nesta dissertação com base na literatura existente, os estudos foram divididos em três seções. Na primeira delas (seção 3.1), são apresentados os trabalhos que procuraram integrar fontes de dados heterogêneas utilizando tecnologias de *Business Intelligence*. Em seguida, na seção 3.2, é apresentada uma análise dos trabalhos que exploram dados semânticos, convencionais ou espaciais. Na terceira seção (seção 3.3), são apresentados estudos que integram tecnologias de BI com Web Semântica, a exemplo de ferramentas exploratórias. A solução proposta nesta dissertação é classificada como uma ferramenta exploratória. Na seção 3.4 as ferramentas exploratórias são comparadas. Por fim, na seção 3.5, são apresentadas as considerações finais.

3.1 Integração entre fontes de dados heterogêneas em BI

As pesquisas na área de *Business Intelligence* tiveram início na década de 1990 e, desde então, a área foi impulsionada pelo sucesso dos trabalhos e ferramentas OLAP, as quais demonstraram êxito na análise de dados convencionais estruturados. Recentemente, a comunidade científica tem trabalhado no sentido de integrar dados de diferentes origens e tipos em uma mesma ferramenta para explorá-los analiticamente.

Park e Song (2011) propuseram uma metodologia de arquitetura para sistemas BI que integra dados estruturados e não-estruturados. Contudo, tal integração é apenas no nível lógico, tendo em vista que os dados não-estruturados são convertidos em um esquema relacional e armazenados em um banco de dados relacional. Desta forma, a análise é realizada com base em dois *Data Warehouses* distintos. Para executar uma consulta, a arquitetura realiza uma pesquisa detalhada através de operação entre as duas tabelas de fatos (*cross-join*). Em outras palavras, há uma integração de duas fontes de dados estruturados e não a partir de fontes de dados heterogêneos diretamente.

Tao et al. (2013) construíram uma plataforma genérica, denominada *EventCube*, que permite a importação de qualquer coleção de texto não-estruturada (e.g., notícias, relatórios) em uma base estruturada. A arquitetura da plataforma proposta consiste em quatro módulos: carregamento e preparação dos dados, materialização e indexação, módulo de busca e apresentação dos resultados. No primeiro módulo, utilizando técnicas de Processamento de

Linguagem Natural e de Extração de Informação, entidades (pessoa, evento, tempo) são extraídas dos dados coletados e, com esses dados, é construído um cubo para análise multidimensional. O módulo de materialização e indexação, indexa e materializa, parcialmente, os resultados de uma provável pesquisa por palavra-chave. O módulo de busca é responsável por processar consultas realizadas pelos usuários. Por fim, há o módulo de apresentação dos resultados. Segundo os autores, a plataforma *EventCube* foi testada, em um contexto real, com dados da NASA (*National Aeronautics and Space Administration*);

Oukid et al. (2013) propuseram uma abordagem contextual *Text*-OLAP para dados não-estruturados (reportagens, e-mails, relatórios). Tais dados são extraídos utilizando técnicas de Recuperação da Informação. No DW desenvolvido, a informação contextual (perfil do usuário, localização, tempo, temperatura) é representada através de dimensões contextuais. A tabela de fatos desenvolvida contém atributos numéricos relativos à relevância (peso numérico) de um termo no texto. Com essa modelagem, o cubo desenvolvido, denominado *CXT-Cube*, possibilita uma análise considerando o contexto, por exemplo, uma mesma consulta realizada em momentos distintos, pode retornar resultados diferentes, específicos para cada momento.

Outros trabalhos focaram em incorporar dados de redes sociais, em sua maioria semiestruturados ou não estruturados, em ambientes de análise. Gallinucci et al. (2013) apresentaram uma arquitetura de *Social Business Intelligence* (SBI). Os dados são coletados da web (*Facebook, Twitter, Blogs,* Fóruns) com base em *keywords* definidas pelo usuário. Após a coleta, os dados capturados passam por um processo de estruturação e ETL, sendo armazenados em uma base temporária (semelhante a uma *Staging Area*). Os dados armazenados na base temporária passam por um processo de enriquecimento semântico e mineração de dados, sendo convertidos para uma estrutura de cubo. A arquitetura proposta pelos autores também prevê a integração, ainda na base temporária, de dados coorporativos internos.

Rehman et al. (2013) também propuseram uma abordagem de SBI. Os autores exploraram os dados do *Twitter*, os quais são capturados – com base em *keywords*, submetidos a um processo de pré-processamento (técnicas de *stemming,* correção gramatical, desambiguações, etc.), para, então, serem estruturados em um *Data Warehouse* e desenvolvido um cubo multidimensional para análise. Para validar a abordagem proposta, os autores coletaram 500 mil *tweets* referentes à Eurocopa de 2012. As palavras chaves utilizadas para captura de dados foram *#Euro2012, #Eurocup*, dentre outras. O processo de

desambiguação consistiu em definir nomes semelhantes (ITvsSP é equivalente a *Italy* versus *Spain*, por exemplo), devido à limitação de 140 caracteres.

Os trabalhos supracitados lidam com a integração de dados não estruturados em ambientes analíticos através de uma fonte estruturada - banco de dados relacional. Em outras palavras, há uma estruturação desses dados para que os mesmos estejam aptos para análise nas ferramentas analíticas. A diferença entre estes trabalhos está na forma como a estruturação dos dados é realizada (tempo real ou não) ou como o processo de ETL é executado. Ademais, todos esses trabalhos não levam em consideração o componente espacial dos dados, realizando a integração apenas com dados convencionais.

A análise de fontes de dados espaciais foi objeto de estudo de outros autores. O *Framework SOLAP*, proposto por Silva (2013), por exemplo, permite a conexão e análise de vários cubos espaciais provenientes de diversos servidores de dados multidimensionais, independente do fabricante. Por exemplo, em uma única ferramenta é possível analisar, simultaneamente, cubos oriundos do *Microsoft SQL Server Analysis Service* (SSAS) e do *Mondrian*. Ademais, Silva (2013) realiza um estudo sobre o estado da arte das ferramentas SOLAP e propõe uma extensão do formalismo VizQL (Stole e Hanrahan, 2000), com adição de uma nova divisória para consulta, a divisória "Camadas". No entanto, o *Framework SOLAP* trabalha estritamente com a integração de fontes de dados espaciais estruturadas.

Bimonte et al. (2014) propuseram uma metodologia SOLAP para a análise de dados semiestruturados voluntariamente obtidos na web. Tais dados, conhecidos como VGI - *Volunteered Geographic Information* (Goodchild, 2007), apresentam alguns problemas, a exemplo da credibilidade e qualidade dos mesmos. Com base nisso, no processo de ETC que estes são submetidos antes de serem carregados no *Spatial* DW, são calculadas e criadas duas novas medidas: uma relacionada à credibilidade da informação (*credibility-based aggregation*) e outra à qualidade dos dados (*data-quality*). Em seguida, um cubo é desenvolvido utilizando o *GeoModrian*, e com tais medidas disponíveis, é possível realizar análises de acordo com o grau de sensibilização de qualidade de dados que os usuários toleram. De acordo com os autores, a ferramenta proposta foi validada através de um cenário real de inundação disponível no *Wikimapia*.

De forma similar a Rehman et al. (2013), Hannachi et al. (2013) exploraram os dados do *Twitter.* O mesmo procedimento de pré-processamento é executado antes do carregamento dos dados em um DW. Entretanto, os autores consideram o contexto espacial do *tweet*. A informação espacial do *tweet* pode ser capturada através de três formas distintas: coordenadas

de latitude e longitude do *tweet*, cidade ou fuso horário do usuário (obtidos no perfil do usuário), dados estes que muitas vezes apresentam inconsistências. Novamente, os trabalhos de Bimonte et al. (2014) e Hannachi et al. (2013) não provêem a integração, ou análise conjunta, com outras fontes de dados, sejam espaciais ou convencionais, estruturadas ou não.

3.2 Dados Semânticos

Nos últimos anos, vários trabalhos que exploram dados semânticos vêm sendo desenvolvidos. Tais estudos exploram informações contextuais, a exemplo de dados disponíveis na Web Semântica. Nesse contexto, diversos problemas permanecem sem uma solução definitiva, como a qualidade dos dados disponíveis na Web, o gerenciamento, a exploração e a interoperabilidade com outras fontes de dados. Nesta seção, apresenta-se uma visão geral desses trabalhos, cujas ideias serviram como base para o desenvolvimento da solução proposta nesta dissertação.

Apesar da grande quantidade de dados disponíveis na Web Semântica, estes nem sempre se apresentam aptos para a exploração devido à sua qualidade. Zaveri et al. (2016) realizaram um *Survey* analisando a qualidade das bases de dados interligadas (*Linked Data*) exploradas em diversos trabalhos. Os critérios de análise do *Survey*, dentre outros aspectos, dizem respeito à disponibilidade, licenciamento, segurança, performance e relevância dos dados interligados. O objetivo do *Survey* é fornecer uma compreensão abrangente dos estudos existentes na área e incentivar pesquisadores a desenvolverem novas abordagens e metodologias (semelhantes a processos ETC) que visem melhorar a qualidade dos *Linked Data*.

Outros trabalhos abordam a questão do gerenciamento de arquivos nativos da Web Semântica. Zou et al. (2014), por exemplo, desenvolveram o *gStore*, um sistema que propõe armazenar vários arquivos RDF como um único grande grafo, possibilitando que consultas SPARQL sejam realizadas de forma escalável. Para tanto, as consultas SPARQL são representadas como um grafo de consultas, de modo que haja consultas de correspondência de subgrafos. Além disso, os autores desenvolveram índices e propuseram um algoritmo efetivo para lidar com atualização de arquivos RDF.

Semelhantemente às VQLs, que oferecem uma linguagem de alto nível para realizar consultas em cubos multidimensionais, alguns estudos semânticos avançaram nesse sentido, objetivando facilitar a formalização de consultas por usuários leigos. Clark (2010)

desenvolveu um módulo, específico para o *Drupal*[3], que permite a construção de consultas visuais em SPARQL utilizando a metodologia *drag-and-drop*. O usuário indica o SPARQL *Endpoint* a ser consultado e o sistema, automaticamente, recupera os prefixos dessa base semântica e já adiciona na consulta SPARQL *Select*. As cláusulas restantes que compõem a consulta SPARQL (*select*, *where* e modificadores) são formuladas pelo usuário de uma maneira facilitada, como por exemplo, as triplas da cláusula *where* são indicadas graficamente (utilizando nós e arestas), e não textualmente.

Haag et al. (2014) desenvolveram um sistema, denominado *SparqlFilterFlow*, que fornece uma interface visual para a composição de consultas SPARQL do tipo *Select* ou *Ask*. A formulação das consultas visuais SPARQL é baseada no modelo de filtro/fluxo (Seitz e Baker, 2009): o usuário, por meio da metodologia de *drag-and-drop*, constrói a consulta conectando nós e filtros a um fluxo. De acordo com os autores, o sistema proposto pode se comunicar com qualquer SPARQL *Endpoint*, a ser indicado pelo usuário.

O estudo de Bikakis et al. (2015) trata da interoperabilidade de dados da Web Semântica com o formato XML. Os autores propuseram um framework, denominado de *SPARQL2XQuery*, que cria um ambiente interoperável entre Web Semântica e XML, o que permite realizar consultas SPARQL em ambientes XML. As consultas SPARQL são automaticamente traduzidas para consultas XQuery, a fim de consultar dados XML na Web. O mapeamento entre esses dois mundos pode ser realizado de forma automática ou manual. No primeiro caso, o framework *SPARQL2XQuery* transforma esquemas XML em ontologias OWL. No caso de mapeamento manual, é fornecido um modelo de mapeamento, expresso em OWL-RDF, para que o usuário identifique os valores correspondentes. Experimentos foram realizados, contabilizando o tempo de transformação de esquemas, geração de mapeamento e tradução de consultas. Contudo, a despeito de realizar a integração entre duas fontes de dados distintas, os autores trabalham estritamente com um único tipo de estruturação, os arquivos semiestruturados, e também não levam em consideração o componente espacial dos dados.

3.2.1 Dados Semânticos Espaciais

Assim como grande parte dos dados estruturados vem incorporada de um componente espacial, os dados disponíveis na Web Semântica também apresentam essa característica. Logo, vários estudos que tem por objeto a exploração espacial dos dados da Web Semântica vêm sendo publicados.

[3] https://www.drupal.org/

Wang et al. (2012) preenchem a lacuna de armazenamento de arquivos semânticos espaciais ao proporem o *Geo-Store,* um sistema que gerencia arquivos RDF com triplas espaciais. Entretanto, o gerenciador proposto dá suporte apenas a tipos geométricos do tipo ponto (caracterizados por uma única tripla com predicado correspondente), permitindo, apenas consultas espaciais do tipo *bounding-box queries*.

Zhai et al. (2010) desenvolveram uma extensão da linguagem SPARQL a fim de oferecer consultas espaciais. Para isso, eles criaram uma ontologia com a definição de um vocabulário que represente características geométricas, suas relações e funções espaciais. Porém, o trabalho dos autores é limitado na medida em que só define funções espaciais topológicas e oferece apenas consultas SPARQL tipo *Select*, ou seja, não é possível inserir dados espaciais, apenas consultar.

De forma semelhante, Battle e Kolas (2012) também desenvolveram uma extensão espacial para a linguagem SPARQL, denominada *GeoSPARQL*. A ontologia desenvolvida apresenta um vocabulário mais rico, contemplando várias características geométricas e oferecendo tanto funções espaciais topológicas quanto funções *equals* e *disjoint* entre objetos espaciais. Além de permitir consultas espaciais, a linguagem *GeoSPARQL* também oferece operações do tipo *Construct*.

Apesar de vários trabalhos propuserem uma extensão espacial para a linguagem SPARQL, ainda não há consenso sobre como tratar o componente espacial nas consultas SPARQL e não há padronização de nenhuma dessas extensões pela W3C. Atualmente, a W3C mantém um grupo de discussão sobre o assunto, que culminou com o desenvolvimento informal do vocabulário *Basic Geo* (WGS84 lat/long)[4]. O status deste documento ainda está em andamento e necessita de uma revisão associada para que seja normalizada e padronizada pelo consórcio.

Não obstante, dentre todas as extensões espaciais propostas, a extensão *GeoSPARQL* é a mais utilizada devido a sua completude. O *Framework Apache Jena*, por exemplo, faz uso desta extensão para prover consultas espaciais utilizando a linguagem SPARQL[5]. Wiegand et al. (2015), por outro lado, desenvolveram uma ferramenta *online* – denominada *GeoQuery* - que visa facilitar a construção de consultas espaciais utilizando *GeoSPARQL.* Para tanto, utiliza de componentes gráficos para formulação de consultas, excluindo a necessidade do usuário ter pleno domínio da linguagem. Os resultados da consulta são exibidos em um mapa.

[4] https://www.w3.org/2003/01/geo/
[5] https://jena.apache.org/documentation/query/spatial-query.html

A integração de dados espaciais interligados foi objeto de estudo de Both et al. (2015). Os autores desenvolveram uma interface *web*, denominada *GeoKnow Generator Workbench*, que integra todos os componentes necessários para o processamento do ciclo de processos interligados, quais sejam: exploração, extração, armazenamento e consulta, autoria, fusão, classificação e enriquecimento, análise e evolução. Com base nesse ciclo de processos, a ferramenta proposta apresenta facilidades na captação e integração de dados interligados espaciais. Não há integração com outras fontes de dados de diferentes tipos de estruturação.

3.3 Data Warehouses Semânticos

O crescimento e a disponibilidade dos dados da Web Semântica aguçaram o interesse em explorar tais dados em um ambiente analítico. Berlanga et al. (2012) argumentam que a convergência entre as tecnologias de *Business Intelligence* e Web Semântica é essencial num contexto cada vez mais competitivo, no qual as fontes de dados primárias para ferramentas analíticas tradicionais (fonte de dados estruturados relacionais) necessitam de enriquecimento semântico a partir da incorporação de informação externa – encontrada na Web em outras formas de estruturação - para tornar a análise mais robusta.

Niinimäki e Niemi (2009) apresentaram uma metodologia ETC para construção sob demanda de cubos multidimensionais com armazenamento seguindo a abordagem ROLAP. O processo ETC faz uso de uma ontologia que serve como base para a concepção e criação do DW. O vocabulário possui definições a respeito das tabelas correspondentes, facilitando, assim, a incorporação de novas fontes de dados nas ferramentas analíticas. Em seguida, os dados são extraídos a partir de consultas SPARQL e armazenados no DW.

Romero e Abelló (2010) apresentaram um *framework* para modelar DW a partir de ontologias. A proposta dos autores é oferecer um sistema que analisa as fontes de dados, identificando as mais propícias – de acordo com modelo de negócio - para então formalizar um DW. Os autores, porém, trabalham apenas no nível conceitual, não havendo criação lógica e física do DW em nenhum banco de dados. Nebot e Berlanga (2010) e Kämpgen e Harth (2011) propuseram abordagens semiautomáticas para construção de DW semântico, visto que há uma definição e validação do usuário para o esquema e dados extraídos. Estes trabalhos se diferenciam quanto à fonte de dados semântica primária para extração dos dados, porque, enquanto o primeiro extrai dados de ontologias OWL, o segundo explora *Linked Data*, em sua maioria no formato RDF.

Neumayr et al. (2012) se diferenciaram ao formalizar uma ontologia multidimensional (MDO – *Multidimensional Ontology)*, caracterizada por um DW enriquecido a partir da

extração de conceitos e relacionamentos relevantes presentes em ontologias. A descoberta da ontologia, a identificação dos termos relevantes e o mapeamento entre as fontes de dados é de responsabilidade do usuário, de modo que a representação MDO pode ser definida como estática e manual. Após a construção do DW semanticamente enriquecido, o mesmo pode ser consultado utilizando SQL ou MDX, caso seja explorado em uma ferramenta OLAP.

Os estudos citados até então integram parcialmente tecnologias BI com Web Semântica. A integração é limitada em carregar dados semânticos em DW para explorá-los em ferramentas analíticas, típicas de BI. Outrossim, nenhum desses trabalhos tratam do componente espacial presente na Web Semântica.

Outros estudos focaram em fazer o intercâmbio da modelagem multidimensional para o mundo da Web Semântica. Neumayr et al. (2013) aperfeiçoaram seu trabalho anterior efetuando a conversão do DW semanticamente enriquecido para ontologias OWL. Prat et al. (2012) também exploraram essa lacuna e apresentaram uma abordagem para representação multidimensional como uma ontologia OWL-DL. A linguagem *Description Logic* é utilizada para realizar a verificação e sumarização do modelo.

A representação multidimensional em ontologias tem suas vantagens, a citar a facilidade e possibilidade de integração com outros domínios de conhecimento, assim como o compartilhamento e reuso da informação entre pessoas e máquinas. Não obstante, ao converter o mundo multidimensional em ontologias, perde-se a facilidade de análise oferecido pelas ferramentas analíticas tradicionais. Nessa perspectiva, Kämpagen et al. (2012) propuseram uma maneira de ferramentas OLAP interagirem com *Linked Data.* Os autores definiram operações OLAP comuns (*slice*, *dice*, *roll-up* e *projection*) sobre cubos de dados modelados em RDF e desenvolveram um compilador que traduz operações OLAP em linguagem MDX para consultas SPARQL.

Etcheverry e Vaisman (2012) apresentaram um novo vocabulário, denominado QB4OLAP, que, além de representar cubos OLAP em RDF, implementa operadores OLAP (*roll-up*, *slice*, *dice*, etc.) em SPARQL, permitindo consultá-los diretamente nas triplas. Em um trabalho posterior, Etcheverry et al. (2014) adicionam mais recursos ao vocabulário QB4OLAP, a exemplo da representação de hierarquias multidimensionais (1:N e N:M). Outrossim, os autores apresentam um conjunto de regras para conversão de um modelo multidimensional para RDF utilizando QB4OLAP e demonstram exemplos de consultas SPARQL, utilizando operadores OLAP, no cubo RDF.

Gür et al. (2015) exploraram o vocabulário QB4OLAP expandindo-o com a adição de conceitos espaciais, com a definição de tipos de dados espaciais, operadores espaciais e relacionamentos topológicos, possibilitando tanto a representação quanto a análise espacial em dados semânticos armazenados em arquivos RDF. Esta extensão espacial foi chamada de QB4SOLAP[6]. Os autores validaram sua proposta através de um estudo de caso utilizando o contexto do banco de dados GeoNorthwind; entretanto, não há nenhuma integração espacial entre fontes de dados heterogêneas.

3.3.1 Ferramentas Exploratórias

É sabido que a integração entre dados da Web Semântica, juntamente com dados estruturados, nas ferramentas analíticas não é uma tarefa trivial. Berlanga et al. (2012) citam os desafios encontrados neste âmbito. Segundo os autores, o primeiro desafio está relacionado à representação da semântica no domínio analítico, que torne possível sua exploração pelas ferramentas e pelos tomadores de decisão. O segundo desafio diz respeito à integração, com consonância, das fontes de informações heterogêneas tão díspares. Nesse contexto, emerge uma nova área de pesquisa e, consequentemente, um novo tipo de ferramenta, definida como *Exploratory OLAP* (Abelló et al., 2015).

Ibragimov et al. (2015) propuseram uma ferramenta exploratória conceitual que utiliza um esquema multidimensional OLAP, expresso em vocabulários RDF, na integração de fontes de dados. A proposta vislumbra consultar fontes de dados externas automaticamente, armazenadas em formato semiestruturado – HTML, XML, CSV, RDF. A descoberta das fontes externas para integração dos dados pode ser realizada utilizando três metodologias distintas: consultando bases de conhecimento (*DBPedia*, *Yago*, *Freebase*) via SPARQL *Endpoint*; plataforma de dados (CKAN) ou mecanismos de buscas online. Nesses últimos dois casos, a consulta dá-se através da utilização de APIs específicas.

A partir dos dados capturados é construído um cubo virtual (não há materialização dos dados) para então os mesmos serem explorados. A integração das fontes de dados dá-se sob demanda: quando o usuário requer uma nova informação, o sistema formula a consulta na linguagem correspondente, por exemplo SPARQL, e esta é executada na base de dados integrada. Na proposta conceitual da ferramenta exploratória, não foi detalhado o nível de transformações realizadas nos dados integrados. A ferramenta conceitual proposta pelos autores apresenta algumas limitações. Inicialmente, a ferramenta é conceitual, portanto não foi implementada nem validada, não podendo ser reproduzida. Também, não há integração de

[6] http://extbi.cs.aau.dk/QB4SOLAP/data/qb4solap.ttl

diferentes fontes de dados - estruturados, semiestruturados ou não estruturados. Outrossim, a mesma trabalha apenas com dados convencionais. É uma ferramenta exploratória OLAP, não sendo possível manipular dados espaciais.

Furtado et al. (2015) apresentam uma ferramenta exploratória focada na capacidade do DW de monitorar, atualizar, e agregar novos dados continuamente. O estudo é concentrado na detecção e geração automática de *baselines* (métrica estatística sobre a distribuição dos dados) e de KPIs (*Key Perfomance Indicator*). Ao detectar um indicador a respeito dos dados estruturados originais, o mesmo é calculado e materializado em uma visão para, então, ser incorporado, automaticamente (sob-demanda), na ferramenta OLAP para análise, juntamente com os dados originais. As transformações necessárias na geração de *baselines* e de KPIs são simples, geralmente apenas cálculo de uma fórmula ou sumarizações. Por exemplo, quando há dados fora da normalidade (*baselines*) ou KPIs são identificados, um alerta é informado ao usuário, sendo materializados como uma visão no DW. A ferramenta proposta não apresenta mecanismos que facilite a integração com novas fontes de dados. Também, dá suporte apenas a dados estruturados. É uma ferramenta exploratória OLAP, não sendo possível manipular dados espaciais.

3.4 Comparativo dos Trabalhos Relacionados

Conforme discussão na seção 2.3, Abelló et al. (2015) caracterizaram as ferramentas exploratórias com base em cinco critérios: materialização, transformações, frequência de integração, estruturação e extensibilidade. A partir dessa discussão, realizou-se uma avaliação comparativa entre os dois trabalhos científicos encontrados na área. Os trabalhos foram avaliados de acordo com o nível de cada um desses critérios. O resultado da comparação entre os trabalhos é apresentado no Quadro 1.

A solução proposta por Ibragimov et al. (2015) se diferencia ao virtualizar as fontes de dados e ao oferecer mecanismos para descoberta e integração de novas fontes de dados, critério não abordado por Furtado et al. (2015). Nesse sentido, a solução de Furtado et al. (2015) se aproxima mais de uma ferramenta analítica tradicional. Por outro lado, por materializar KPIs em visões, o nível de transformações a serem realizadas são mais simples, necessitando apenas de sumarizações; e, com base nos critérios de Abelló et al. (2015), pode ser considerada exploratória.

Quadro 1: Comparativo entre as ferramentas exploratórias investigadas

	Ibragimov et al. (2015)	**Furtado et al. (2015)**
Materialização	Virtual	Total
Transformações	------	Leve
Frequência de Integração	Sob Demanda	Sob Demanda
Estruturação	Dados semiestruturados (semânticos e não semânticos)	Dados estruturados (Não semânticos)
Extensibilidade	Adaptação	-----

3.5 Considerações Finais

Este capítulo apresentou os principais trabalhos relacionados com a proposta desta dissertação. Considerando tais estudos, promoveu-se uma discussão sobre a importância da integração entre as áreas de Web Semântica e *Business Intelligence*. Essa integração está permeada de desafios, impulsionando o surgimento das ferramentas exploratórias – *Exploratory* OLAP. Em meio aos poucos estudos científicos publicados que propuseram ferramentas exploratórias e com base nos critérios propostos por Abelló et al. (2015), foi realizada uma avaliação comparativa entre as soluções propostas nesses estudos e ficou evidente que nenhuma solução proposta promoveu avanços no critério de estruturação, visto que todas elas só trabalham com dados estruturados. Ademais, nenhum dos trabalhos avaliados explora a dimensionalidade espacial dos dad
os, promovendo o *Exploratory* SOLAP.

No próximo capítulo, será descrito em detalhes a solução exploratória espacial proposta neste trabalho.

Capítulo 4 – ExpSOLAP: *Exploratory SOLAP System*

Neste capítulo, a solução ExpSOLAP - *Exploratory* SOLAP, que integra dados espaciais semiestruturados semânticos com fontes tradicionais de dados espaciais estruturados, é apresentada. As fontes de dados integradas à solução ExpSOLAP são caracterizadas, respectivamente, por uma ontologia representada no formato RDF e por um cubo multidimensional armazenado seguindo a abordagem MOLAP.

A solução ExpSOLAP foi desenvolvida como uma extensão do Framework SOLAP (Silva, 2013), acrescida de um módulo de gerenciamento de fontes de dados semânticos. Este módulo é responsável pela incorporação de dados semânticos semiestruturados à análise, possibilitando a exploração espacial, simultânea, nos dois tipos de fontes de dados espaciais heterogêneas integradas: dados semiestruturados e dados estruturados.

O restante deste capítulo é organizado como segue. Na seção 4.1 é proposta uma nova categorização de ferramentas exploratórias considerando o critério de dimensionalidade dos dados. Em seguida, na seção 4.2, é descrita a solução ExpSOLAP sob aspectos de projeto. Na seção 4.3, a solução ExpSOLAP é descrita sob aspectos tecnológicos. Na seção 4.4, é ilustrado um diagrama de atividades exemplificando o funcionamento da solução, desde a consulta até a exibição dos resultados oriundos de ambas as fontes de dados. Por fim, na seção 4.5, são expostas as considerações finais.

4.1 Novo critério de categorização: Dimensionalidade

Um ponto de discussão sobre os critérios definidos por Abelló et al. (2015) é que estes foram baseados em ferramentas OLAP, e não em ferramentas SOLAP. Com base nisso, estendemos a categorização de Abelló et al. (2015) adicionando uma sexta vertente para categorização: dimensionalidade. Tal critério diz respeito ao perfil dos dados integrados nas ferramentas e possui quatro níveis: convencional, espacial, temporal e espaço-temporal. Os níveis do referido critério são aditivos.

Os dados convencionais são aqueles presentes nas ferramentas OLAP tradicionais. Apesar destes possuírem um histórico temporal da informação que auxiliam a tomada de decisão para um ponto futuro, não há gerenciamento temporal controlado pela ferramenta. Jensen e Snodgrass (1999) argumenta que há outras características pertinentes aos dados

temporais, a exemplo da ordem do tempo (linear, ramificada, circular) e rótulo temporal (instante, intervalo e elemento temporal). Esse detalhe da informação temporal não é considerado em ferramentas analíticas tradicionais.

Os dados espaciais são aqueles que representam informações sobre local físico e a forma de objetos geométricos, podendo ser pontos, polígonos ou linhas. Esse tipo de dado é peculiar para banco de dados espaciais (Shekhar e Chawla, 2003). Tais banco de dados oferecem um conjunto de funções espaciais que lidam com características peculiares dos dados espaciais, a exemplo da área, perímetro, centroide, entre outras. Os dados espaciais são comuns em ferramentas SOLAP tradicionais.

Por fim, o nível mais externo exploratório é caracterizados pelos dados espaço-temporais, aqueles que permitem o registro da informação considerando a mudança de posição de um objeto móvel no espaço durante um dado intervalo de tempo (Wolfson et al., 1998). Tais dados são utilizados em aplicações de trajetória. Essa nova categorização de ferramentas exploratória comprova a questão de pesquisa Q1.

Diante deste cenário, este trabalho de dissertação apresenta uma ferramenta exploratória espacial, denominada ExpSOLAP, que integra dados estruturados e semiestruturados considerando o componente espacial dos dados. Os dados oriundos das fontes de dados são integrados em *batches* e submetidos a simples transformações. Além disso, a ferramenta apresenta facilidades para integração de novas fontes de dados. Os dados estruturados serão materializados e convertidos em um DW, enquanto os dados semânticos serão persistidos (materializados) em disco.

Na Figura 6, são apresentados os critérios para categorização das ferramentas exploratórias. A área destacada na imagem representa a posição da ExpSOLAP com base nos critérios exploratórios.

No Quadro 2 é apresentada um novo comparativo entre as ferramentas analíticas, considerando os critérios de categorização propostos neste trabalho de dissertação. Como é possível observar, cada trabalho explorou um critério em específico. A solução proposta nesta dissertação, ao incorporar dados espaciais semiestruturados, priorizou o critério de estruturação dos dados e da dimensionalidade. Ao enriquecer dados internos através de dados externos semiestruturados espaciais, o tomador de decisão terá seu domínio de análise expandido, possibilitando uma análise mais profunda. A incorporação de dados semiestruturados, juntamente com dados estruturados, à análise não foi explorada em nenhum outro estudo.

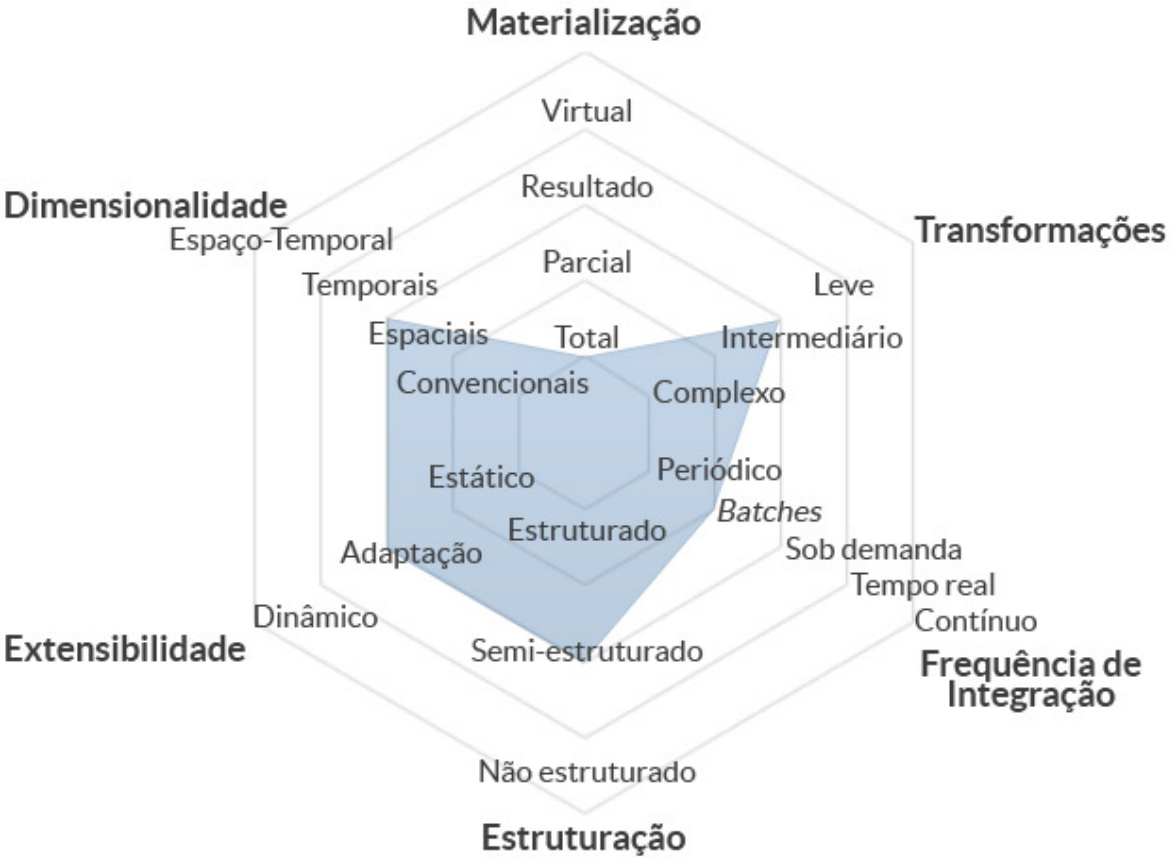

Figura 6: Nova categorização de ferramentas analíticas, destacando a posição da ferramenta ExpSOLAP. Adaptado de Abelló et al. (2015)

Quadro 2: Novo comparativo entre ferramentas exploratórias

	Ibragimov et al. (2015)	**Furtado et al. (2015)**	**ExpSOLAP**
Materialização	Virtual	Total	Total
Transformações	----	Leve	Intermediário
Frequência de Integração	Sob Demanda	Sob Demanda	*Batches*
Estruturação	Dados semiestruturados (semânticos e não-semânticos)	Dados estruturados (Não-semânticos)	Dados Estruturados + Semiestruturados (Semânticos)
Extensibilidade	Adaptação	------	Adaptação
Dimensionalidade	Convencionais	Convencionais	Espaciais

4.2 ExpSOLAP: Aspectos de projeto

A solução ExpSOLAP foi desenvolvida como extensão do Framework SOLAP (Silva, 2013), cujo principal objetivo é realizar a análise espacial dos dados provenientes de várias fontes de dados espaciais multidimensionais estruturadas. Ademais, também apresenta como característica permitir a criação de consultas por meio de uma linguagem de especificação visual; e possibilitar a geocodificação dos dados, possibilitando a análise espacial em fontes puramente convencionais.

A análise espacial em servidores puramente OLAP é garantida pela geocodificação (processo de associar coordenadas geográficas a endereços). Ao geocodificar membros de dimensão, é possível realizar a análise espacial em cubos OLAP sem que seja necessário o retrabalho de implementação desses cubos com o acréscimo dos componentes espaciais utilizando tecnologias SOLAP. Nesse contexto, o Framework SOLAP segue abordagem federada, ou seja, os dados espaciais são armazenados em um repositório diferente dos dados multidimensionais, cujo acesso é realizado através do driver JDBC.

Para realizar a análise de fontes de dados multidimensionais é necessário importar, através do protocolo XMLA, a estrutura do cubo multidimensional. Os metadados da fonte multidimensional (dimensões, níveis e medidas) serão recuperados e estarão disponíveis para o usuário manipulá-lo e formular consultas utilizando uma linguagem de consulta visual (VQL). A consulta VQL formulada será convertida para a linguagem nativa de servidores multidimensionais, para então consultar a base multidimensional. Por fim, os resultados são convertidos e exibidos ao usuário. Na Figura 7, é ilustrado um Diagrama de Caso de Uso (Miles e Hamilton, 2006), simplificado, indicando as principais atividades relacionadas ao cadastro e análise de fontes de dados multidimensionais. No caso da formulação de consultas com restrições espaciais, a consulta é executada inicialmente no repositório de dados da geocodificação, para então ser convertida para linguagem nativa, considerando restrições formuladas com os resultados obtidos, e executada na base multidimensional.

O principal ponto de extensão dessa solução quanto ao Framework SOLAP é a adição de uma fonte de dados semântica, externa à análise. A nova fonte de dados incorporada à solução deve ter casos de uso semelhantes às atividades das fontes de dados multidimensionais já incorporadas no Framework SOLAP (Figura 7). Na Figura 8, é ilustrado a o Diagrama de Casos de Uso indicando as atividades inerentes à adição de uma fonte de dados semânticos. Ressalta-se que as atividades são praticamente idênticas ao cadastro de uma fonte multidimensional, diferenciando que o cadastro de uma fonte de dados semânticos além de

incorporar o cadastro de metadados da fonte de dados semânticos (caracterizados por prefixos/*namespaces*, localização dos arquivos semânticos), também engloba o mapeamento entre as fontes de dados heterogêneas. O mapeamento entre as fontes de dados heterogêneas trata da ligação entre os metadados do cubo multidimensional com os metadados da fonte de dados semântica. Este mapeamento é realizado de forma manual.

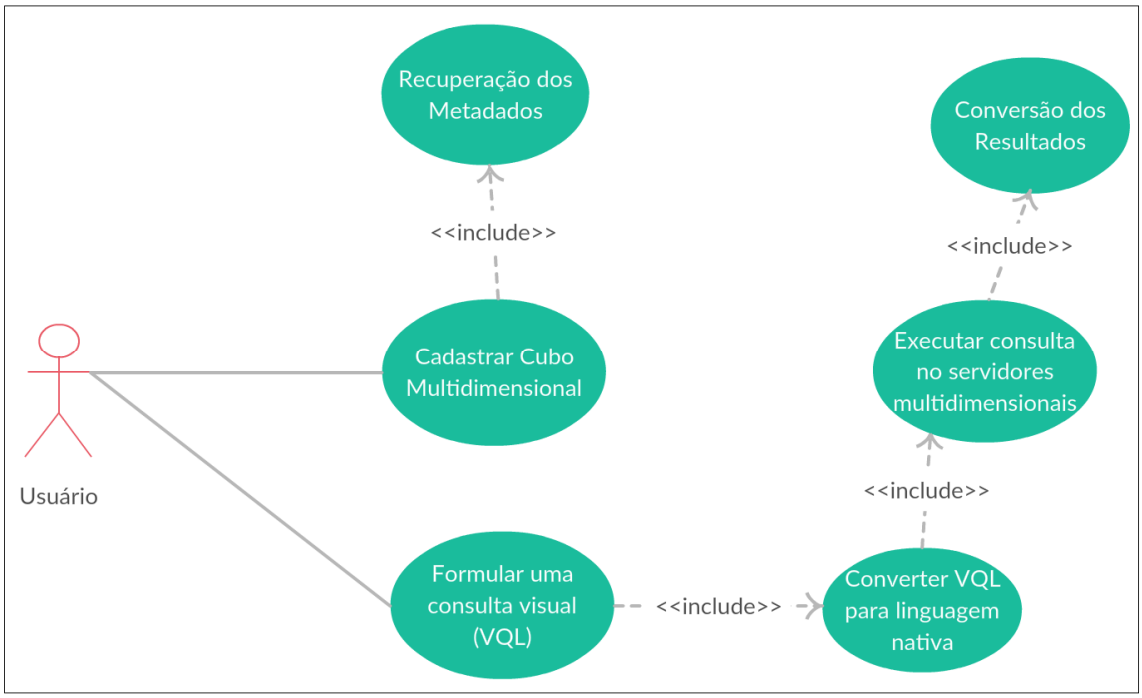

Figura 7: Diagrama de Casos de Uso para cadastro de uma fonte de dados multidimensional

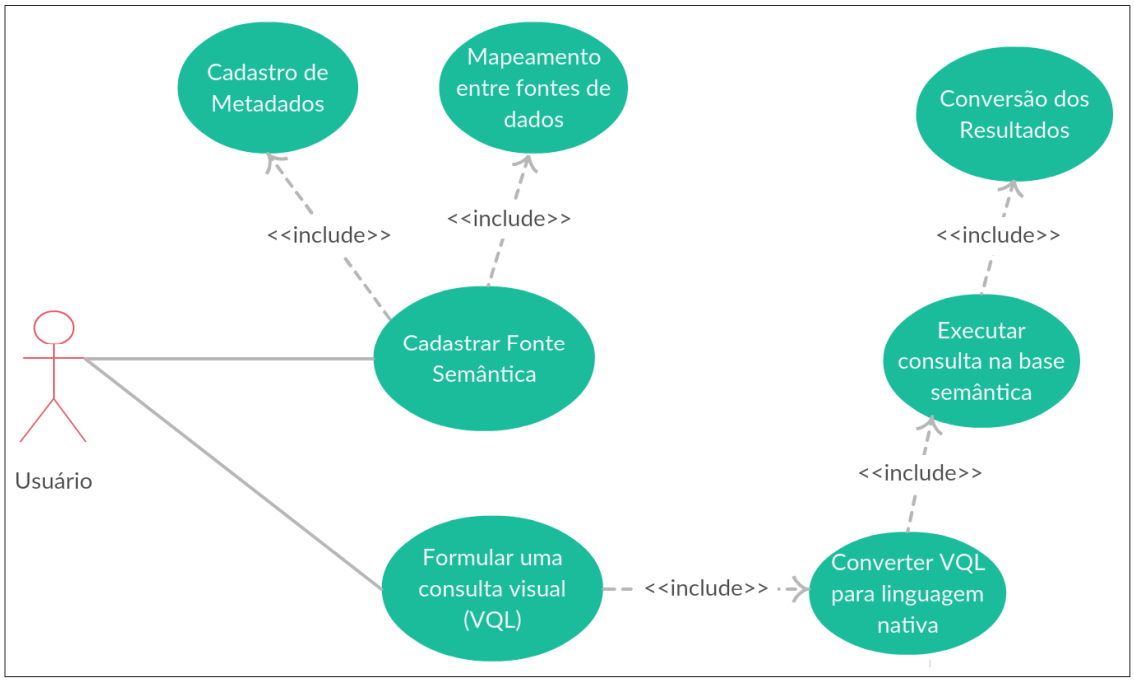

Figura 8: Diagrama de Casos de Uso para fontes semânticas

Os casos de usos referentes tanto às fontes multidimensionais quanto à fonte semântica foram modularizados na arquitetura da solução. A arquitetura da solução é caracterizada como uma arquitetura MVC (*Model-view-controller*) composta de três camadas: a camada cliente, a camada de aplicação e a camada de dados.

Essa modelagem apresenta algumas vantagens. A principal delas é a separação entre a lógica da aplicação e a camada cliente, o que resulta, consequentemente, na diminuição do acoplamento e permite a substituição ou adição, com facilidade, de algum componente isoladamente. Outra vantagem da arquitetura em camadas é a simplicidade do funcionamento, na medida em que os usuários interagem com a camada cliente, que transfere as requisições para a camada de aplicação – responsável pela lógica da solução – e esta solicita à camada de dados os elementos necessários para processar as requisições e retorná-los à camada cliente para que sejam exibidos ao usuário.

Na Figura 9, é apresentada a arquitetura da solução ExpSOLAP utilizando um Diagrama de Componentes UML (Miles e Hamilton, 2006). Os módulos destacados em cinza foram os módulos adicionados na solução ExpSOLAP.

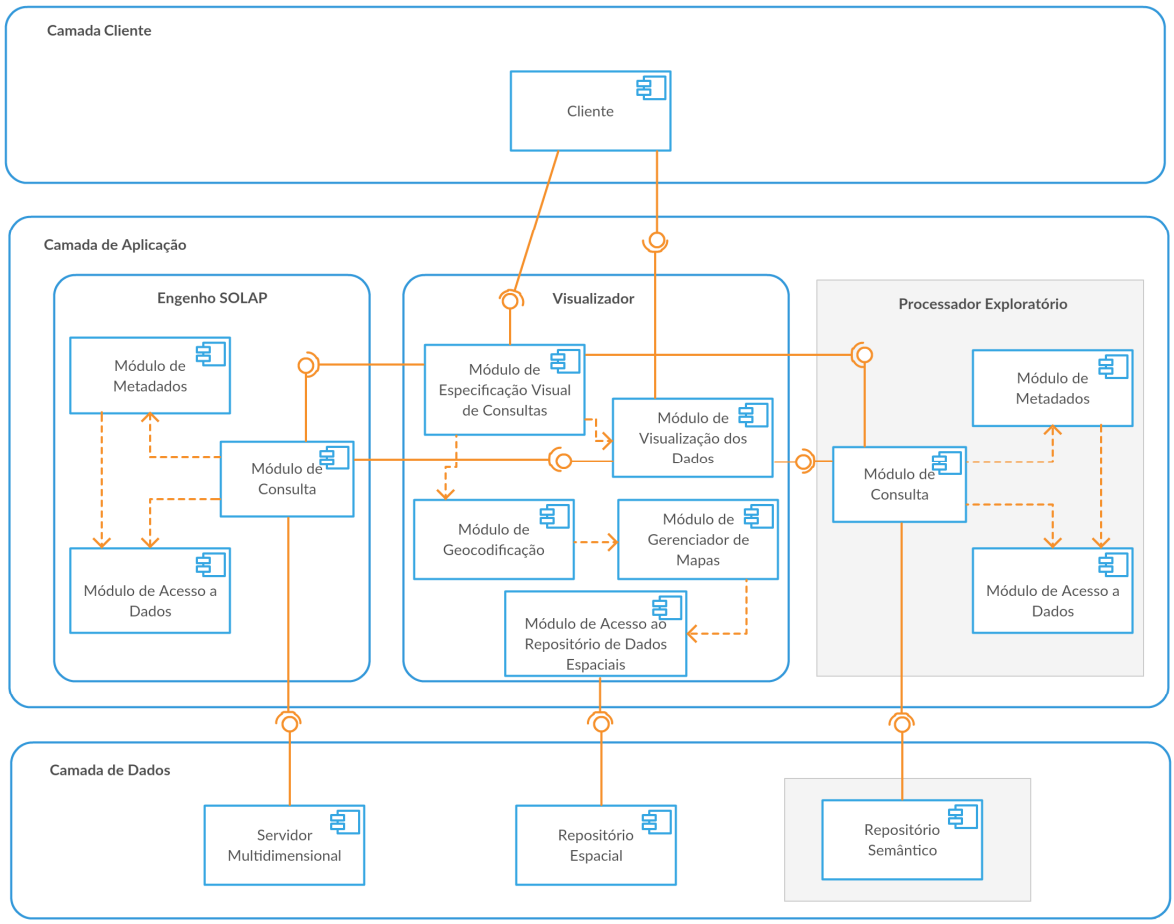

Figura 9: Diagrama de Componentes representando a arquitetura da solução ExpSOLAP. Adaptado de Silva (2013)

A Camada Cliente compreende um conjunto de componentes gráficos Web com os quais o usuário pode formular consultas e analisar/visualizar os dados através de gráficos, relatórios ou mapas. Ademais, permite a interface visual para conexão com fontes de dados multidimensionais e semânticas, bem como a geocodificação de membros de dimensões. A consulta visual formulada na Camada Cliente é repassada para a Camada de Aplicação, especificamente o módulo de Especificação Visual de Consultas. Este último módulo irá transformar a consulta VQL em uma instância do objeto *VisualQuery* (Figura 10). A consulta visual é composta de divisórias, que serão chamadas de eixos (*Axis*) nos objetos. Os eixos são compostos de hierarquias. Cada nível da hierarquia está associado a um estado de nível (*LevelState*), que mantém informações sobre quais membros estão selecionados, se o nível está sendo usado e se está detalhado (*drilled*).

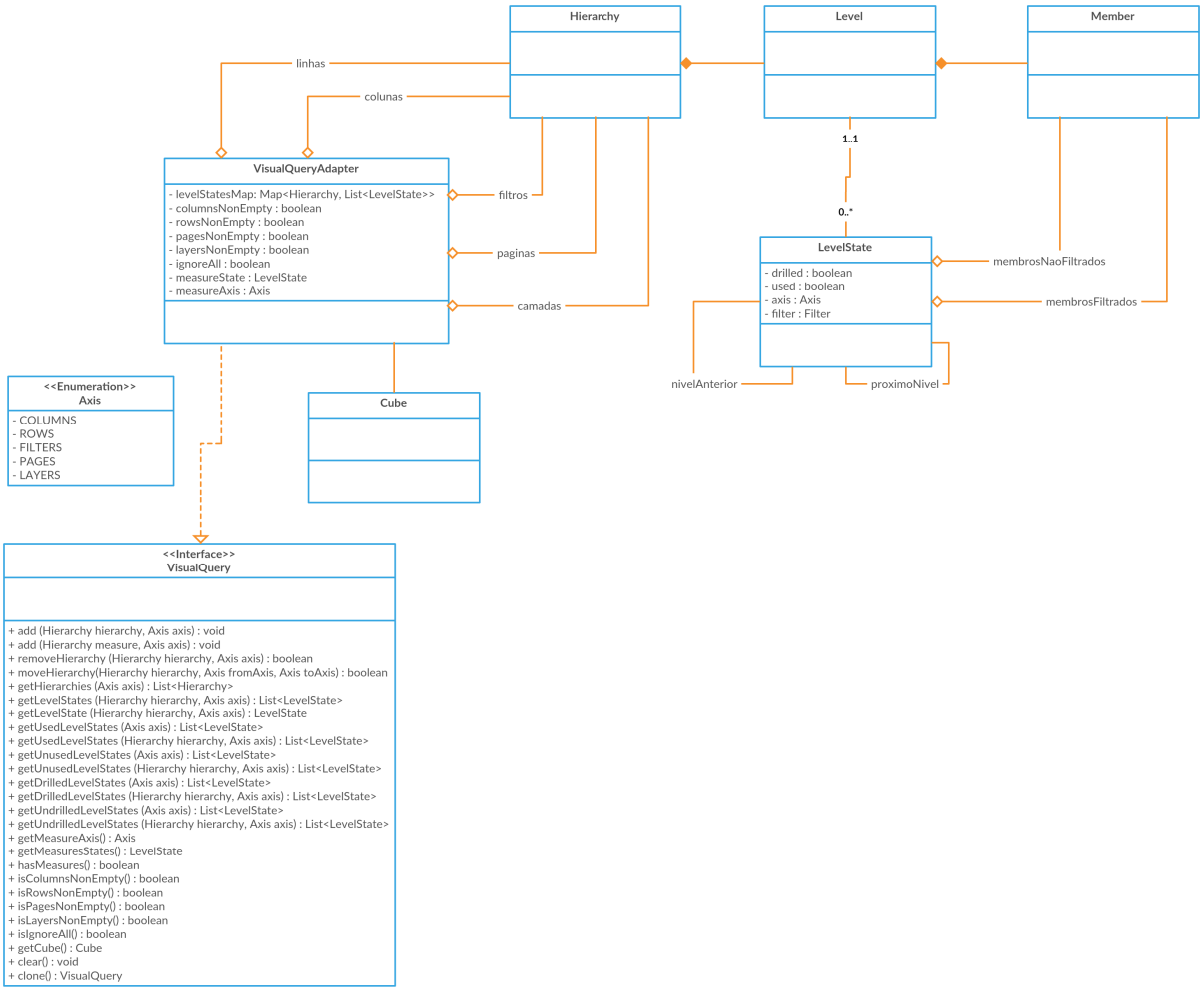

Figura 10: Diagrama de Classe para consulta visual – *VisualQuery*. Traduzido de Silva (2013)

Já na Camada de Aplicação, uma instância do objeto *VisualQuery* será repassada para os módulos de consulta do Engenho SOLAP e do Processador Exploratório para que estes convertam para linguagem nativa (MDX e SPARQL, respectivamente), consulte na base de

dados específica e recupere os resultados. Para realizar a conversão da VQL para linguagem nativa, os módulos de Consulta contam com o auxílio dos demais módulos do pacote (módulo de Metadados e módulo de Acesso a Dados). Na execução das consultas, os módulos de Consulta acessam a Camada de Dados, Servidor Multidimensional e Repositório Semântico, respectivamente, que oferecerem mecanismos para execução das consultas na base específica.

Para que uma consulta seja executada no Engenho SOLAP e no Processador Exploratório de maneira correspondente, as fontes de dados acessadas por esses dois módulos precisam estar mapeadas. O mapeamento realizado será persistido no banco de dados seguindo o esquema ilustrado na Figura 11. O mapeamento (*RDF_Cubo*_Mapeamento) mantém uma referência tanto para o cubo quanto para uma fonte semântica (*Fonte_Dados_RDF*), que possui uma lista de prefixos, relacionados aos *namespaces* da base semântica. A entidade *RDF_Cubo*_Mapeamento também possuem três listas referentes a ligação entre os metadados das fontes heterogêneas: lista do mapeamento de dimensões (*Dimensoes_Predicado*), medidas (*Medidas_Predicado*) e níveis (*Nivel_Predicado*). Estas entidades mantêm registro para o mapeamento das fontes heterogêneas, o *alias* relacionado ao metadado do cubo (nome da dimensão, nível ou medida) e ao predicado (*RDF_Predicado*). Este último está relacionado aos atributos da fonte semântica importada através dos *namespaces*. Outrossim, o mapeamento também será convertido em objetos, sendo carregado em memória para ser utilizado posteriormente pelo módulo de consulta, que recuperará tais informações no momento da conversão da consulta VQL para consulta SPARQL.

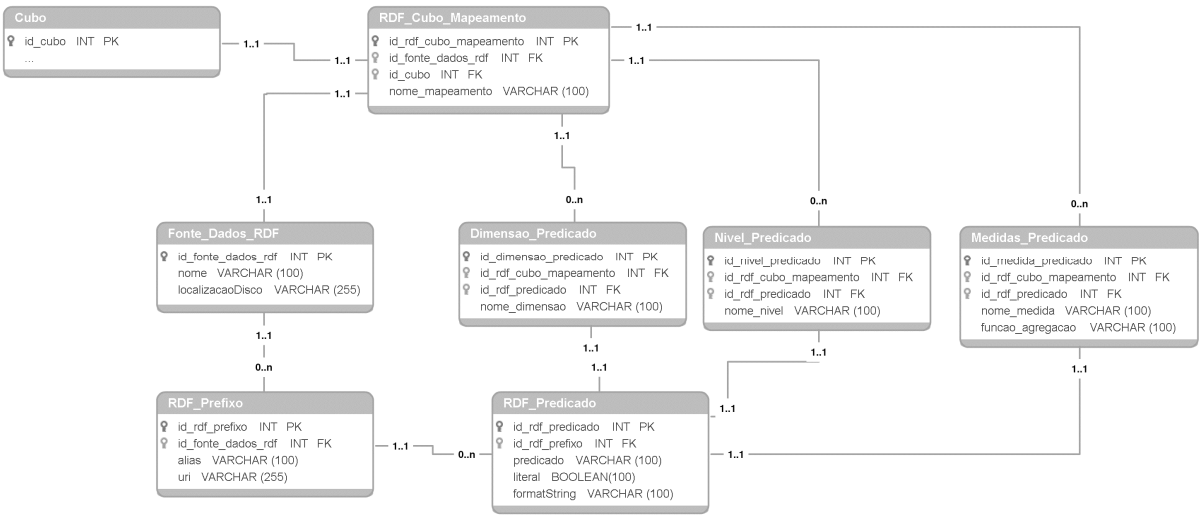

Figura 11: Diagrama Entidade-Relacionamento para armazenar o mapeamento entre as fontes heterogêneas

Todas as atividades relativas ao modulo de Consulta do Engenho SOLAP são realizadas com o auxílio da API do olap4j[7], que provê a comunicação entre a solução e o servidor OLAP utilizando o protocolo XMLA, tendo seu funcionamento similar ao protocolo JDBC. No caso do Processador Exploratório, toda lógica das atividades do módulo de Consulta foi desenvolvida, em especial o desenvolvimento de um tradutor, que converte a consulta VQL pra SPARQL. A execução das consultas na fonte semântica é realizada com auxílio do *Framework Apache Jena*. O resultado da tradução, uma consulta SPARQL do tipo *Select* compilável, será mantida em memória representada como uma instância do objeto *Sparql*.

O objeto *Sparql* representa uma consulta SPARQL e foi modelado com base nas cláusulas que compõem uma consulta SPARQL do tipo *Select*: prefixos, variáveis contidas na cláusula SELECT, cláusula WHERE e modificadores. Uma consulta SPARQL pode ter vários prefixos associados (*RDFPrefix*), caracterizados por um *alias* e um *URI* correspondente. Pelo menos uma variável (*Variable*) deve estar contida na cláusula SELECT, especificadas por um identificador, nome único, uma indicação se é medida agregadora com sua respectiva função de agregação (*SUM*, *AVG* ou *COUNT*) e o atributo *eixo* relativo à divisória deste valor na consulta visual. A cláusula WHERE é opcional numa consulta SPARQL e foi modelada através do objeto *WhereClause*, sendo composta por uma lista de triplas (*Triple*), correspondente às triplas dos padrões de grafo, e por filtros (*Filter*), compostos pela expressão e por indicação se é texto, decimal ou inteiro. Por fim os modificadores, também opcional, que são compostos por uma lista de nome de variáveis para ordenar ou agrupar e foram modelados através da classe *SolutionModifier*. Ademais, o objeto *Sparql* contém método para realizar a tradução da consulta VQL para SPARQL. O diagrama de classes para o objeto *Sparql* é ilustrado na Figura 12.

Caso a consulta espacial seja formulada, o módulo de Especificação Visual acessa os módulos de geocodificação, gerenciados de mapas e de acesso ao repositório espacial. Este último módulo se comunica com o Repositório Espacial, da Camada de Dados, para recuperar as características espaciais (coordenadas) relativas à consulta formulada.

Após a recuperação dos resultados das consultas, os módulos de Consulta do Engenho SOLAP e do Processador Exploratório irão repassar os resultados para o módulo de Visualização (módulo de Visualização de Dados). Este se encarregará de realizar o *merge* dos resultados – oriundos das duas fontes de dados – e de aplicar ou não a visualização espacial, convertendo-o para o objeto *VisualQueryResult* (Figura 13). O *merge* dos resultados está

[7] http://www.olap4j.org/

relacionado a interseção dos resultados e será realizado através da comparação entre os nomes dos atributos. Exemplificando no contexto de vendas, se os resultados das duas bases contiverem o membro "*Creme Dental*" com valor de sua venda e esta medida for aditiva, o resultado final será o somatório dos valores oriundos das duas fontes de dados.

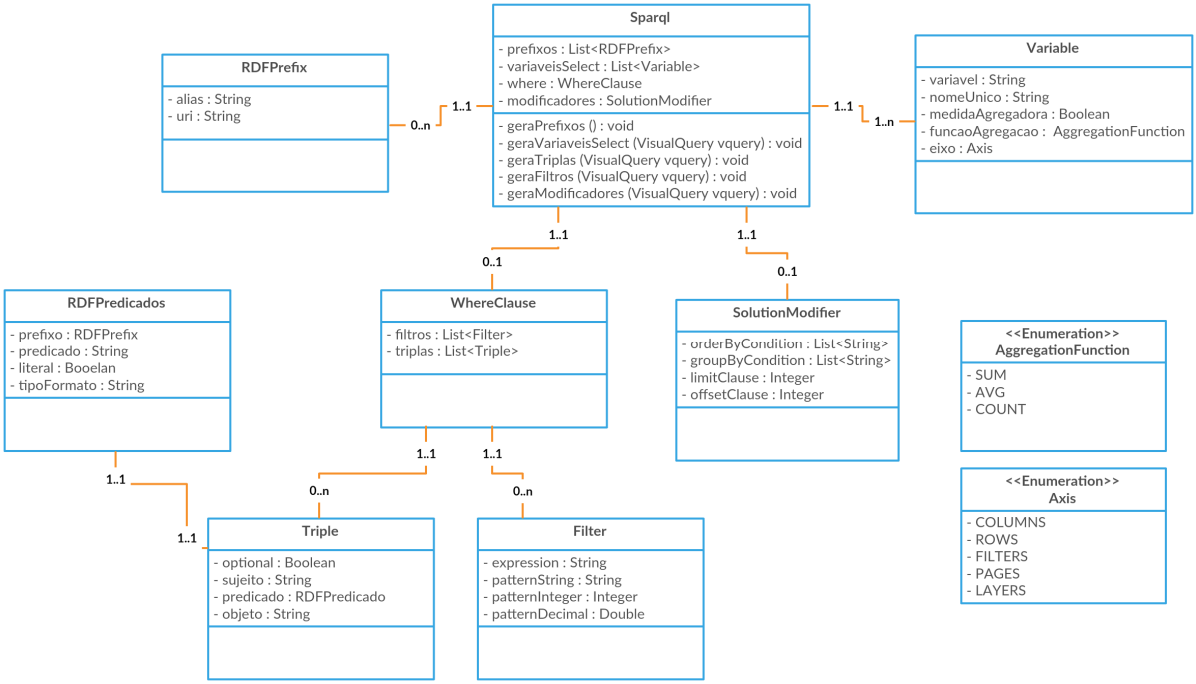

Figura 12: Diagrama de Classes para o objeto *Sparql*

O resultado de uma consulta visual é composto por células, *tuplas* e grupos. Para os eixos Colunas, Linhas e Páginas da consulta visual, o resultado da consulta retorna uma lista de *tuplas*. Para o eixo Camadas, o resultado da consulta retorna *tuplas* agrupadas em grupos. O eixo Filtro da consulta visual só serve para filtrar o conteúdo das células.

Uma *tupla* é composta por membros e contém um atributo (*ordinal*) que indica sua posição no eixo. A *tupla* tem uma dimensionalidade, ou seja, para cada nível utilizado no eixo da consulta visual, há um membro representando o nível, respeitando a ordem de inserção dos níveis nos eixos.

Os resultados oriundos das duas fontes de dados precisam ser formatados para o objeto *VisualQueryResult* para facilitar sua exibição ao usuário. No caso do resultado oriundo da fonte de dados semântica, foi modelado, incialmente, o objeto *SparqlResult* para encapsular as informações fornecidas na resposta do *Framework Apache Jena.* Esse objeto armazena os resultados de forma semelhante a uma célula do contexto multidimensional: um conjunto de membros dispostos nos eixos e um conjunto de valores na intersecção desses membros. Em outras palavras, um conjunto de membro de dimensão e os valores de medidas para cada

combinação dos membros (*SparqlQuerySolutionResult*). O diagrama de classes para o objeto *SparqlResult* é ilustrado na Figura 14.

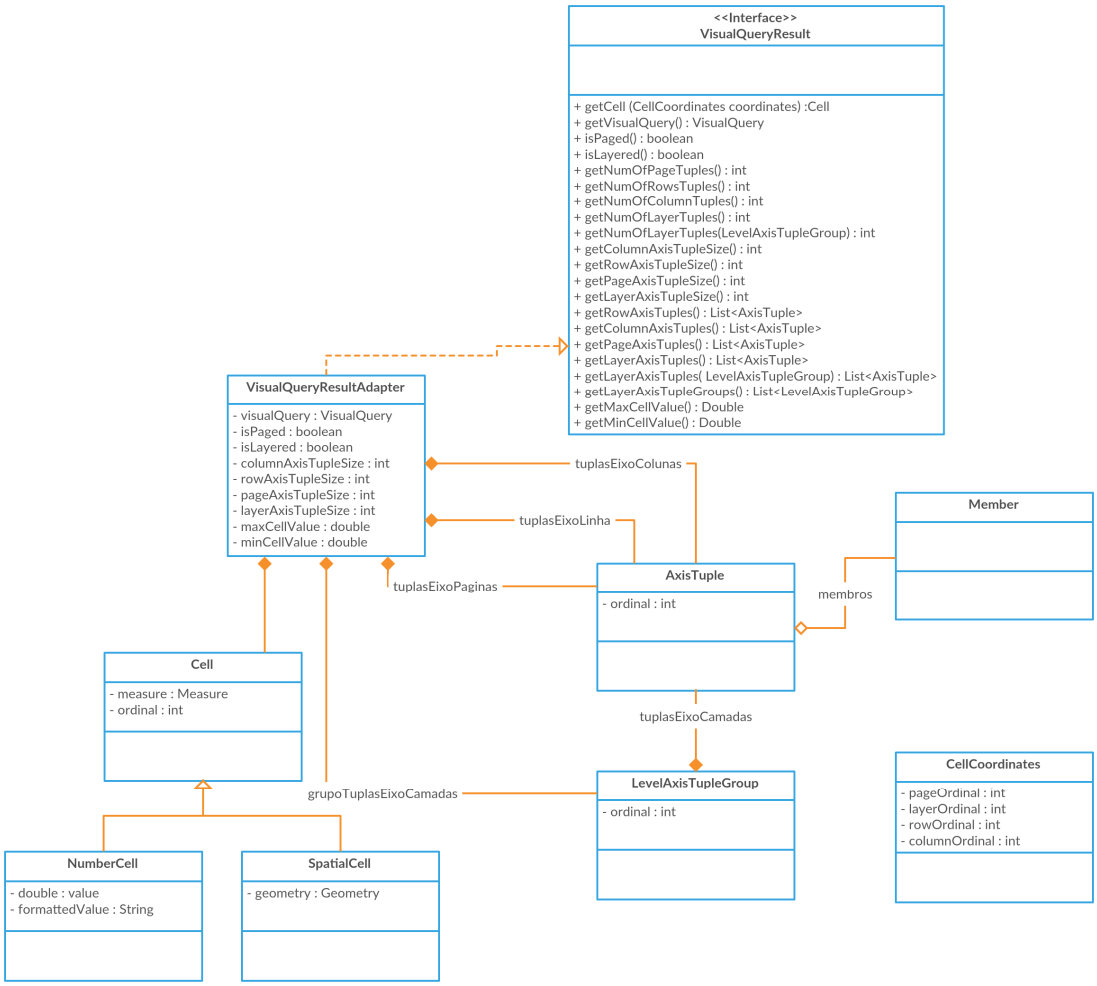

Figura 13: Diagrama de Classe para o resultado da consulta visual – *VisualQueryResult*. Traduzido de Silva (2013)

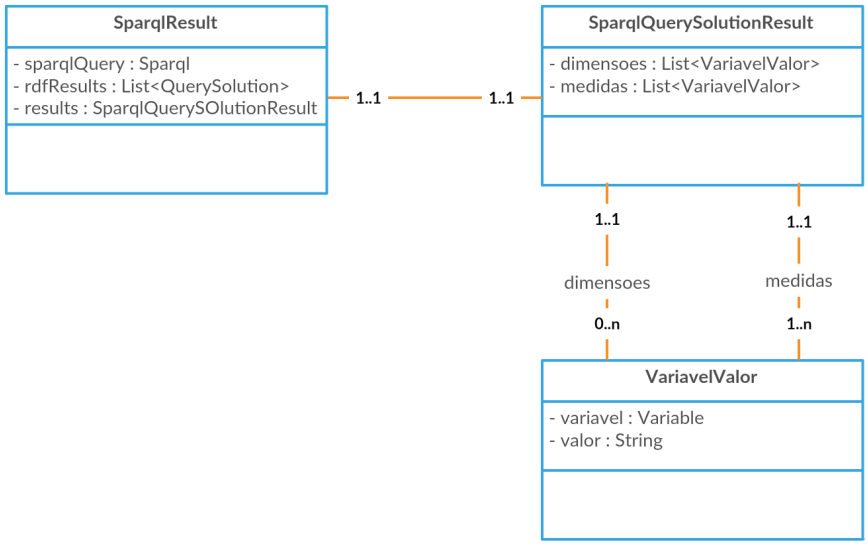

Figura 14: Diagrama de Classes para objeto *SparqlResult*

4.3 ExpSOLAP: Aspectos de implementação

Na seção anterior foi apresentada uma visão geral da solução ExpSOLAP no que tange aos aspectos de projeto. Nesta seção, a solução ExpSOLAP será descrita sob o aspecto tecnológicos, descrevendo as tecnologias utilizadas no seu desenvolvimento e detalhando a implementação da solução ExpSOLAP.

Na Figura 15 é apresentada uma nova versão da arquitetura da solução ExpSOLAP (Figura 9) indicando as tecnologias utilizadas em cada módulo. Para melhor visualização, a comunicação entre os módulos das camadas foi omitida. Os módulos destacados em azul foram os módulos adicionados na solução ExpSOLAP. A seguir, será descrita cada camada da arquitetura, dando-se ênfase aos novos módulos implementados e detalhes da implementação.

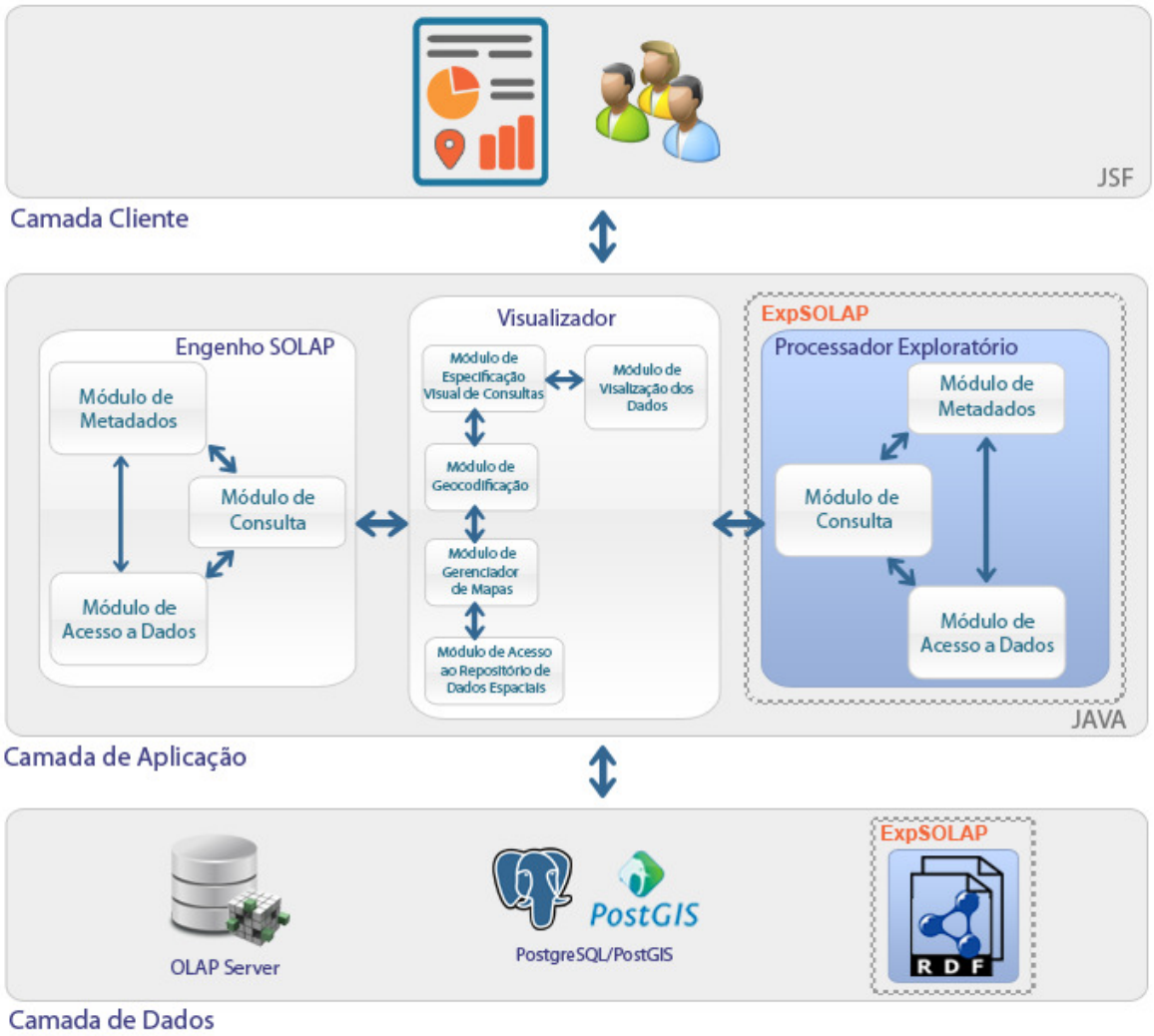

Figura 15: Arquitetura da solução ExpSOLAP (Figura 9) detalhando tecnologias utilizadas. Adaptado de Silva (2013)

4.3.1 Camada Cliente

A interface de exploração, desenvolvida utilizando as tecnologias *JavaServer Faces* (JSF) 2.0[8], *PrimeFaces*[9] e *Open Layers*[10], é dividida em dois painéis: um à esquerda da tela, contendo os metadados do cubo (medidas e dimensões) e um outro à direita, contendo as divisórias de consulta (colunas, linhas, filtros, páginas e camadas) e um painel para exibição dos resultados, em formato tabular. A formulação de consultas se dá utilizando uma linguagem de especificação visual, na qual deve-se arrastar e soltar (*drag-and-drop*) os metadados para a divisória desejada (Figura 16).

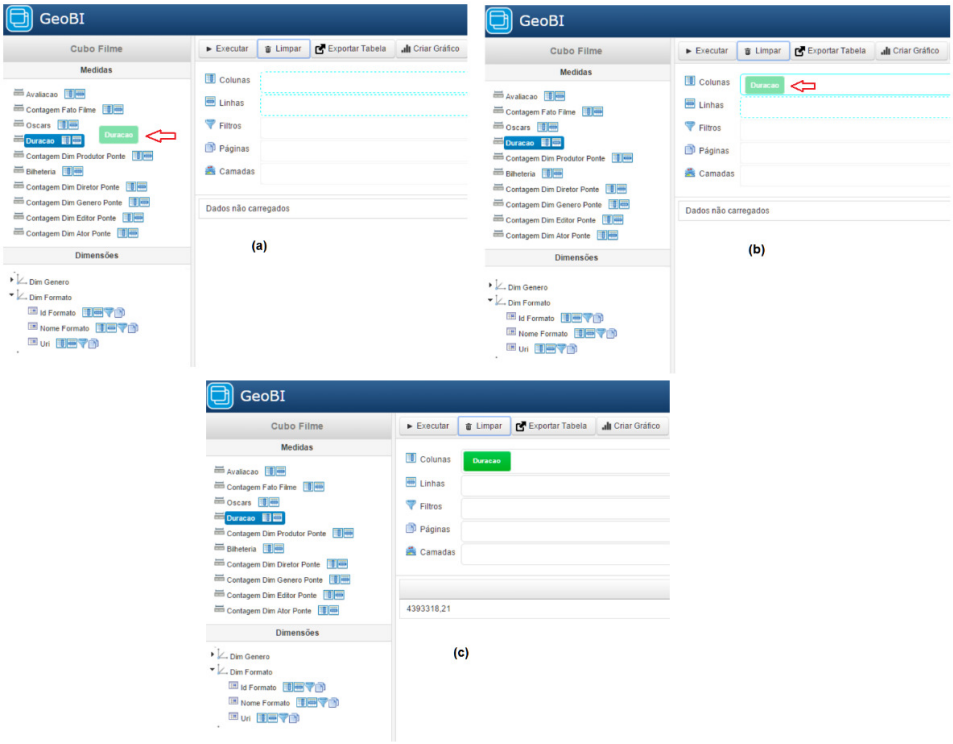

Figura 16: (a) Início do *drag-and-drop*; (b) Medida sobre o campo "Colunas"; (c) Medida no campo "Colunas" após o *drag-and-drop*

A especificação visual utilizada por Silva (2013) é uma extensão do formalismo VizQL, presente na ferramenta *Polaris* (Stolte e Hanrahan, 2000), adicionada de uma nova divisória denominada "Camadas", responsável pela seleção de membros espaciais. Essa extensão abriu o leque de possibilidades, uma vez que viabilizou o uso de operadores espaciais topológicos e a exibição dos dados em mapas, com sobreposição de camadas.

As divisórias "Linhas" e "Colunas" são utilizadas para particionar os dados em linhas e colunas, formando uma tabela. Cada célula da tabela contém um painel onde os valores das

[8] http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html
[9] http://primefaces.org/
[10] http://openlayers.org/

medidas são apresentados. A divisória “Páginas” particiona os dados em páginas; cada página detém uma porção dos dados no formato tabular especificado pelo usuário (tabela); enquanto a divisória “Camadas” é responsável por particionar a tabela por nível espacial, semelhante à divisória “Páginas”, que particiona a tabela de acordo com a combinação dos membros.

A divisória “Filtros” serve para selecionar uma determinada porção dos dados sem que seja necessário modificar a estrutura da tabela. Somente os dados relacionados aos membros presentes nessa divisória serão sumarizados e exibidos nas tabelas. Ao adicionar um nível à divisória “Filtros”, a interface irá exibir uma tela para que o usuário filtre os membros desse nível. O filtro pode ser convencional ou geográfico. No primeiro caso, os membros são incluídos ou excluídos pelo usuário (Figura 17). Já no filtro geográfico, o usuário irá especificar uma restrição espacial topológica para os membros (Figura 18). Outras dimensões espaciais podem ser utilizadas na criação das restrições.

As restrições criadas, convencionais ou geográficas, são sempre de igualdade (<membro == valor>). No caso de filtros geográficos, a consulta é executada inicialmente no repositório espacial e expressões de igualdade são formuladas com os resultados obtidos. Essa estratégia possibilitou filtrar, de forma idêntica, em todas as fontes de dados integradas no Framework SOLAP.

A geocodificação de membros de dimensão dá-se em três fases: inicialmente, o usuário deve selecionar a hierarquia em que desejar geocodificar os membros (Figura 19); em seguida, deve selecionar a tabela/coluna onde se encontram os dados espaciais para o processo de geocodificação (Figura 20); e, por fim, o mapeamento é exibido ao usuário, que pode confirmar ou modificar antes de persistir no banco de dados (Figura 21).

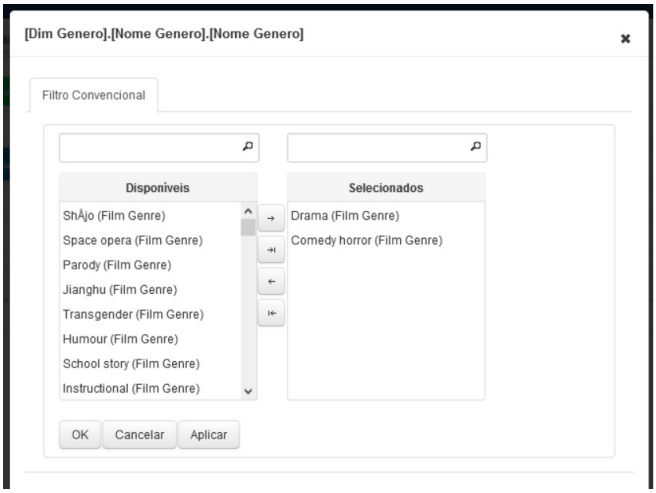

Figura 17: Filtro Convencional

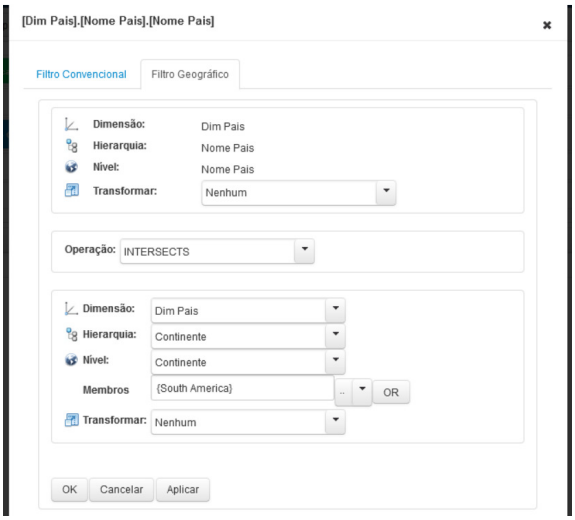

Figura 18: Filtro Geográfico

Figura 19: Geocodificação - seleção do nível para o processo de geocodificação

Figura 20: Geocodificação - seleção da tabela espacial

Figura 21: Geocodificação - resultado do processo de geocodificação

4.3.2 Camada de Aplicação: Processador Exploratório

A camada de aplicação é responsável por implementar toda a lógica da aplicação, dispondo, para tanto, de três módulos: Visualizador, Engenho SOLAP e Processador Exploratório. Este último é responsável por prover a comunicação entre a aplicação e a fonte de dados semântica. Foi desenvolvido por completo utilizando a linguagem de programação Java.

O Processador Exploratório possui funcionalidades semelhantes às do Engenho SOLAP. Porém, o foco deste módulo é gerenciar fonte de dados semânticas, a exemplo de acessar e consultar, utilizando a linguagem SPARQL, fonte de dados semânticas representadas no formato RDF. A conversão da VQL em SPARQL foi possível pelo desenvolvimento de um tradutor, enquanto a execução é realizada com auxílio do *Framework Apache Jena*. O Processador Exploratório divide-se em três módulos: módulo de acesso aos dados, módulo de carga dos metadados e módulo de consulta, que serão descritos a seguir.

Módulo de Acesso aos Dados

O módulo de acesso aos dados é responsável pelo cadastro de uma fonte de dados semântica, possibilitando que ontologias sejam incorporadas e exploradas pela solução ExpSOLAP.

O cadastro da fonte de dados semântica se dá através de propriedades de conexão, as quais referem-se à localização física aonde os arquivos RDF estão armazenados (Figura 22) – que deve ser relativa à máquina na qual está sendo executada a solução ExpSOLAP – e a prefixos/*namespaces* presentes na fonte semântica integrada (Figura 23). Se a ontologia integrada contiver, por exemplo, dez *namespaces* em seus arquivos RDF, todos eles devem ser informados.

Tanto o módulo de carga de metadados quanto o módulo de consulta do Processador Exploratório interagem com este módulo consumindo as propriedades de conexão. Enquanto o primeiro recupera os prefixos/*namespaces* para associar seus predicados aos metadados de um cubo multidimensional, o módulo de consulta recupera a localização física dos arquivos RDF para explorá-los diretamente utilizando *Framework Apache Jena*.

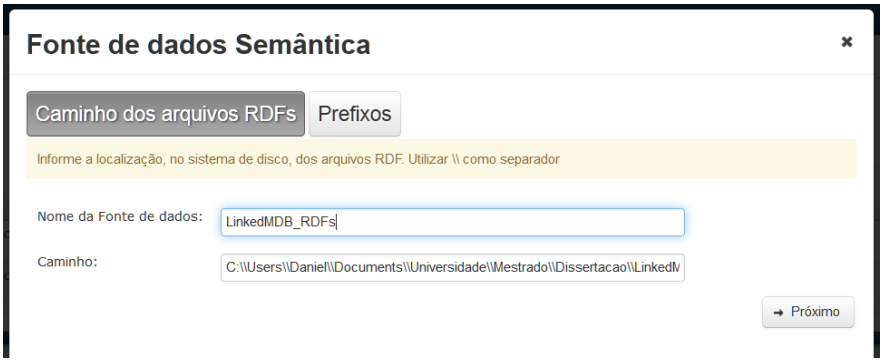

Figura 22: Cadastro de uma nova fonte de dados semântica

Figura 23: Cadastro de uma nova fonte de dados semântica - prefixos

Módulo de Carga de Metadados

A integração entre as duas fontes de dados heterogêneas (semiestruturada semântica e estruturada multidimensional) incorporadas à solução ExpSOLAP só ocorre de fato após a correlação entre as mesmas. Em outras palavras, é necessário mapear os metadados da ontologia integrada com os metadados do cubo multidimensional.

O módulo de carga de metadados tem a responsabilidade de realizar esse mapeamento, possibilitando que uma mesma consulta – formalizada através da linguagem de consulta visual - seja executada em ambas as fontes de dados de forma correspondente. Para tanto, o referido módulo, ao mesmo tempo em que cadastra os metadados da fonte de dados semântica (caracterizados por predicados contidos na fonte semântica importada), recupera os metadados do cubo (e.g: membros de dimensões, níveis e medidas), associando ambos e indicando, caso existam, as transformações necessárias. O cadastro, a conexão com o cubo multidimensional e a leitura de seus metadados é de responsabilidade do módulo de carga de metadados do Engenho SOLAP, conforme descrito por Silva (2013).

O processo de mapeamento consiste em quatro etapas: correlação das fontes de dados heterogêneas, mapeamento de dimensão, de nível e de medidas. Na primeira etapa, selecionando uma fonte de dados multidimensional já existente (Figura 24), o usuário deve informar a fonte de dados semântica - previamente cadastrada no módulo de acesso aos dados - e nomear a associação (Figura 25).

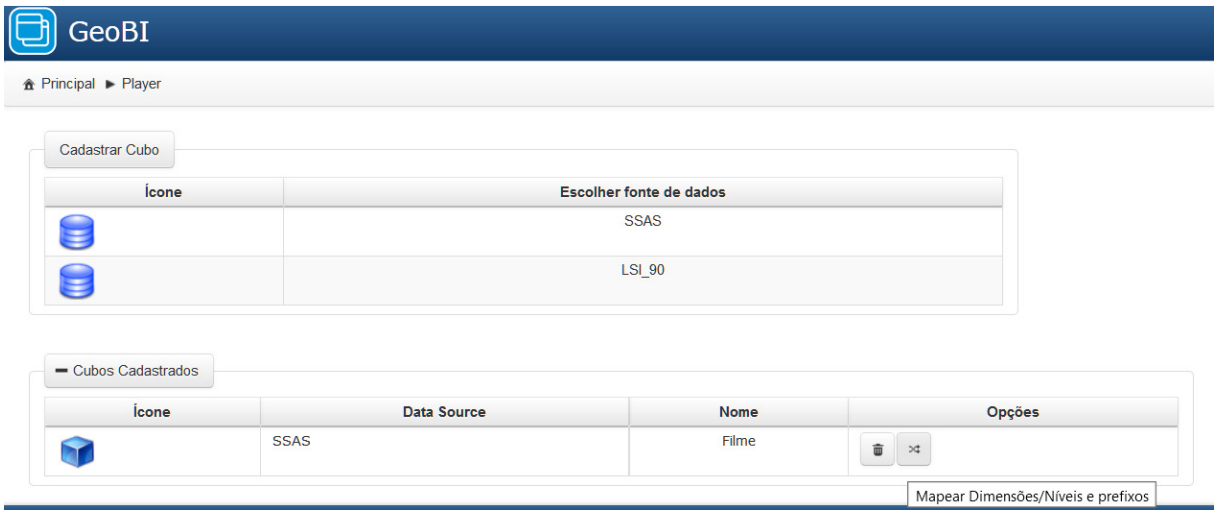

Figura 24: Interface para mapear Cubo com Fonte Semântica

Figura 25: Mapeamento Cubo com Fonte Semântica - Escolha da Fonte Semântica

As próximas fases consistem, respectivamente, em associar cada dimensão (Figura 26), nível (Figura 27) e medida (Figura 28) a exatamente um predicado. Por exemplo, vamos considerar um DW no contexto de Filmes que contenha a dimensão "*Ator*" com três atributos: *id, nome_ator* e *uri.* Essa dimensão deve ser mapeada nos seguintes predicados, respectivamente: *movie:actor_id*, *movie:actor_name* e *movie:actor*.

Ademais, deve-se indicar se o predicado é um recurso ou um literal, sendo que, neste último caso, faz-se necessário preencher informações relativas ao tipo e formato do literal. No caso específico de medida, o mapeamento ainda requer que uma função de agregação (SUM, AVG ou COUNT) seja indicada, pois será a função padrão utilizada para sumarizar os valores desta medida. A sumarização dos valores deve ser a mesma nas duas fontes de dados.

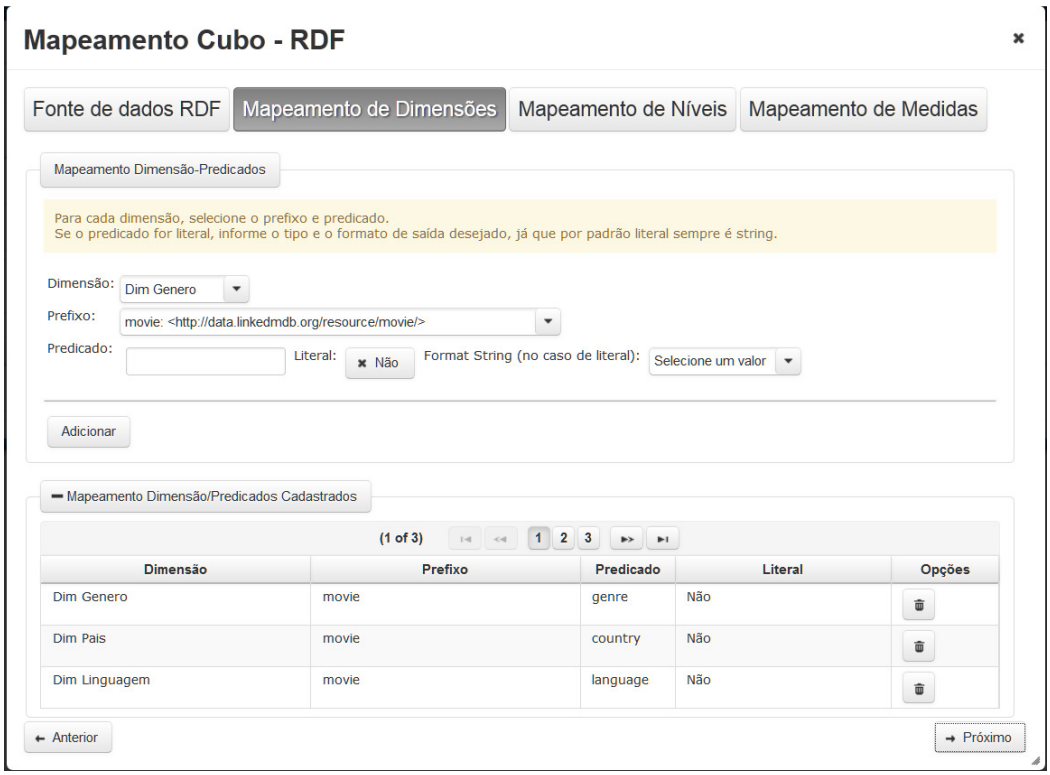

Figura 26: Mapeamento Cubo com Fonte Semântica - Mapeamento de Dimensões

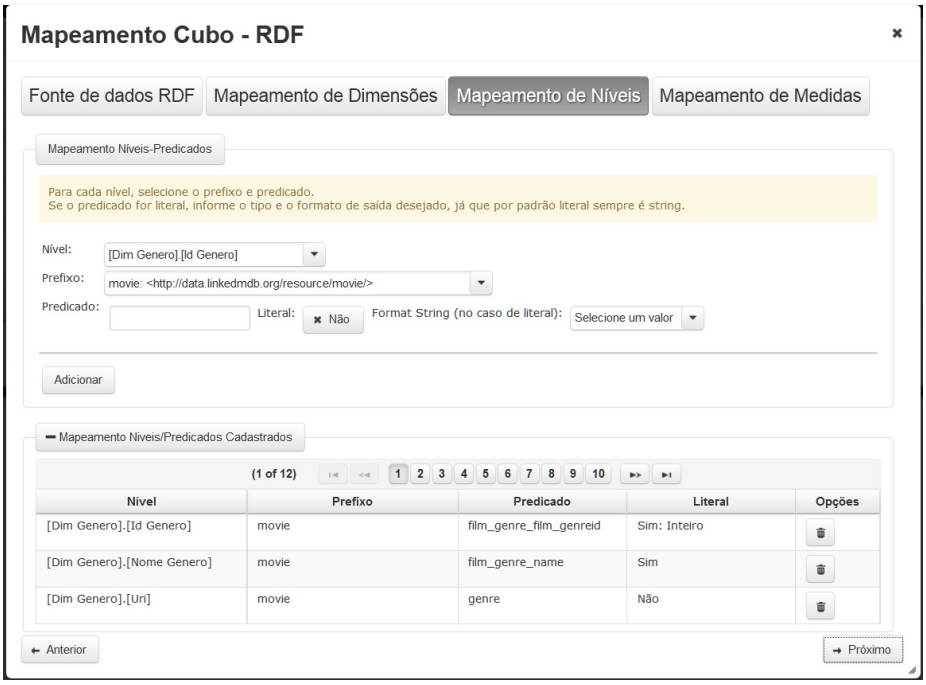

Figura 27: Mapeamento Cubo com Fonte Semântica - Mapeamento de Níveis

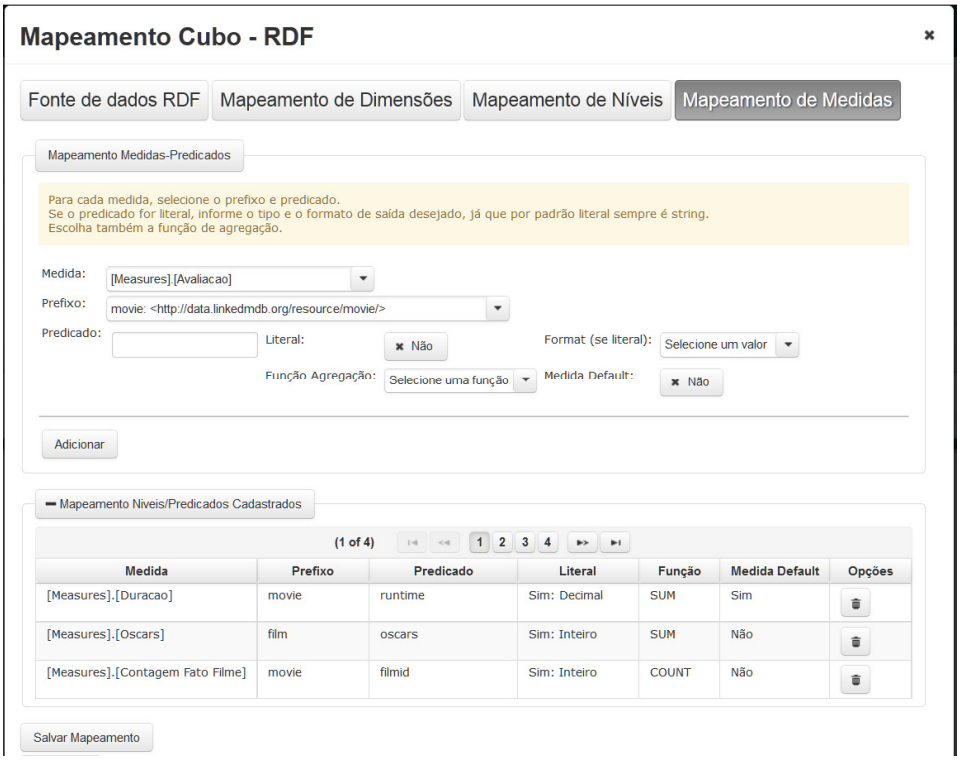

Figura 28: Mapeamento Cubo com Fonte Semântica - Mapeamento de Medidas

Os módulos de Acesso aos Dados e de Carga de Metadados são responsáveis por integrar novas fontes de dados que vierem a ser incorporadas à solução ExpSOLAP. Para tanto, o processo de integração é feito de forma manual. O usuário deve cadastrar as propriedades de conexão e realizar a associação entre predicados da fonte semântica e metadados (dimensões,

níveis e medidas) do cubo. A fonte semântica integrada não precisa, necessariamente, conter todas as informações (predicados) relacionados aos metadados do cubo. Assim, é possível integrar fontes semânticas relacionadas apenas a algumas dimensões do cubo. Neste caso, a fonte semântica só será consultada caso a consulta visual, formulada na interface da solução ExpSOLAP, contenha a dimensão mapeada.

Módulo de Consulta

O módulo de consulta é o cerne do Processador Exploratório, sendo responsável por traduzir as consultas visuais para a linguagem destino (SPARQL), de executá-las – por meio do módulo de acesso aos dados, e de converter os resultados para um modelo padrão utilizado pela aplicação, repassando-os para o módulo visualizador, que se encarregará da exibição ao usuário. O diagrama UML de atividades ilustrado na Figura 29 exibe o fluxo das atividades necessárias para a realização da consulta.

Figura 29: Diagrama UML de atividades referente às funcionalidades do Módulo de Consulta

As atividades relacionadas ao módulo de consulta têm início no momento em que o usuário formula uma consulta em VQL na interface cliente da solução. A consulta visual, representada por uma instância do objeto *VisualQuery* (Figura 10), será passada como parâmetro para o referido módulo, para, então, ser traduzida e executada. Para realizar a tradução da consulta VQL para SPARQL (atividade A1), um tradutor foi desenvolvido como parte deste trabalho de dissertação e integrado à solução ExpSOLAP. A tradução da consulta visual é baseada na gramática BNF de SPARQL (W3C, 2008).

O algoritmo exibido no Código Fonte 5 ilustra, em alto nível, o método *vql2Sparql*, responsável pela tradução da VQL para SPARQL. O método recebe como parâmetro apenas a consulta visual (*VisualQuery*) e executa operações específicas para cada uma das cláusulas que compõem uma consulta SPARQL (linha 3 a 7), com exceção da cláusula *Dataset Definition* que foi omitida, visto que, a partir da localização física dos arquivos (previamente cadastrada no módulo de acesso a dados), os mesmos são consultados diretamente.

A primeira operação realizada pelo método exibido no Código Fonte 5 (linha 3) é preencher a cláusula de prefixos do objeto *Sparql* e, consequentemente, da consulta SPARQL gerada. Através da função *geraPrefixos()*, o módulo de consulta se comunica com o módulo

de acesso a dados recuperando todos os prefixos cadastrados (propriedades da fonte semântica), associando-os ao objeto *Sparql*.

Código Fonte 5: Algoritmo, em alto nível, da conversão VQL para SPARQL

```
1    public static Sparql vql2Sparql(VisualQuery vQuery) {
2       Sparql sparql = new Sparql();
3       sparql.geraPrefixos();
4       sparql.geraVariaveisSelect(vQuery);
5       sparql.geraTriplas(vQuery);
6       sparql.geraFiltros(vQuery);
7       sparql.geraModificadores(vQuery);
8       return sparql;
9    }
```

A próxima operação é gerar as variáveis presentes na cláusula SELECT do objeto *Sparql* e da consulta (*result clause*). A função *geraVariaveisSelect(vQuery)* (Código Fonte 6), irá percorrer todas as divisórias da consulta visual (linhas, colunas, páginas e camadas) e, para cada componente selecionado (hierarquia de níveis ou de medidas), será criada uma nova instância do objeto *Variable*, indicando um nome único (gerado automaticamente) e o eixo em que este está presente (linhas 3 a 6). No caso de medidas selecionadas nas divisórias, especificamente nas linhas ou colunas, ao criar uma instância do objeto *Variable*, ter-se-á uma indicação de que a variável é uma medida (*medidaAgregadora*), informando também a sua respectiva função agregadora (linhas 7 a 10). A função de agregação é recuperada através do módulo de metadados, que disponibiliza o mapeamento entre as fontes de dados e, por conseguinte, a função agregadora da medida. Todas as instâncias de variáveis criadas serão adicionadas à cláusula SELECT.

O procedimento seguinte para conversão da VQL para SPARQL é especificar as triplas contidas nos padrões de grafo da cláusula WHERE. De forma semelhante à função *geraVariaveisSelect*, o método *geraTriplas(vQuery)*(

Código Fonte 7) irá percorrer todas as divisórias da consulta visual e, para cada componente selecionado, será criada uma nova instância do objeto *Triple* (linhas 3 a 8). Os valores dos atributos desse objeto serão preenchidos com o auxílio do mapeamento entre as fontes de dados, recuperado via módulo de metadados. A tripla será caracterizada por: *i)* um *alias* atribuído ao sujeito - um valor único para a fonte semântica e gerado com base no nome do mapeamento entre as fontes semânticas (*getSujeitoCubo()* - linha 4); *ii)* por um predicado, recuperado do mapeamento, correspondente ao nível da dimensão ou a medida (linhas 5 e 6); e por um nome único da variável, idêntico ao adicionado à clausula SELECT, atribuído ao objeto da tripla. Dessa forma, a tripla formulada seguirá o seguinte padrão: <*alias,*

predicadoRecuperado, nomeUnico>. Todas as instâncias de triplas criadas serão adicionadas à cláusula WHERE.

Código Fonte 6: Algoritmo, em alto nível, da geração das variáveis da cláusula SELECT

```
1   private static void geraVariaveisSelect(VisualQuery vQuery) {
2      for (Hierarchy h : vQuery.getHierarchies(Axis.COLUMNS)) {
3          Variable v = new Variable();
4          v.setVariavel(getVarName(h.getName()));
5          v.setUniqueName(h.getUniqueName());
6          v.setEixo(Axis.COLUMNS);
7          if (h.isMeasure()) {
8             v.setMedidaAgregadora(true);
9             v.setFuncaoAgregacao(getAggFunctMapeamento(h.getName()));
10         }
11         sparql.addVariable(v);
12     }
13
14     for (Hierarchy h : vQuery.getHierarchies(Axis.ROWS)) {...}
15     for (Hierarchy h : vQuery.getHierarchies(Axis.LAYERS)) {...}
16     for (Hierarchy h : vQuery.getHierarchies(Axis.PAGES)) {...}
17  }
```

Código Fonte 7: Algoritmo, em alto nível, da formulação das triplas da cláusula WHERE

```
1   private static void geraTriplas(VisualQuery vQuery) {
2      for (Hierarchy h : vQuery.getHierarchies(Axis.COLUMNS)) {
3          Triple t = new Triple();
4          t.setSujeito(getSujeitoCubo());
5          RDFPredicados predicado = getPredicado(h.getUniqueName());
6          t.setPredicado(predicado);
7          t.setObjeto(getVarName(h.getName()));
8          sparql.getWhere().getTriplas().add(t);
9      }
10
11     for (Hierarchy h : vQuery.getHierarchies(Axis.ROWS)) {...}
12     for (Hierarchy h : vQuery.getHierarchies(Axis.LAYERS)) {...}
13     for (Hierarchy h : vQuery.getHierarchies(Axis.PAGES)) {...}
14  }
```

Se a consulta visual contiver alguma restrição, isto é, algum componente selecionado na divisória Filtros, a mesma também deve estar presente no objeto *Sparql* e na consulta SPARQL resultante. Nesse sentido, o método *geraFiltros(vQuery)* (Código Fonte 8) irá percorrer os membros das dimensões filtradas, criando instâncias do objeto *Filter*, caracterizadas pelo nome da variável (idêntico ao adicionado à clausula SELECT) como expressão e o nome do membro selecionado como valor. Assim, tem-se filtros de igualdade

(<*membro* == *valor*>) adicionados à lista de restrições da cláusula WHERE do objeto *Sparql*.

Código Fonte 8: Algoritmo, em alto nível, da formulação de filtros

```
1    private static void geraFiltros(VisualQuery vQuery) {
2       for (Member m : vQuery.getDrilledLevelMap().getValues()) {
3           Filter f = new Filter();
4           f.setExpression(getVarName(m.getName()));
5           f.setPattern(m.getValue());
6           sparql.getWhere().getFiltros().add(f);
7       }
8    }
```

Essa metodologia de restrições por igualdade implementada no Framework SOLAP serve tanto para restrições convencionais quanto geográficas. No caso do filtro geográfico, este primeiro é executado no repositório espacial e, a partir dos resultados dessa consulta, são formuladas as expressões do filtro. Por exemplo, ao formular a seguinte restrição topológica: "Países que fazem fronteira – *touches* – com o Brasil", o repositório espacial é consultado e retorna todos os países que fazem fronteira com o Brasil (Argentina, Paraguai, Uruguai, etc.). O método *geraFiltros(vQuery)* do tradutor desenvolvido irá gerar restrições no seguinte formato: *?nomePais == 'Argentina', ?nomePais == 'Paraguai'* ou *?nomePais == 'Uruguai'*. Em outras palavras, o filtro geográfico é executado como filtro convencional textual na consulta SPARQL gerada. A adoção da referida metodologia se adaptou à peculiaridade da linguagem SPARQL, que ainda não oferece nenhum consenso ou padrão para consultas espaciais, viabilizando a integração espacial entre fontes de dados heterogêneas. Com a utilização de extensões espaciais da linguagem SPARQL, o método *geraFiltros(vQuery)* poderá ser aperfeiçoado para criar restrições espaciais utilizando funções e relacionamentos geográficos (*bounding box*, *touches*, etc).

Por fim, mas não menos importante, tem-se a cláusula dos modificadores do objeto *Sparql* e da consulta SPARQL (*query modifiers*). O método *geraModificadores(vQuery)*(Código Fonte 9) percorre todas as variáveis presentes na cláusula SELECT, agrupando-as, com exceção das medidas, através de uma cláusula GROUP BY.

A fim de ilustrar todo o processo de tradução da VQL para SPARQL realizada pelo módulo de consulta, um exemplo de consulta visual formalizada na interface da solução ExpSOLAP é ilustrado na Figura 30, enquanto no Código Fonte 10 tem-se a consulta SPARQL resultante gerada pelo tradutor. A consulta SPARQL, em sintaxe adequada, é obtida acessando-se o método *toString()* do objeto *Sparql*.

Código Fonte 9: Algoritmo, em alto nível, da geração de modificadores

```
1    private static void geraModificadores(VisualQuery vQuery) {
2       SolutionModifier solMod = new SolutionModifier();
3       for (Variable v : sparql.getVariaveisSelect()) {
4           if (!v.getMedidaAgregadora()) {
5              solMod.getGroupByCondition().add(v.getVariavel());
6           }
7       }
8       sparql.setSolutionModifiers(solMod);
9    }
```

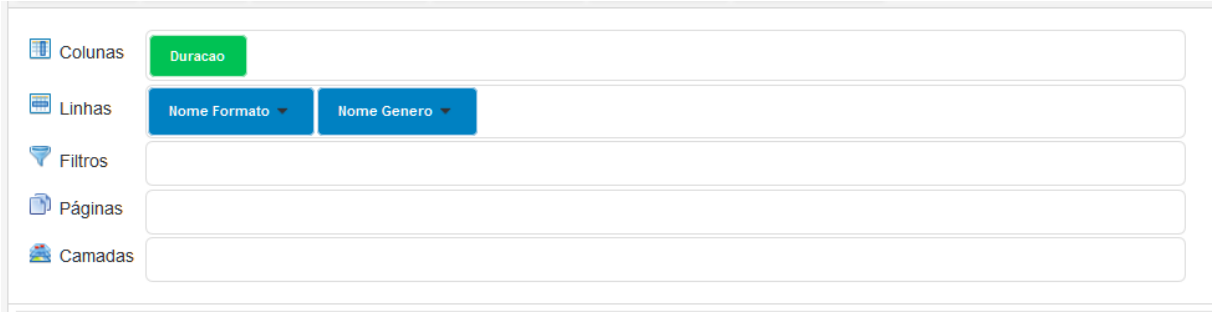

Figura 30: Exemplo de Consulta Visual utilizando as divisórias Colunas e Linhas

Código Fonte 10: Consulta SPARQL, gerada pelo tradutor implementado, correspondente a consulta visual da Figura 30

```
1    PREFIX movie: <http://data.linkedmdb.org/resource/movie/>
2    PREFIX dbpedia: <http://dbpedia.org/property/>
3    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4    ...
5    PREFIX time: <http://www.w3.org/2006/time#>
6    PREFIX timeExt: <http://150.165.75.163/2015/10/time#>
7    PREFIX film: <http://150.165.75.163/2015/10/movie#>
8    SELECT ?nomeformato ?nomegenero (SUM(xsd:decimal(?duracao)) AS
     ?duracao_sum)
9    WHERE {
10      ?filme_uri_suj movie:film_format ?dimformato_uri .
11      ?dimformato_uri movie:film_format_name ?nomeformato .
12      ?filme_uri_suj movie:genre ?dimgenero_uri .
13      ?dimgenero_uri movie:film_genre_name ?nomegenero .
14      ?filme_uri_suj movie:runtime ?duracao .
15   }
16   GROUP BY ?nomeformato ?nomegenero
```

Concluída a tradução da VQL para SPARQL, o referido módulo irá executar a consulta nos arquivos RDF correspondentes à fonte de dados semântica integrada (atividade A2). Para tanto, o módulo de consulta se comunica com o módulo de acesso aos dados, recuperando a localização física que indica onde os arquivos RDF estão armazenados (propriedades de conexão). Com a consulta SPARQL formulada (encapsulada no objeto *Sparql*) e a localidade

dos arquivos RDF, o módulo de consulta, utilizando o *Framework Apache Jena*, irá executar a consulta SPARQL na base semântica, conforme ilustrado no Código Fonte 11.

Código Fonte 11: Algoritmo, em alto nível, para execução da consulta na fonte semântica

```
1    private static List<QuerySolution> query(Sparql sparql) {
2       String rdfDirectory = sparql.getMapeamentoMDXRDF();
3       List<String> path = new ArrayList<String>();
4       File rdfsFolderFile = new File(rdfDirectory);
5       if (rdfsFolderFile.exists()){
6          for (File f : rdfsFolderFile.listFiles()) {
7             path.add("file:" + f.getAbsolutePath());
8          }
9          Dataset dataset = DatasetFactory.create(path);
10         Query query = QueryFactory.create(sparql.toString());
11         QueryExecution qe = QueryExecutionFactory.create(query, dataset);
12         ResultSet results = qe.execSelect();
13         List<QuerySolution> rows = new ArrayList<QuerySolution>();
14         while (results.hasNext()) {
15            QuerySolution row = results.nextSolution();
16            rows.add(row);
17         }
18         return rows;
19      }
20      return null;
21   }
```

Todos os arquivos RDF disponíveis na localização obtida serão explorados, pelo *Framework Apache Jena*, e armazenados em memória, como um único grande grafo (linhas 7 a 10). Esse grafo construído pelo *Apache Jena* é complexo e volumoso, sendo composto por milhares de triplas. Tal procedimento efetuado pelo *Framework Apache Jena* apresenta como desvantagem o desempenho de execução da consulta. Isso porque para cada tripla contida na cláusula WHERE da consulta SPARQL, é necessário comparar, individualmente, com cada tripla contida no grafo construído. Em seguida, a consulta é criada através do texto, sendo executada sob o grafo correspondente às triplas de todos os arquivos RDF (linhas 11 e 12).

O tipo do resultado retornado pela execução da consulta é um *ResultSet* (linha 13). Estes são iterados e recuperados isoladamente através do objeto *QuerySolution* (linha 16). O algoritmo exibido no

Código Fonte 12 ilustra um trecho da resposta da consulta visual formalizada na Figura 30 encapsulada no objeto *QuerySolution.*

Para realizar a conversão do formato da resposta (atividade A3), a lista de resultados, encapsulados no objeto *QuerySolution*, é iterada, tendo suas informações encapsuladas em

objeto *SparqlResult* (Figura 14). O algoritmo exibido no Código Fonte 13 ilustra um trecho da resposta da consulta visual formalizada na Figura 30 encapsulada no objeto *SparqlResult*. Em seguida, os resultados encapsulados no *SparqlResult* serão convertidos em uma instância do *VisualQueryResult* (Figura 13) para, em seguida, realizar o *merge* – combinação dos resultados pelo nome – dos resultados com a base estruturada.

Código Fonte 12: Resultado, em objeto *QuerySolution*, da consulta visual (Figura 30)

```
1    (?nomeformato = "Super 35 mm film (Film Format)") (?nomegenero =
     "Black comedy (Film Genre)") (?duracao_sum = "259"^^xsd:decimal)
2    (?nomeformato = "DVD-Video (Film Format)") (?nomegenero =
     "Musical (Film Genre)") (?duracao_sum = "127"^^xsd:decimal)
3    (?nomeformato = "Super 35 mm film (Film Format)") (?nomegenero =
     "Disaster (Film Genre)") (?duracao_sum = "1031"^^xsd:decimal)
4    (?nomeformato = "70 mm Todd-AO (Film Format)" ) (?nomegenero =
     "Biographical (Film Genre)" ) (?duracao_sum = "504"^^xsd:decimal)
5    ...
```

Código Fonte 13: Resultado, em objeto *SparqlResult*, da conversão da resposta obtida (Código Fonte 12)

```
1    ({[Dim Formato].[Nome Formato] = Super 35 mm film (Film Format),
     [Dim Genero].[Nome Genero] = Black comedy (Film Genre)},
     {[Measures].[Duracao] = 259})
2    ({[Dim Formato].[Nome Formato] = DVD-Video (Film Format), [Dim
     Genero].[Nome Genero] = Musical (Film Genre)},
     {[Measures].[Duracao] = 127})
3    ({[Dim Formato].[Nome Formato] = Super 35 mm film (Film Format),
     [Dim Genero].[Nome Genero] = Disaster (Film Genre)},
     {[Measures].[Duracao] = 1031})
4    ({[Dim Formato].[Nome Formato] = 70 mm Todd-AO (Film Format),
     [Dim Genero].[Nome Genero] = Biographical (Film Genre)},
     {[Measures].[Duracao] = 504})
5    ...
```

Observa-se que a principal diferença na representação das respostas entre os objetos *QuerySolution* e *SparqlResult* é o modo como a chave é especificada. No primeiro tipo, a chave é o nome único da variável da consulta SPARQL, enquanto no segundo caso é o nome correspondente ao membro da fonte de dados multidimensional. Essa diferença é primordial para garantir a eficiência da exibição dos resultados, porque o módulo visualizador, ao montar a exibição dos resultados, seja por meio de tabelas, gráficos ou mapas, já possui os valores resultantes da consulta na fonte de dados multidimensional e precisa apenas atualizar o valor com base no resultado da fonte de dados semântica, que será recuperada a partir do nome do membro (chave). Em outras palavras, a junção é realizada pelo nome.

4.2.3 Camada de Dados

Na Camada de Dados, encontram-se os elementos que serão processados pela camada de aplicação e visualizados pelo usuário na camada cliente. Na solução ExpSOLAP, a referida camada é constituída de três repositórios de dados: Servidores multidimensionais (servidores de cubo) - espaciais ou não, SGBD *PostgreSQL* com extensão *PostGIS* e um repositório de arquivos semânticos (ontologias representadas em arquivos RDF).

Os dois primeiros repositórios são nativos do Framework SOLAP e permitem o acesso a vários servidores multidimensionais, de diferentes tecnologias e fabricantes, podendo ser espacial ou não. O repositório de dados no SGBD *PostgreSQL + PostGIS* é responsável por armazenar os dados espaciais resultantes da geocodificação dos membros do cubo.

O repositório de arquivos semânticos foi adicionado à camada de dados na solução ExpSOLAP, e tem a responsabilidade de armazenar, especificadamente, arquivos semânticos semiestruturados - arquivos RDF.

4.4 Fluxo de Execução

Nesta seção, pretende-se exemplificar o funcionamento da solução e a interação entre os módulos da arquitetura. Na Figura 31, é ilustrado o diagrama UML de atividades correspondente.

O processo se inicia com o usuário, manipulando componentes na interface da camada cliente. O módulo de especificação visual, do módulo visualizador, transforma as interações do usuário com a interface em uma consulta visual (VQL). Em seguida, paralelamente, a VQL é repassada para os módulos de consulta do Engenho SOLAP e do Processador Exploratório, que têm a responsabilidade de converter a VQL para MDX e SPARQL, respectivamente. Se a consulta visual formulada contiver filtros geográficos, a restrição espacial será primeiramente executada no repositório espacial, tendo seu resultado encaminhado ao Engenho SOLAP e Processador Exploratório juntamente com a VQL.

Após a conversão da consulta para a linguagem destino, o módulo de consulta do Engenho SOLAP e do Processador Exploratório executa a consulta na respectiva fonte de dados. Para tanto, se comunicam com seus respectivos módulos de metadados e acesso aos dados, recuperando informações sobre a fonte de dados. Por fim, os dados resultantes são convertidos para um formato padrão, sendo encaminhados para o módulo de especificação visual do módulo Visualizador. Esse último tem a função de realizar a junção dos resultados

oriundos de ambas fontes de dados, transformando-os para exibi-los ao usuário final através da camada cliente.

Caso os dados resultantes contenham algum componente espacial, os mesmos serão exibidos em um mapa após recuperação das características espaciais (coordenadas). Esse fluxo é alternativo e só será executado caso algum membro espacial esteja presente na consulta. Se o fluxo for executado, contará com o envolvimento dos outros três módulos do Visualizador: o gerenciador de mapas, o repositório espacial e a geocodificação.

4.5 Considerações Finais

Neste capítulo, foi apresentada a solução ExpSOLAP, que integra, em uma única ferramenta analítica, dados semiestruturados semânticos com fontes tradicionais de dados estruturados, permitindo análises espaciais em ambas fontes de dados. O desenvolvimento da solução partiu do Framework SOLAP, o qual foi estendido com a adição de novos módulos referentes ao gerenciamento de fontes semânticas. O processo de extensão do Framework resultou em uma nova arquitetura, descrita neste capítulo.

O principal ponto positivo da solução é a integração de fontes de dados heterogêneas, nos formatos estruturados e semiestruturados, possibilitando uma análise mais rica e completa. Por outro lado, uma limitação refere-se às restrições impostas pelo Framework SOLAP, a exemplo da disponibilização de apenas operadores topológicos para análise espacial. O ponto crítico está relacionado ao custo considerável para executar a consulta na fonte semântica, visto que os arquivos RDF são armazenados em memória, sendo necessário um bom poder computacional para que estes sejam explorados. Entretanto, apesar da deficiência no desempenho, a solução ExpSOLAP ao integrar bases semânticas expande seu domínio de análise e ganha flexibilidade ao poder adicionar novas fontes de dados através de um mapeamento semântico manual.

A seguir, será apresentado o exemplo prático realizado para validar a solução proposta, assim como um experimento realizado a fim de avaliar o tempo de execução das consultas.

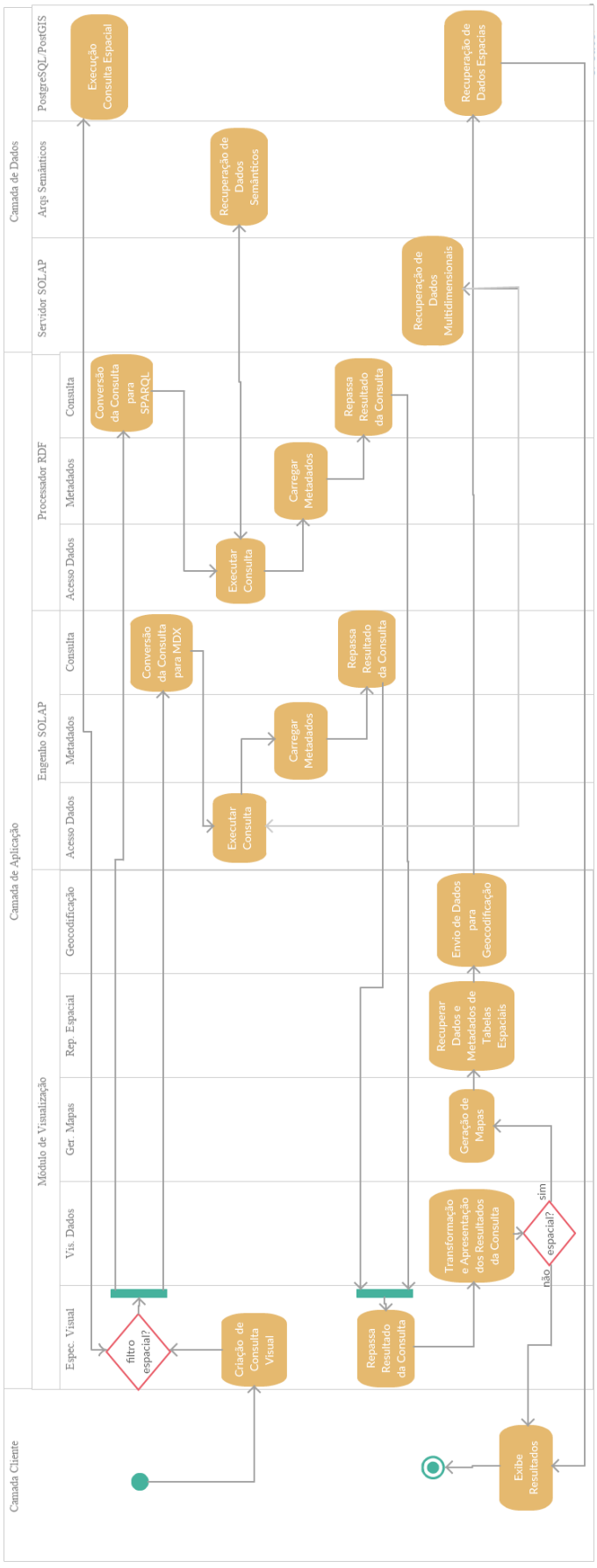

Figura 31: Diagrama de Atividades simulando o funcionamento da solução proposta

Capítulo 5 – Avaliação e Exemplo Prático

Este capítulo trata da validação da solução ExpSOLAP, proposta nesta dissertação. Para isso, um exemplo prático utilizando dados de filme interconectados (*Linked Data*) será apresentado. Na seção 5.1, é ilustrada uma análise dos dados coletados para o exemplo prático. Na seção 5.2, é detalhado o exemplo prático aplicado ao contexto de filmes, apresentando as consultas analíticas a partir da exploração da solução ExpSOLAP, assim como a avaliação realizada considerando o tempo de execução das consultas. Por fim, na seção 5.3, são expostas as considerações finais deste capítulo.

5.1 Análises das Fontes de Dados

Para validar a solução proposta nesta dissertação, um exemplo prático no contexto de filmes foi formulado. A solução ExpSOLAP se conecta a duas fontes de dados distintas: uma fonte multidimensional estruturada; e outra fonte de dados semântica representada por arquivos RDF semiestruturados. Na Figura 32 é ilustrado a metodologia do exemplo prático.

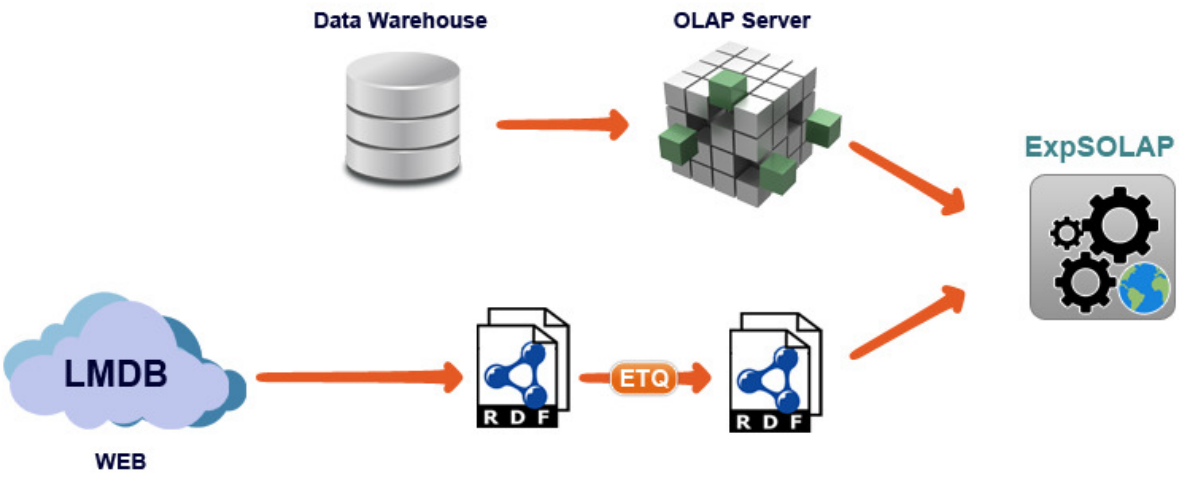

Figura 32: Metodologia do Exemplo Prático

A fonte de dados estruturada é caracterizada por um cubo multidimensional armazenado seguindo a abordagem MOLAP. O DW está representado através do esquema estrela e teve sua criação física no banco de dados *Microsoft SQL Server*. A tabela de fatos está relacionada ao evento mais importante da base de dados – Filmes -, enquanto as informações numéricas (duração do filme, avaliação, bilheteria e número de premiação do tipo *Oscar*) são tratadas como medida por permitir a realização de cálculos (soma, contagem, média, mínimo, máximo). A tabela de fatos também é caracterizada como uma dimensão degenerada

(*degenerate dimension*), visto que possui atributos não numéricos relacionados ao título do filme e a URI. Uma dimensão degenerada é uma dimensão armazenada na tabela de fatos; o que elimina a necessidade de realizar junções entre mais uma tabela (Kimball et al., 2013).

As demais informações relacionadas a um filme (Editor, Ator, Diretor, Idioma, País, Tempo, Escritor, Produtor, Formato e Gênero) estão modeladas como tabelas de dimensão, porque possibilitam uma análise do evento (filme) sob qualquer uma dessas perspectivas.

As dimensões Editor, Ator, Diretor, Gênero, Produtor e Escritor foram modeladas através de um relacionamento N:M, visto que um filme pode ter a participação de mais de um editor, ator, diretor, gênero, produtor ou escritor. Nesse caso, Kimball et al. (2013) sugere que tais dimensões sejam modeladas através de tabelas pontes, responsáveis por armazenar a referência para tabela de fatos e para a tabela de dimensão. Em outras palavras, a ligação da tabela de fatos será com a tabela ponte e não diretamente com a tabela da dimensão. O DW de filmes obtido para esse exemplo prático é ilustrado na Figura 33.

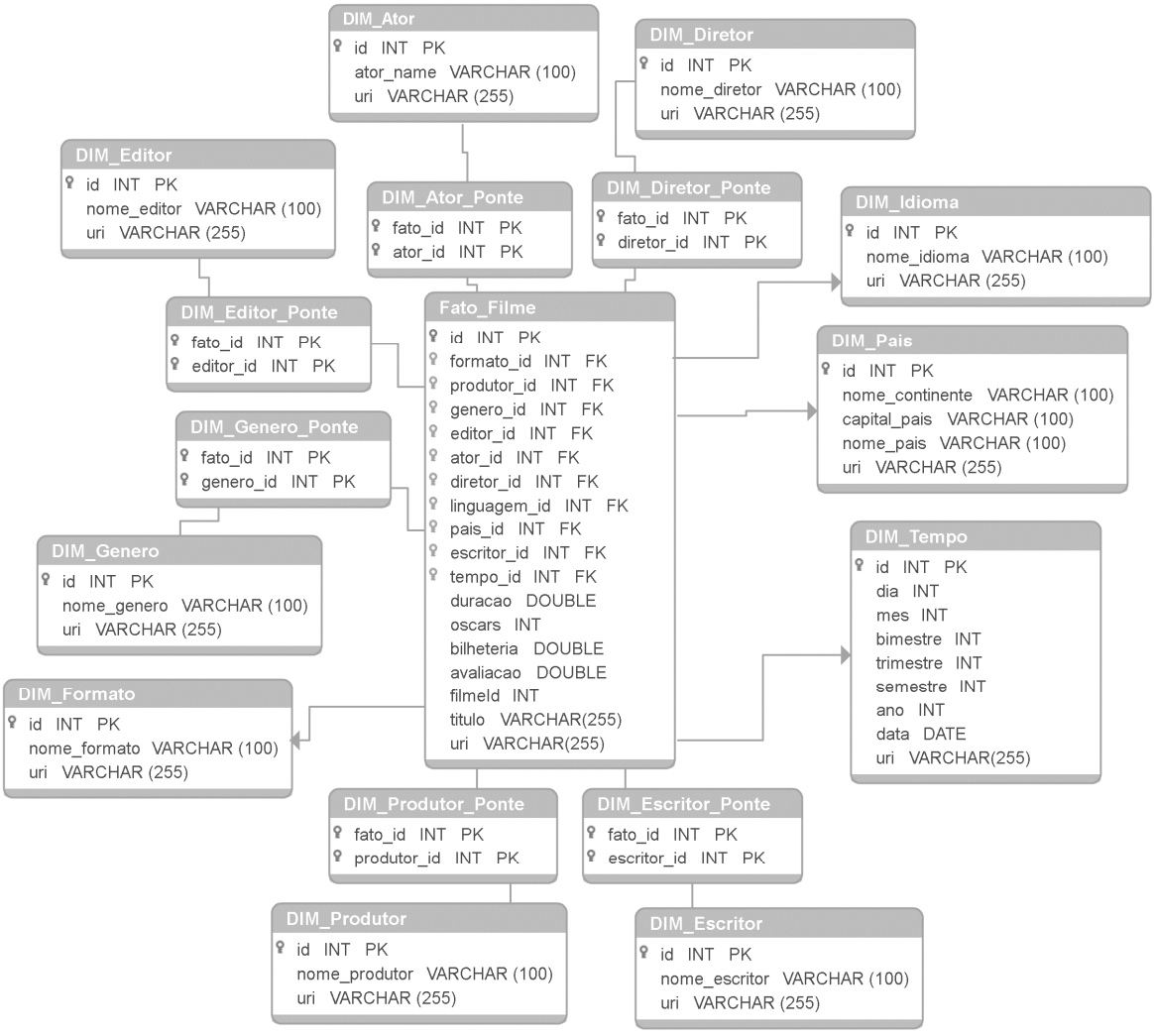

Figura 33: Modelo do DW para cubo de Filmes

Um cubo multidimensional foi construído, através da ferramenta *SQL Server Analysis Server* (SSAS), com base nesse DW. O SSAS permite projetar, criar e gerenciar estruturas multidimensionais que contenham detalhes e dados de agregação de várias fontes de dados, como bancos de dados relacionais, arquivos XML e planilhas, em um único modelo lógico e unificado com suporte para cálculos internos. Fornece análise rápida e interativa de grandes quantidades de dados contidos nesse modelo de dados unificado e que podem ser entregues a usuários em vários idiomas e moedas. O cubo desenvolvido no SSAS foi disponibilizado via XMLA para ser explorado pela solução ExpSOLAP.

A fonte de dados semiestruturada, caracterizada por arquivos RDF, foi obtida através da base de dados pública do *Linked Movie Data Base* (LMDB)[11], um site que publica a base semântica aberta de filmes, incluindo um grande número de *interlinks* para outras bases de dados semânticas (Figura 34) e páginas relacionadas, como, por exemplo, *DBPedia*[12], *IMDB*[13] e *Freebase*[14]. A entidade filme, disponibilizada no LMDB, é caracterizada por conter as seguintes informações: duração, data de lançamento, formato de sua distribuição (DVD, Cinema, etc.), país e idioma em que foi produzido, gênero (comédia, drama, etc.), além de editores, diretores, atores, produtores e escritores que participaram do desenvolvimento do mesmo.

A coleta de dados do LMDB deu-se através de um *crawler*, que coletou, entre 26 de abril e 05 de maio de 2015, 98.816 arquivos RDF relacionados a filmes disponíveis no referido site. Destes, 1.347 arquivos apresentaram alguma inconsistência. Por algum motivo, seu conteúdo é inexistente ou foi corrompido. Os arquivos restantes foram categorizados em três categorias:

- **Completo**: Composto por triplas correspondentes a todas as informações sobre o filme (duração, data de lançamento, formato de distribuição, atores, editor etc.), como, por exemplo: <*filmeId, movie:runtime, 119*>;
- **Incompleto**: Composto por, pelo menos, uma tripla relacionada à alguma informação de filme; e
- **Cópia**: Composto por uma única tripla que referencia algum *link* externo para maiores informações a respeito do filme, como por exemplo: <*filmeId, foaf:page, URL*>.

[11] http://data.linkedmdb.org/
[12] http://wiki.dbpedia.org/
[13] http://www.imdb.com/
[14] http://www.freebase.com/

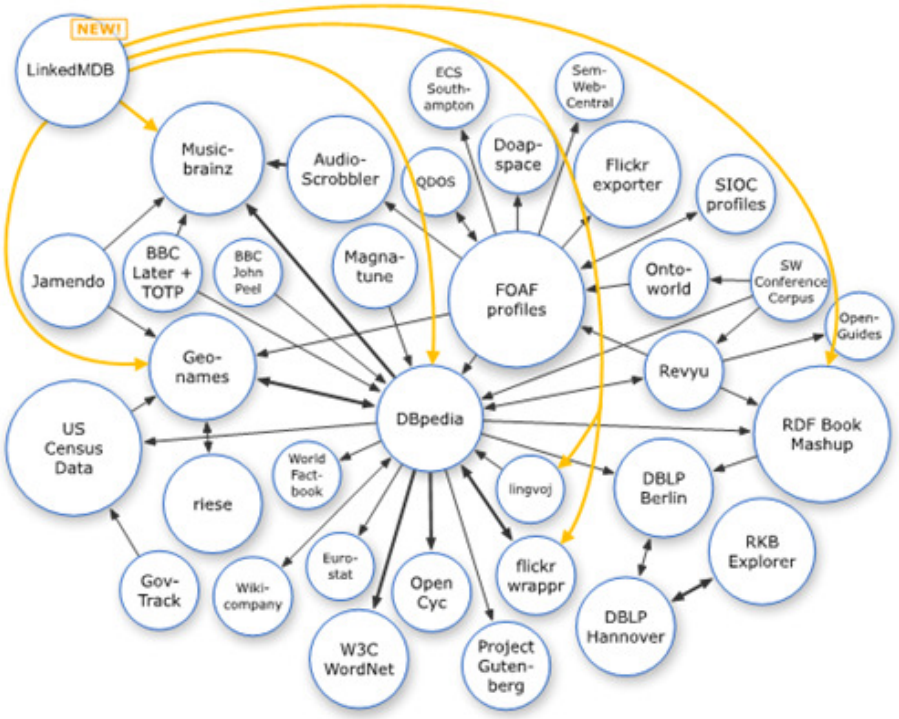

Figura 34: LMDB *Interlinks* de Ontologias. Fonte: http://wiki.linkedmdb.org/Main/Interlinking

Os arquivos RDF do tipo Cópia não apresentam padrão no link externo presente na tripla. Além disso, na maioria das vezes, redirecionam para uma página na qual a informação não é inteligível por máquinas. Nesse sentido, os arquivos RDF do tipo Cópia, no total de 11.849, foram descartados do experimento. Assim, a quantidade efetiva de arquivos RDF utilizados no nosso exemplo prático foi de 85.620 arquivos (em espaço de disco, 425 MB) e estão categorizados em Completo ou Incompleto. Na Figura 35, é apresentado um gráfico contendo a proporção de dados LMDB obtidos pelo *crawler*.

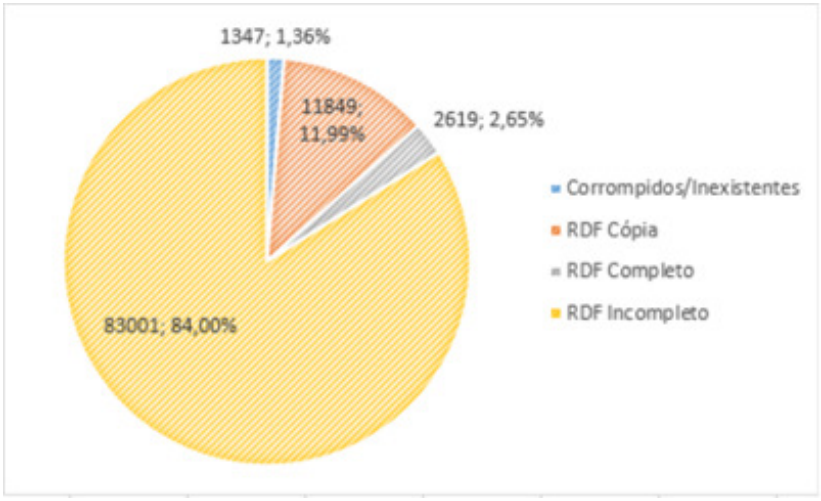

Figura 35: Proporção dos dados coletados do LMDB

Os arquivos RDF que compõem a base semântica também precisam ser expostos a um processo similar a um ETC. Esse processo, definido por Abelló et al. (2015) como ETQ (*Extract, Transform* and *Query*), é diferente do processo ETC tradicional visto que os dados serão transformados para serem consultados sob demanda e de forma virtual. Na subseção seguinte serão detalhadas as transformações adicionais realizadas na fonte de dados semiestruturada.

5.1.1 Fonte Semiestruturada Semântica: Processo de ETQ nos arquivos RDF

As modificações realizadas nos arquivos RDF que compõem a base semântica foram simplificadas, e objetivam deixar as informações presente nos arquivos mais consistentes e correlacionadas às informações contidas no cubo multidimensional. Tais modificações consistiram na adição de novas triplas aos arquivos RDF, isso porque os objetos das triplas dos arquivos (equivalente aos valores das dimensões) estão representados através de *resources* (Código Fonte 14). Embora a utilização de URI como nome de coisas em triplas RDF seja um dos princípios que norteiam a *Linked Data*, com o intuito de facilitar a descoberta e reutilização da informação, a exibição da informação nesse formato não é interessante em ambientes analíticos. Esse tipo de ambiente, típico para usuários tomadores de decisão, requer a informação mais simples e representativa possível, e não um link para que o usuário consuma informação em um local externo. A informação tem que estar disponível na própria ferramenta.

Código Fonte 14: Exemplificação das triplas presentes nos arquivos RDF

```
1   <rdf:Description
    rdf:about="http://data.linkedmdb.org/resource/film/55843">
2      <movie:writer
       rdf:resource="http://data.linkedmdb.org/resource/writer/14850"
       />
3      <movie:producer
       rdf:resource="http://data.linkedmdb.org/resource/producer/14158"
       />
4      <movie:actor
       rdf:resource="http://data.linkedmdb.org/resource/actor/30728" />
5      <movie:runtime>113</movie:runtime>
6      <dc:date>2009-11-13</dc:date>
7   ...
8
    </rdf:Description>
9   ...
```

Nesse sentido, foi desenvolvido um processo de transformação, utilizando o arcabouço *Apache Jena*, que percorreu todos os arquivos RDF, interpretando cada tripla e adicionando novas triplas correspondentes às propriedades, porém com valores literais ao invés de recursos.

Adicionalmente, os arquivos RDF foram complementados com triplas referentes às medidas presente no cubo e ausentes na fonte semântica (avaliação, bilheteria e número de *Oscars*). A inserção dessas triplas seguiu uma metodologia aleatória. No caso da medida avaliação, foram alocados valores variando entre 0.0 e 10.0; para bilheteria, valores variando entre R$ 1.000.000,00 e R$ 99.000.000,00; e entre 1 e 10 para número de premiações de *Oscar*. Nesse último caso, apenas 1.000 arquivos RDF foram contemplados, visto que nem todo filme é premiado.

De forma análoga, os arquivos RDF foram incorporados com triplas que destrincham a informação temporal da data de lançamento do filme, com a adição de predicados relativos ao ano, mês e dia. Alguns arquivos RDF continham a informação do ano do lançamento do filme; no entanto, essa informação não estava definida em um predicado específico e, na maioria das ocorrências, estava registrada juntamente com a data do lançamento do filme, no mesmo predicado (<*movie:1479, movie:initial_release_date, "1989,1989-11-03"*>). Para padronizar as informações, novas triplas foram inseridas extraindo informação da data completa do lançamento do filme.

Nesses últimos dois casos, da inserção das triplas referentes às medidas e à informação temporal, a inserção das triplas foi possível com a criação de duas novas ontologias (utilizando a ferramenta *Protégé*), denominadas, respectivamente, de *movie* (Apêndice A) e *time* (Apêndice B), esta última desenvolvida como uma extensão da ontologia owl-time[15].

Essas transformações possibilitaram deixar ambas as fontes de dados, a estruturada e semiestruturada semântica, correlacionadas, ou seja, o mesmo tipo de informação encontrado em uma estará disponível na outra. Entretanto, não foi possível identificar nos arquivos RDF a existência de triplas cujo predicado é equivalente a alguma dimensão, impossibilitando a inserção de triplas com um valor *default* (*Unknown*).

Exemplificando o processo ETQ realizado nos arquivos RDF da base semântica, as triplas exibidas no Código Fonte 15 ilustram a nova configuração do arquivo RDF exibido no Código Fonte 14, acrescido das novas triplas inseridas.

[15] http://www.w3.org/TR/owl-time/

Código Fonte 15: Nova configuração do arquivo RDF

```
1   <rdf:Description
    rdf:about="http://data.linkedmdb.org/resource/film/55843">
2      <movie:writer
       rdf:resource="http://data.linkedmdb.org/resource/writer/14850"
       />
3      <movie:producer
       rdf:resource="http://data.linkedmdb.org/resource/producer/14158"
       />
4      <movie:actor
       rdf:resource="http://data.linkedmdb.org/resource/actor/30728" />
5      <movie:runtime>113</movie:runtime>
6      <dc:date>2009-11-13</dc:date>
7      <time:year>2009</time:year>
8      <time:month>11</time:month>
9      <time:day>13</time:day>
10     <film:rating>8.1</film:rating>
11     <film:box_office>9814202.883</film:box_office>
12     <film:oscars>0</film:oscars>
13  ...
14  </rdf:Description>
15
16  <rdf:Description
    rdf:about="http://data.linkedmdb.org/resource/writer/14850">
17     <movie:writer_name>Roland Emmerich</movie:writer_name>
18  </rdf:Description>
19
20  <rdf:Description
    rdf:about="http://data.linkedmdb.org/resource/actor/14158">
21     <movie:producer_name>Larry J. Franco</movie:producer_name>
22  </rdf:Description>
23
24  <rdf:Description
    rdf:about="http://data.linkedmdb.org/resource/actor/30728">
25     <movie:actor_name>Woody Harrelson</movie:actor_name>
26  </rdf:Description>
27  ...
```

5.2 Exemplo Prático aplicado ao contexto de Filmes

Nesta seção, é apresentado um exemplo prático proposto no qual a solução ExpSOLAP explora o cubo de filme disponibilizado, via XMLA, pela ferramenta SSAS e os arquivos semiestruturados semânticos, com auxílio do *Framework Apache Jena.* Como o servidor de cubos SSAS não oferece suporte a dados espaciais, os membros do nível *Nome Continente* e *Nome País*, da *Dimensão País*, foram geocodificados. Medidas espaciais não foram utilizadas nesse exemplo.

5.2.1 Consultas

As consultas a seguir apresentam – e validam – a análise convencional e espacial, possibilitada pela solução ExpSOLAP na exploração de ambas as fontes de dados, estruturada multidimensional e semiestruturada semântica. As consultas foram classificadas em duas categorias: consultas exploratórias OLAP e consultas exploratórias SOLAP. Ambas as categorias são exploratórias porque exploram dados de fonte de dados heterogêneas. Enquanto a primeira categoria foca em análises considerando apenas dados convencionais, fazendo uso de operadores OLAP (*drill-down*, *roll-up*, etc.), a categoria exploratória SOLAP realiza análise espacial utilizando operadores SOLAP (*spatial drill down*, *spatial roll-up*, etc.).

Oito consultas foram elaboradas; sendo as três primeiras exploratórias OLAP e as cinco últimas exploratória SOLAP. Com o intuito de deixar a leitura mais limpa e menos repetitiva, os prefixos das consultas geradas estão listados no Código Fonte 16.

Código Fonte 16: Prefixos utilizados nas consultas SPARQL geradas pela solução ExpSOLAP

```
1   PREFIX movie: <http://data.linkedmdb.org/resource/movie/>
2   PREFIX dbpedia: <http://dbpedia.org/property/>
3   PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4   PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
5   PREFIX foaf: <http://xmlns.com/foaf/0.1/>
6   PREFIX   d2r:   <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-
    server/config.rdf#>
7   PREFIX owl: <http://www.w3.org/2002/07/owl#>
8   PREFIX dc: <http://purl.org/dc/terms/>
9   PREFIX oddlinker: <http://data.linkedmdb.org/resource/oddlinker/>
10  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
11  PREFIX db: <http://data.linkedmdb.org/resource/>
12  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
13  PREFIX time: <http://www.w3.org/2006/time#>
14  PREFIX timeExt: <http://150.165.75.163/2015/10/time#>
15  PREFIX film: <http://150.165.75.163/2015/10/movie#>
```

Consulta 1

"*Exiba o somatório da duração dos filmes e do número de oscars, agrupados por gênero e formato de distribuição dos mesmos*".

A execução desta consulta consistiu em adicionar as medidas *Duração* e *Oscars* na divisória "Colunas", e os membros *Nome Gênero* e *Nome Formato* na divisória "Linhas". As medidas serão agregadas conforme a função de agregação previamente informada no cadastro de fonte semântica; no caso das medidas *Duração* e *Oscars*, a função de agregação será o

somatório. Os resultados são calculados automaticamente no momento em que se adiciona membros na divisória, e são exibidos em uma tabela, conforme ilustrado na Figura 36, enquanto no Código Fonte 17 é ilustrada a consulta SPARQL resultante da consulta, gerada pelo tradutor desenvolvido. O objetivo desta consulta é exemplificar a navegação em medidas e dimensões, assim como na visualização tabular.

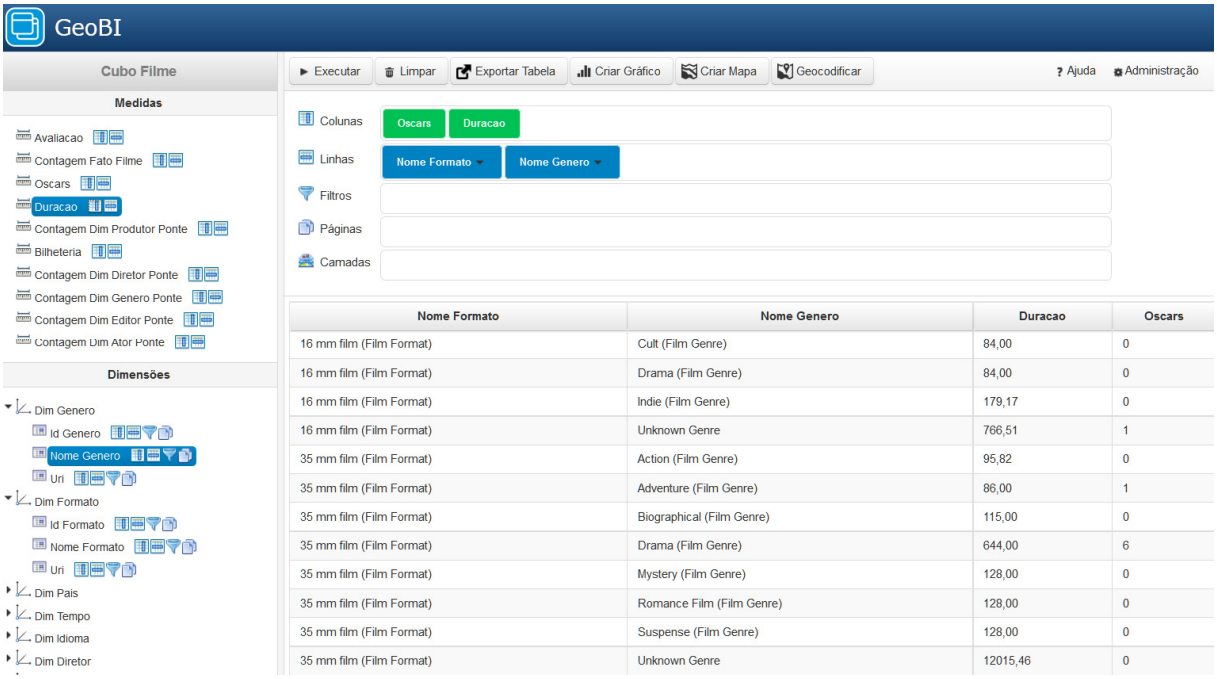

Figura 36: Consulta 1 - Resultados em formato tabular

Código Fonte 17: Consulta 1 - Consulta SPARQL gerada

```
1    <prefixos...>
2
3    SELECT ?nomegenero ?nomeformato (SUM(xsd:decimal(?duracao)) AS
     ?duracao_sum) (SUM(xsd:integer(?oscars)) AS ?oscars_sum)
4    WHERE
5    {
6       ?filme_uri_suj movie:genre ?dimgenero_uri .
7       ?dimgenero_uri movie:film_genre_name ?nomegenero .
8       ?filme_uri_suj movie:film_format ?dimformato_uri .
9       ?dimformato_uri movie:film_format_name ?nomeformato .
10      ?filme_uri_suj movie:runtime ?duracao .
11      ?filme_uri_suj film:oscars ?oscars .
12   }
13   GROUP BY ?nomegenero ?nomeformato
```

Consulta 2

"*Qual a soma dos valores obtidos pela bilheteria dos filmes produzidos, por mês, da categoria 'Drama'? Exiba um gráfico de barras com esses valores*".

Nesta consulta, três divisórias foram exploradas. A medida *Bilheteria* foi adicionada à divisória "Colunas" e o nível *Nome Mês,* da *Dimensão Tempo*, à divisória "Linhas". Na divisória "Filtros" foi adicionado o nível *Nome Gênero,* o qual foi selecionado apenas o membro *Drama*, através do filtro convencional (Figura 37). O resultado, em formato tabular, pode ser visualizado na Figura 38, enquanto o resultado, em formato gráfico, está disponível na Figura 39. No Código Fonte 18 é ilustrada a consulta SPARQL resultante. O objetivo desta consulta é exemplificar a utilização da divisória "Filtros".

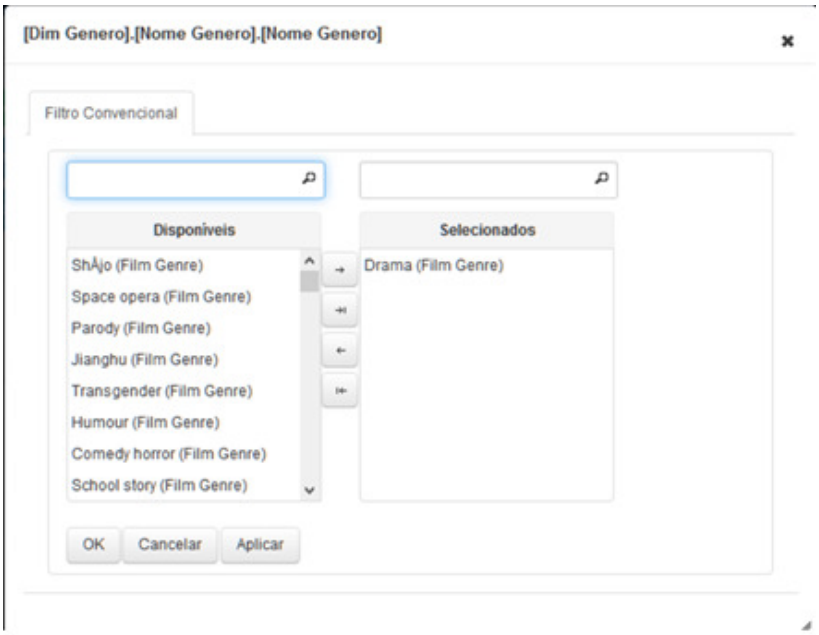

Figura 37: Consulta 2 - Filtro Convencional

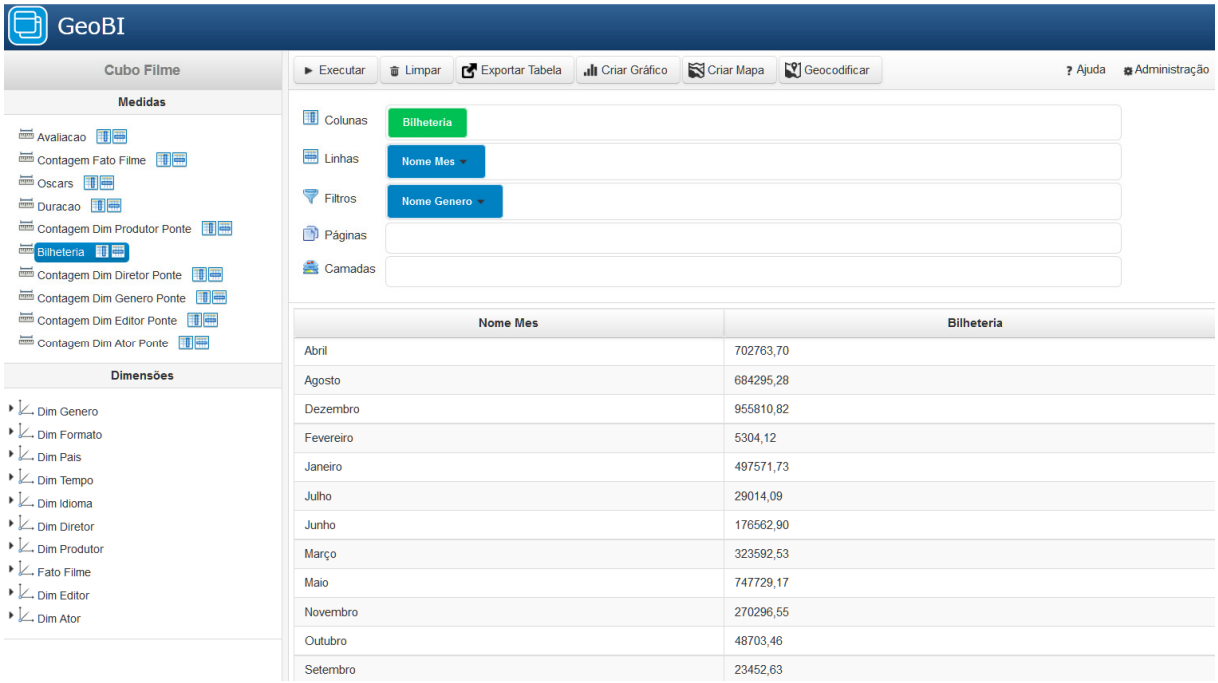

Figura 38: Consulta 2 - Resultados em formato tabular

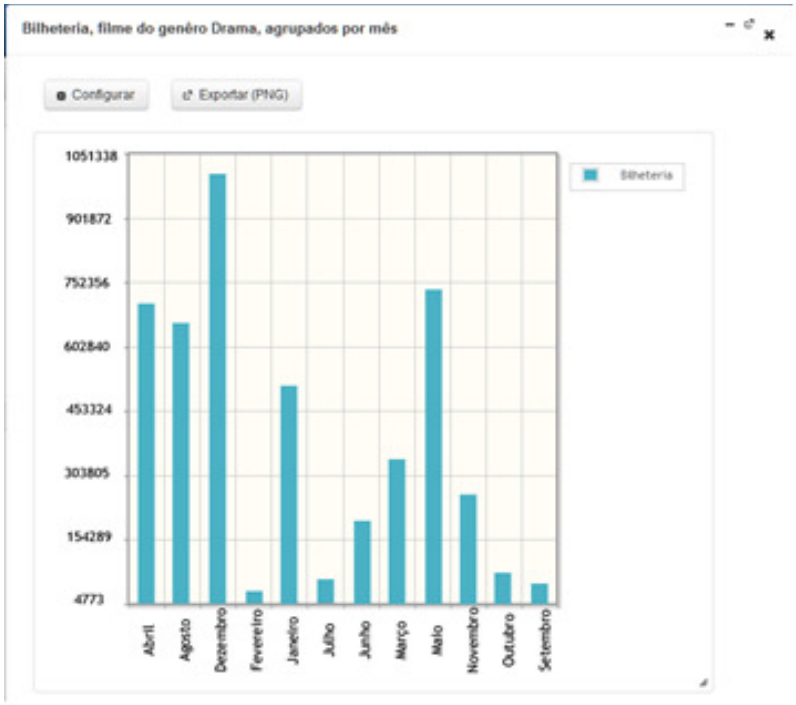

Figura 39: Consulta 2 - Resultados em formato gráfico

Código Fonte 18: Consulta 2 - Consulta SPARQL gerada

```
1    <prefixos...>
2
3    SELECT       ?nomemes       (AVG(xsd:decimal(?bilheteria))       AS
     ?bilheteria_avg)
4    WHERE {
5       ?filme_uri_suj time:month ?nomemes .
6       ?filme_uri_suj film:box_office ?bilheteria .
7       ?filme_uri_suj movie:genre ?dimgenero_uri .
8       ?dimgenero_uri movie:film_genre_name ?nomegenero .
9       FILTER (regex(?nomegenero, "^Drama", "i"))}
10   GROUP BY ?nomemes
```

Consulta 3

"*Exiba, por páginas, a avaliação de cada filme*".

Para exemplificar a paginação, a medida *Avaliação* foi adicionada à divisória "Colunas", enquanto o membro *Nome do Filme,* foi adicionado à divisória "Páginas". O resultado, em formato tabular e paginado, pode ser visualizado nas Figura 40 e Figura 41. No Código Fonte 19 é ilustrada a consulta SPARQL resultante.

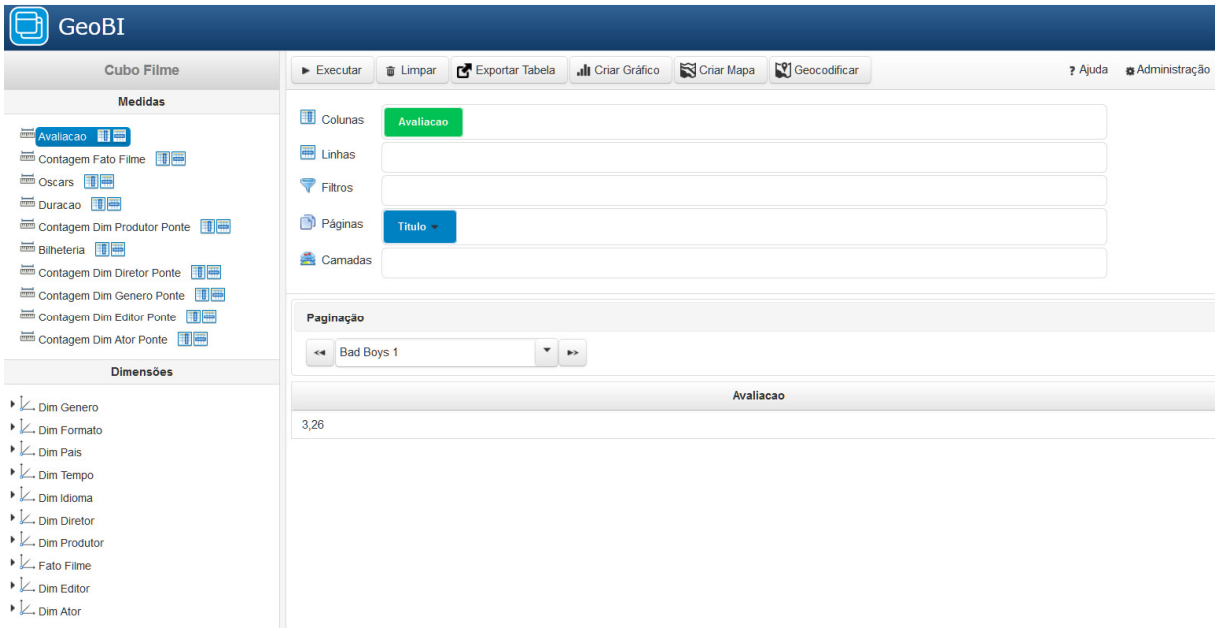

Figura 40: Consulta 3 - Resultados paginados 1

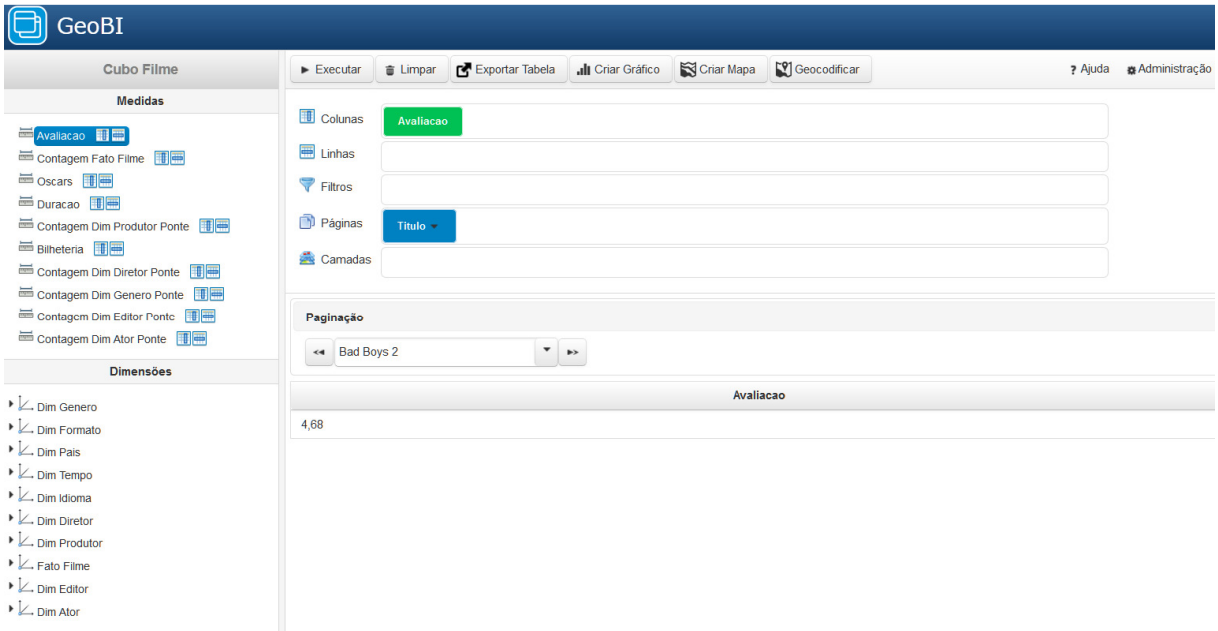

Figura 41: Consulta 3 - Resultados paginados 2

Código Fonte 19: Consulta 3 - Consulta SPARQL gerada

```
1    <prefixos...>
2
3    SELECT ?titulo (AVG(xsd:decimal(?avaliacao)) AS ?avaliacao_avg)
4    WHERE
5    {
6       ?filme_uri_suj dc:title ?titulo .
7       ?filme_uri_suj film:rating ?avaliacao .
8    }
9    GROUP BY ?titulo
```

Consulta 4

"*Exiba no mapa a duração dos filmes e detalhe por continente e país*".

Nesta consulta, a medida *Duração* foi adicionada à divisória "Colunas", e a hierarquia *Continente – País*, da *Dimensão País*, à divisória "Camadas". Ao adicionar a hierarquia à divisória, o nível menos detalhado da hierarquia é exibido junto com o ícone de navegação +, que é utilizado para aumentar o nível de detalhes (Figura 42). Nesse nível já é possível realizar uma análise tabular e espacial dos dados (Figura 43). Para detalhar os dados, o ícone + foi utilizado para analisar dados do nível *País – spatial drill down*, novamente de forma tabular (Figura 44) e espacial (Figura 45). Se a geometria não tiver um valor de medida associado, ela não será exibida no mapa. No Código Fonte 20 é ilustrada a consulta SPARQL resultante do último nível explorado da hierarquia. O objetivo desta consulta é de exemplificar a navegação nas hierarquias e a visualização espacial.

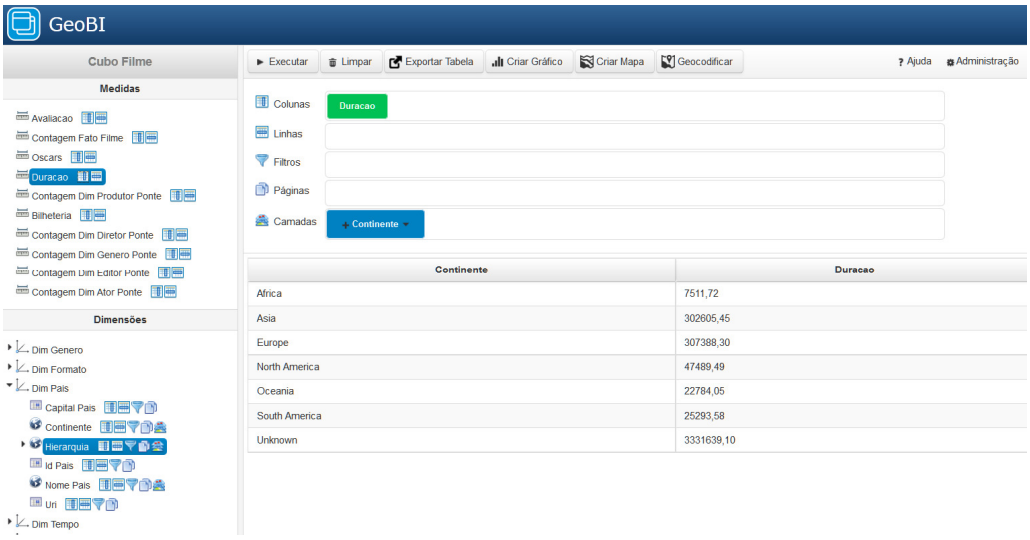

Figura 42: Consulta 4 - Resultado com hierarquia retraída

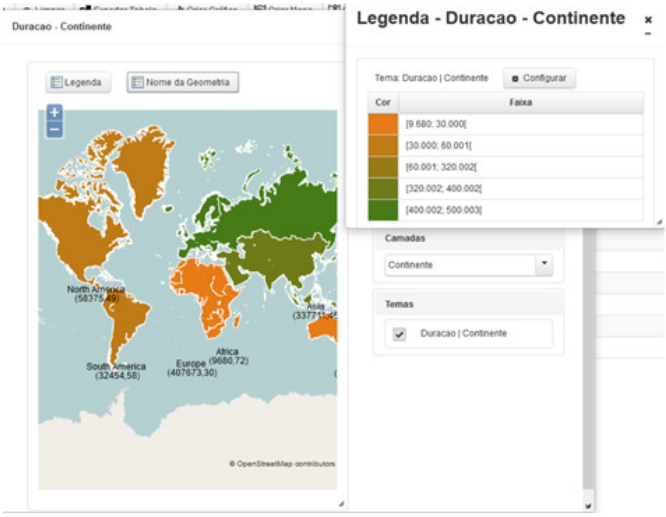

Figura 43: Consulta 4 - Visualização Espacial dos resultados com hierarquia retraída

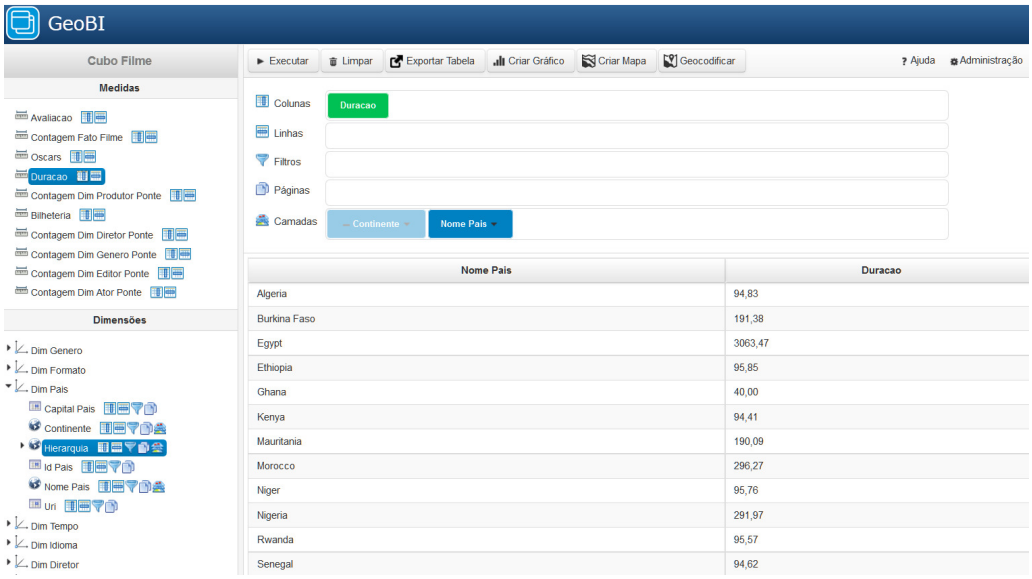

Figura 44: Consultas 4 - Resultados com a hierarquia expandida (*drill-down*)

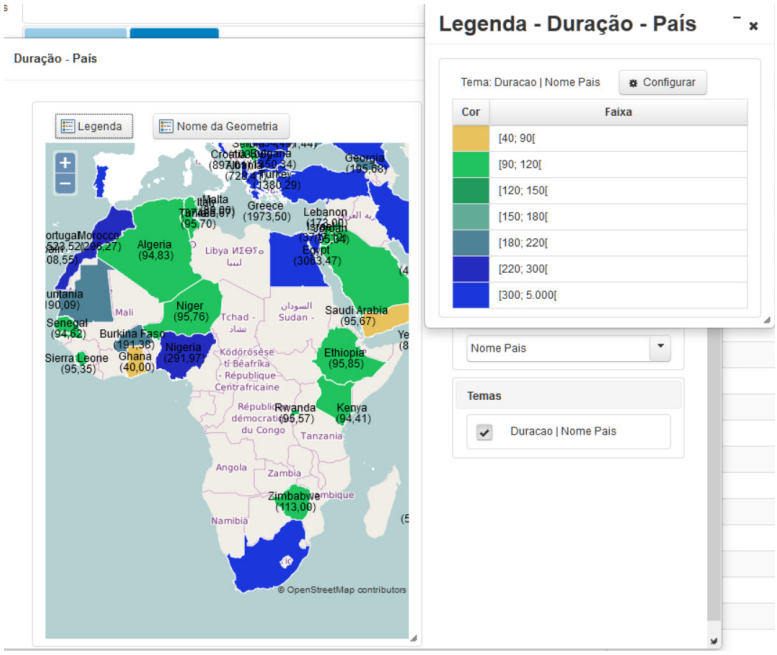

Figura 45: Consulta 4 - Visualização Espacial dos resultados com hierarquia expandida (*spatial drill-down*)

Código Fonte 20: Consulta 4 - Consulta SPARQL gerada

```
1   <prefixos...>
2
3   SELECT ?nomepais (SUM(xsd:decimal(?duracao)) AS ?duracao_sum)
4   WHERE
5   {
6      ?filme_uri_suj movie:country ?dimpais_uri .
7      ?dimpais_uri movie:country_name ?nomepais .
8      ?filme_uri_suj movie:runtime ?duracao .
9   }
10  GROUP BY ?nomepais
```

Consulta 5

"*Exiba no mapa valor o médio das avaliações dos filmes produzidos em países que fazem fronteira com os Estados Unidos*".

Essa consulta demonstra a utilização dos filtros espaciais disponíveis na solução ExpSOLAP. A medida *Avaliação* foi adicionada à divisória "Colunas" e o nível *Nome País* à divisória "Camadas". Os membros do nível *Nome País* foram filtrados através do filtro geográfico com operador *Touches*, utilizado, neste exemplo, para filtrar os países vizinhos aos Estados Unidos (Figura 46). A visualização dos resultados pode ser em formato tabular (Figura 47) ou espacial (Figura 48). No Código Fonte 21 é ilustrada a consulta SPARQL resultante.

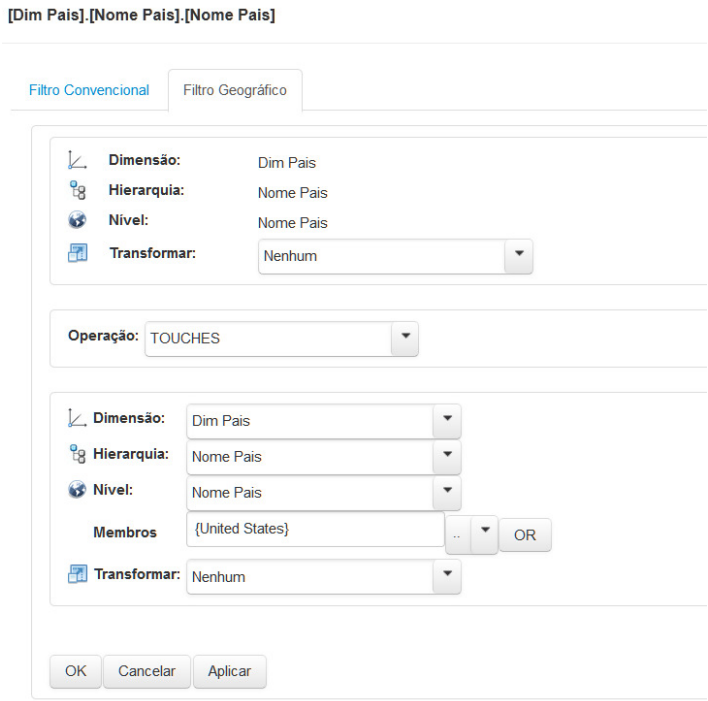

Figura 46: Consulta 5 - Filtro geográfico (operador *touches*) aplicado

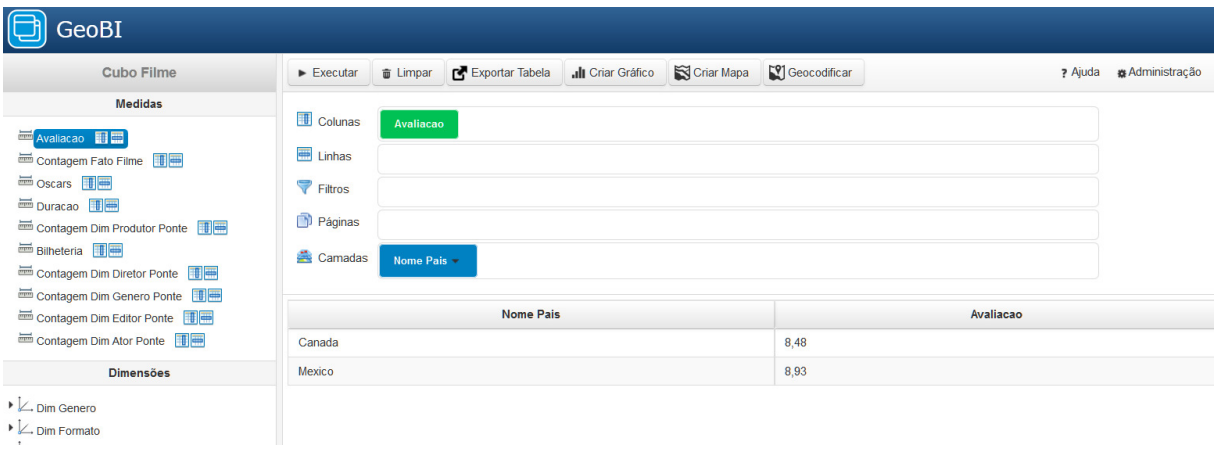

Figura 47: Consulta 5 - Resultados em formato tabular

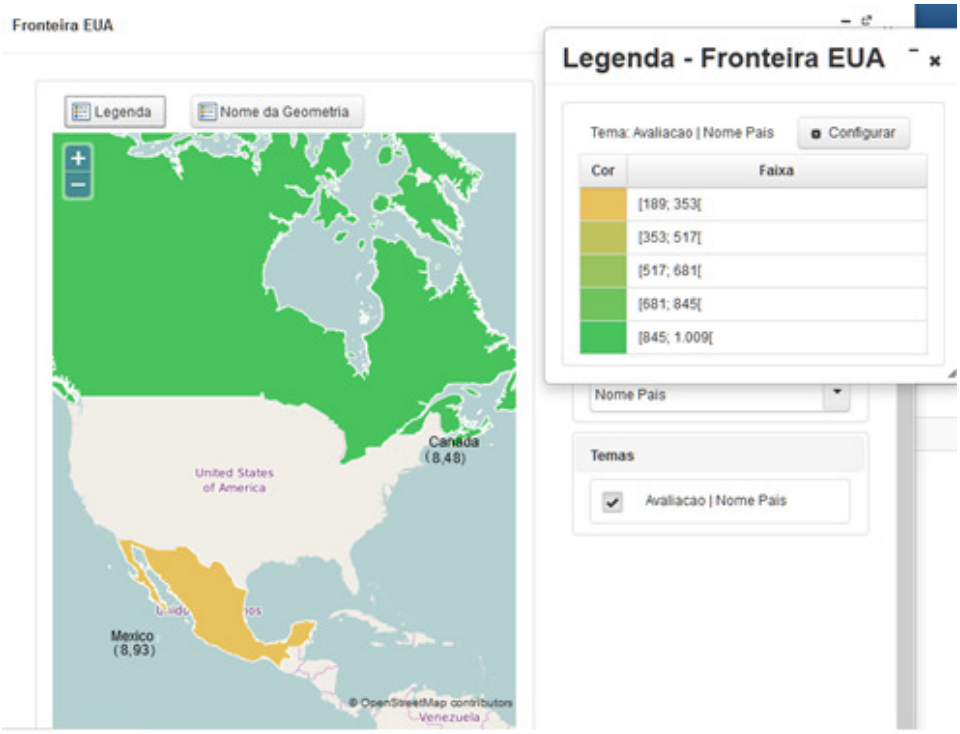

Figura 48: Consulta 5 - Visualização espacial dos resultados

Código Fonte 21: Consulta 5 - Consulta SPARQL gerada

```
1    <prefixos...>
2
3    SELECT ?nomepais (AVG(xsd:decimal(?avaliacao)) AS ?avaliacao_avg)
4    WHERE
5    {
6       ?filme_uri_suj movie:country ?dimpais_uri .
7       ?dimpais_uri movie:country_name ?nomepais .
8       ?filme_uri_suj film:rating ?avaliacao .
9       FILTER (regex(?nomepais, "^Mexico", "i")||
        regex(?nomepais, "^Canada", "i"))
10   }
11   GROUP BY ?nomepais
```

Consulta 6

"*Qual a soma dos valores obtidos pela bilheteria dos filmes produzidos no continente europeu?* ".

Essa consulta demonstra a diversidade dos operadores espaciais disponíveis na solução ExpSOLAP. A medida *Bilheteria* foi adicionada à divisória "Colunas" e o nível *Nome País* à divisória "Camadas". No entanto, diferentemente da consulta anterior, o filtro geográfico foi formulado utilizando diferentes níveis espaciais da *Dimensão País*. Os membros do nível *Nome País* foram filtrados, espacialmente, com base nos membros do nível *Continente*. O operador *Within* foi utilizado para filtrar os países que estão contidos no continente europeu

(Figura 49). A visualização dos resultados pode ser em formato tabular (Figura 50) ou espacial (Figura 51). No Código Fonte 22 é ilustrada a consulta SPARQL resultante.

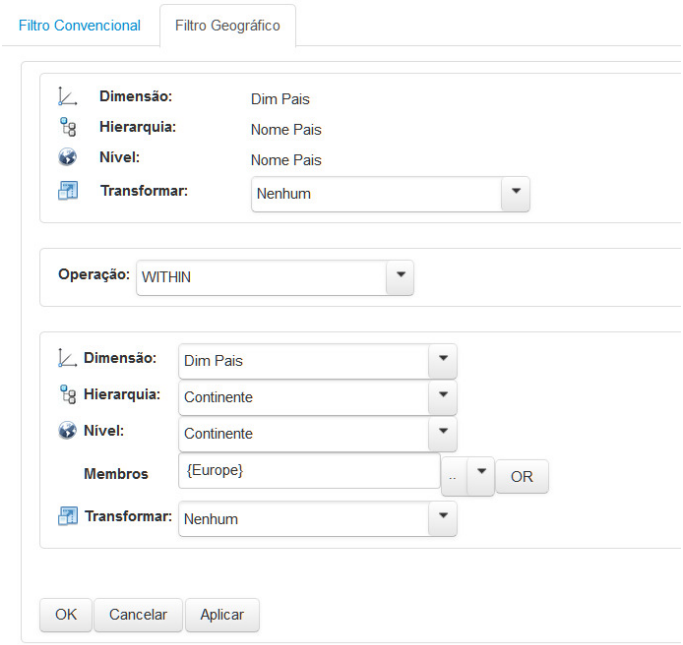

Figura 49: Consulta 6 - Filtro Geográfico (operador *within*) aplicado

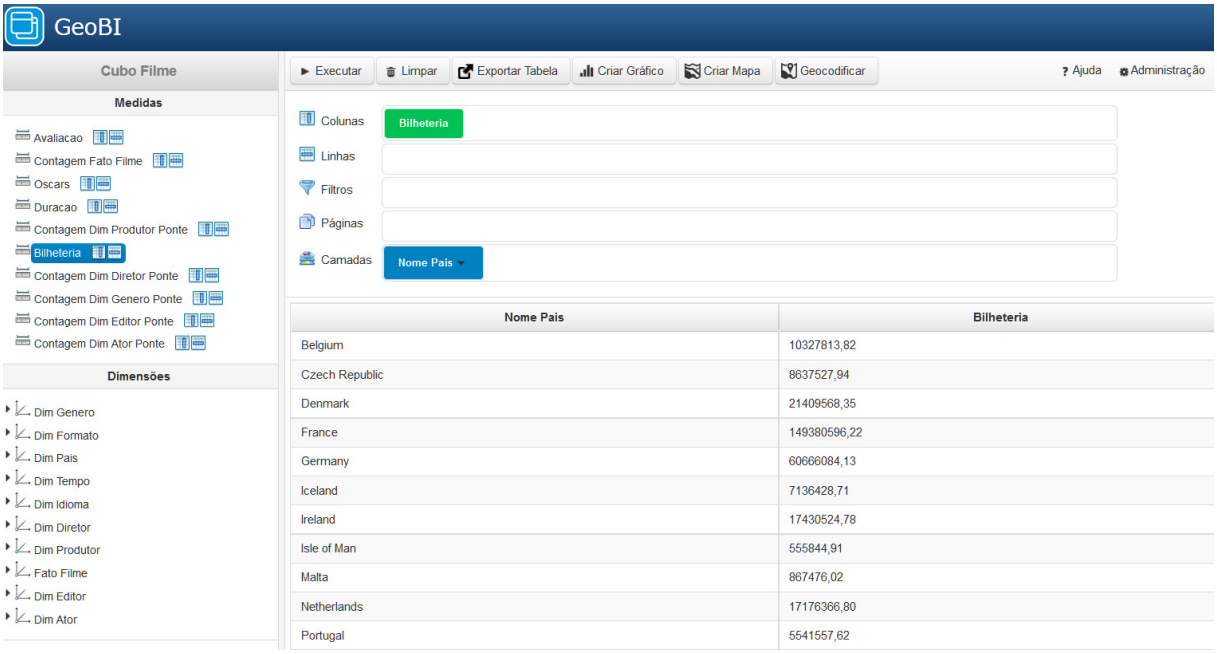

Figura 50: Consulta 6 - Resultados em formato tabular

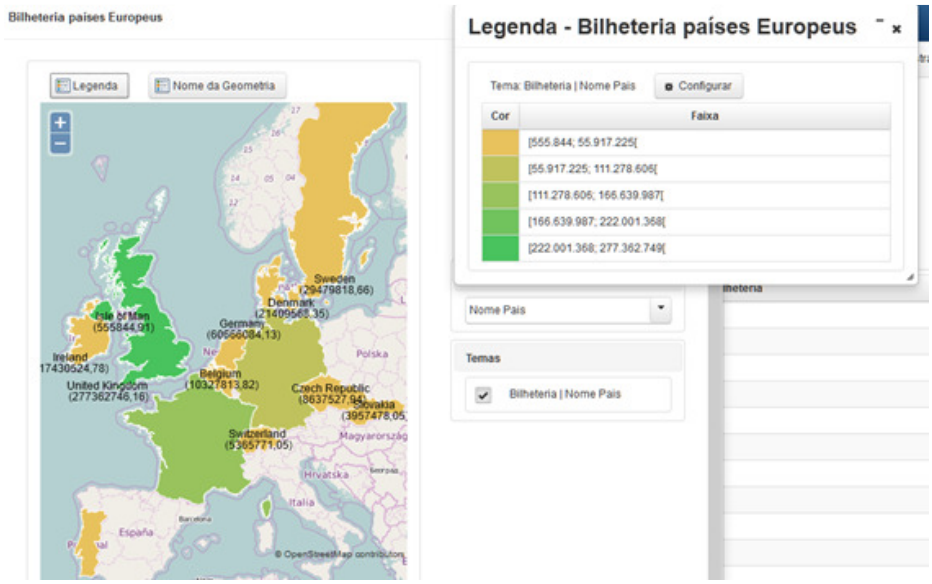

Figura 51: Consulta 6 - Visualização espacial dos resultados

Código Fonte 22: Consulta 6 - Consulta SPARQL gerada

```
1   <prefixos...>
2
3   SELECT ?nomepais (AVG(xsd:decimal(?bilheteria)) AS
    ?bilheteria_avg)
4   WHERE
5   {
6      ?filme_uri_suj movie:country ?dimpais_uri .
7      ?dimpais_uri movie:country_name ?nomepais .
8      ?filme_uri_suj film:box_office ?bilheteria .
9      FILTER (regex(?nomepais, "^Netherlands", "i")||
       regex(?nomepais, "^Denmark", "i")|| regex(?nomepais, "^United
    Kingdom", "i")|| regex(?nomepais, "^Sweden", "i")||
    regex(?nomepais, "^Bosnia and Herzegovina", "i")||
    regex(?nomepais, "^Guernsey", "i")||
       regex(?nomepais, "^France", "i")|| regex(?nomepais,
    "^Gibraltar", "i")|| regex(?nomepais, "^Germany", "i")||
    regex(?nomepais, "^Vatican", "i")|| regex(?nomepais, "^Belgium",
    "i")|| regex(?nomepais, "^Liechtenstein", "i")|| regex(?nomepais,
    "^Faroe Islands", "i")|| regex(?nomepais, "^Czech Republic",
    "i")||
       regex(?nomepais, "^Cyprus", "i")|| regex(?nomepais, "^Isle of
    Man", "i")|| regex(?nomepais, "^Portugal", "i")||
    regex(?nomepais, "^Switzerland", "i")|| regex(?nomepais,
    "^Slovenia", "i")||
       regex(?nomepais, "^Aland Islands", "i")|| regex(?nomepais,
    "^Iceland", "i")|| regex(?nomepais, "^Ireland", "i")||
    regex(?nomepais, "^Slovakia", "i")|| regex(?nomepais, "^Malta",
    "i")|| regex(?nomepais, "^Jersey", "i"))
10  }
11  GROUP BY ?nomepais
```

Consulta 7

"*Qual a soma da duração dos filmes produzidos fora do continente europeu?* ".

Essa consulta demonstra quase que o oposto da consulta anterior. A medida *Duração* foi adicionada à divisória "Colunas" e o nível *Nome País* à divisória "Camadas". O filtro geográfico foi formulado utilizando diferentes níveis espaciais da *Dimensão País*. Os membros do nível *Nome País* foram filtrados, espacialmente, com base nos membros do nível *Continente*. O operador *Disjoint* foi utilizado para filtrar os países que não estão contidos, nem fazem fronteira ou interseção com o continente europeu (Figura 52). A visualização dos resultados pode ser em formato tabular (Figura 53) ou espacial (Figura 54). No Código Fonte 23 é ilustrada a consulta SPARQL resultante.

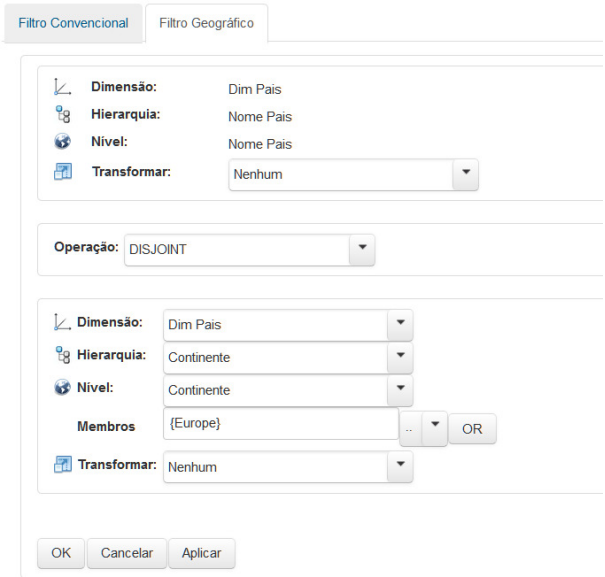

Figura 52: Consulta 7 - Filtro Geográfico (operador *disjoint*) aplicado

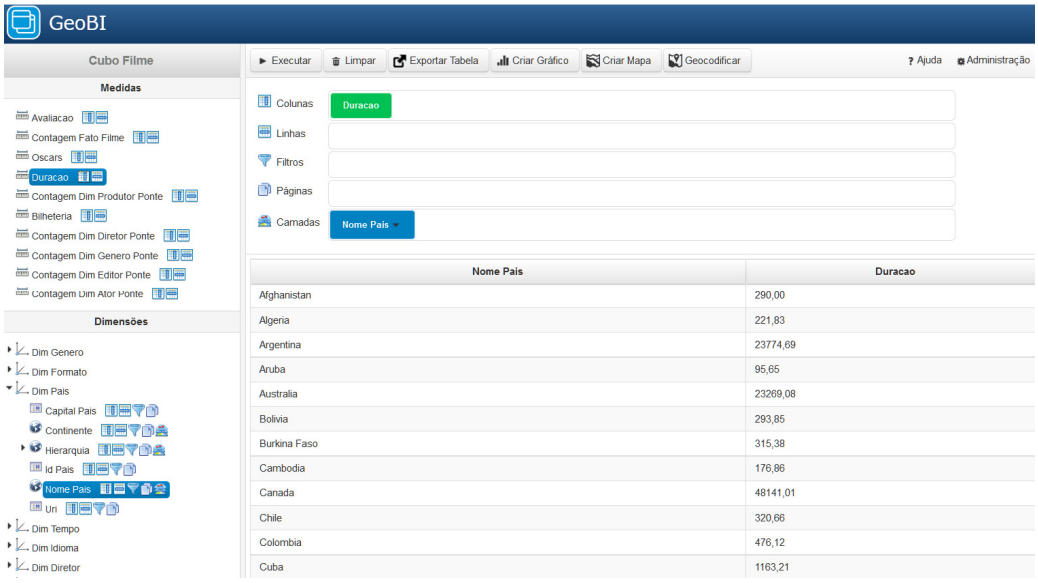

Figura 53: Consulta 7 - Resultados em formato tabular

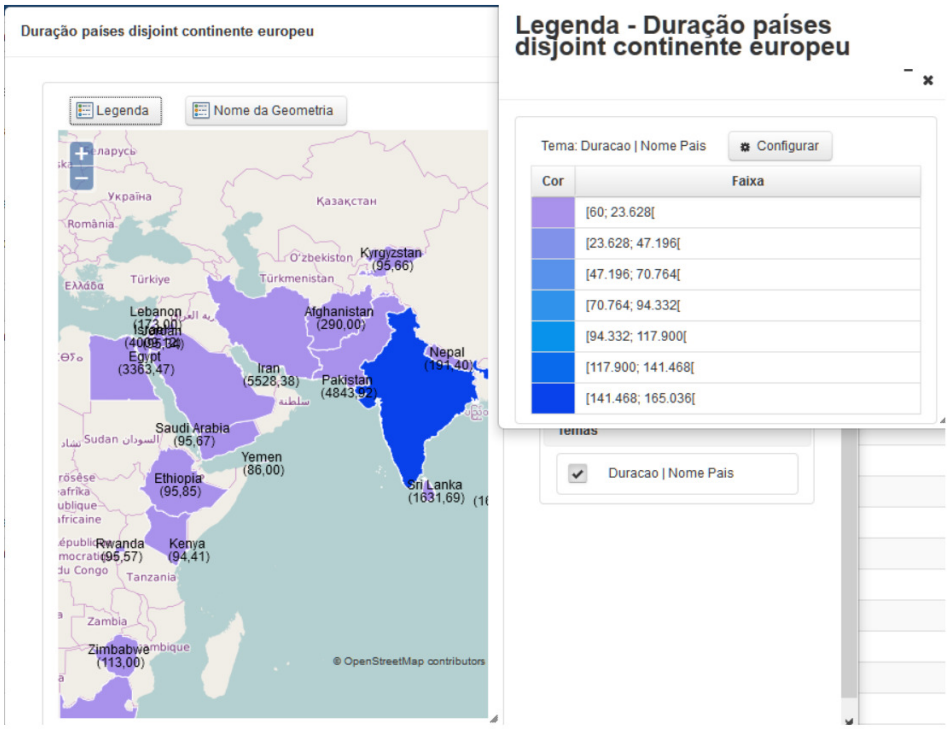

Figura 54: Consulta 7 - Visualização espacial dos resultados

Código Fonte 23: Consulta 7 - Consulta SPARQL gerada

```
1    <prefixos...>
2
3    SELECT ?nomepais (SUM(xsd:decimal(?duracao)) AS ?duracao_sum)
4    WHERE
5    {
6       ?filme_uri_suj movie:country ?dimpais_uri .
7       ?dimpais_uri movie:country_name ?nomepais .
8       ?filme_uri_suj movie:runtime ?duracao .
9       FILTER ((regex(?nomepais, "^Kuwait", "i")|| regex(?nomepais,
     "^Senegal", "i")|| regex(?nomepais, "^Taiwan", "i")||
     regex(?nomepais, "^United Arab Emirates", "i")|| regex(?nomepais,
     "^Yemen", "i")|| regex(?nomepais, "^Vanuatu", "i")||
     regex(?nomepais, "^Gabon", "i")|| regex(?nomepais, "^Australia",
     "i")|| regex(?nomepais, "^Philippines", "i")|| regex(?nomepais,
     "^Congo - Kinshasa", "i")|| regex(?nomepais, "^Indonesia", "i")||
     regex(?nomepais, "^Mexico", "i")|| regex(?nomepais, "^Iraq",
     "i")|| regex(?nomepais, "^Oman", "i")|| regex(?nomepais,
     "^Honduras", "i")|| regex(?nomepais, "^Congo - Brazzaville",
     "i")|| regex(?nomepais, "^Saint Barthelemy", "i")||
     regex(?nomepais, "^Bhutan", "i")|| regex(?nomepais, "^Guatemala",
     "i")|| regex(?nomepais, "^Antigua and Barbuda", "i")||
     regex(?nomepais, "^Somalia", "i")|| regex(?nomepais, "^New
     Zealand", "i")...)
10   }
11   GROUP BY ?nomepais
```

Consulta 8

"*Exiba no mapa o somatório dos oscars por países*".

Essa consulta demonstra a visualização espacial disponibilizada solução ExpSOLAP. A medida *Oscars* foi adicionada à divisória "Colunas" e o nível *Nome País* à divisória "Camadas". A visualização dos resultados pode ser em formato tabular (Figura 55) ou espacial (Figura 56). No Código Fonte 24 é ilustrada a consulta SPARQL resultante.

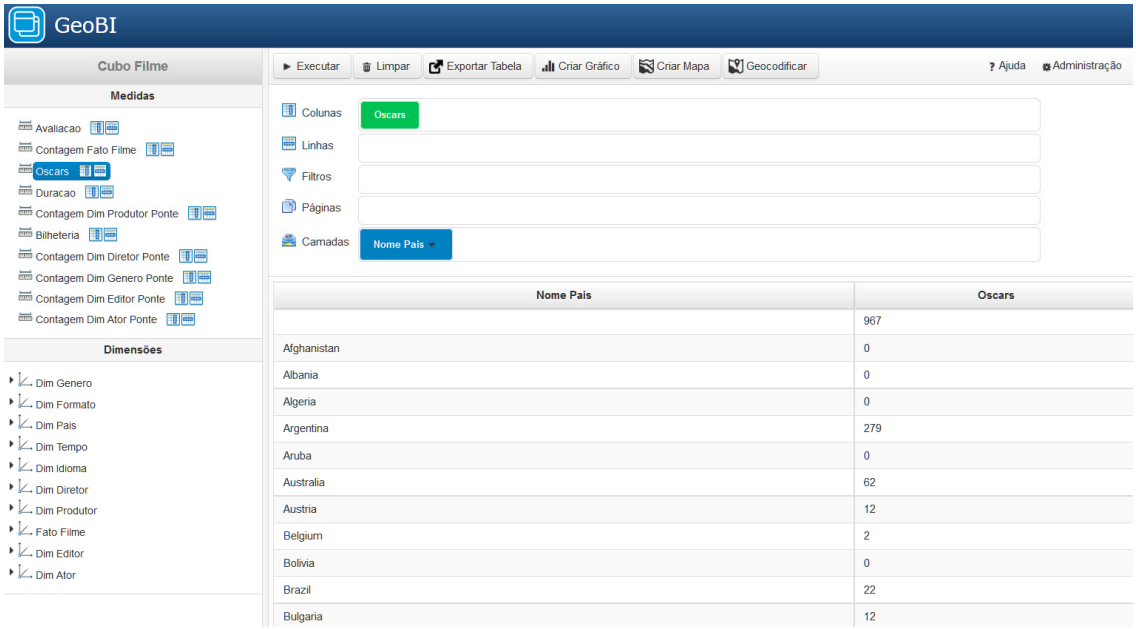

Figura 55: Consulta 8 - Resultados em formato tabular

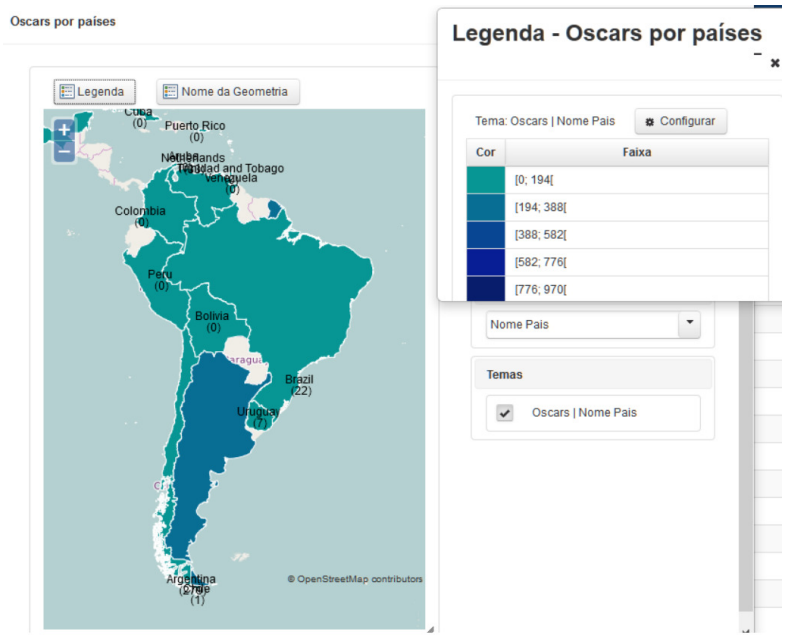

Figura 56: Consulta 8 - Visualização espacial dos resultados

Código Fonte 24: Consulta 8 - Consulta SPARQL gerada

```
1    <prefixos...>
2
3    SELECT ?nomepais (SUM(xsd:integer(?oscars)) AS ?oscars_sum)
4    WHERE
5    {
6       ?filme_uri_suj movie:country ?dimpais_uri .
7       ?dimpais_uri movie:country_name ?nomepais .
8       ?filme_uri_suj film:oscars ?oscars .
9    }
10   GROUP BY ?nomepais
```

5.2.2 Avaliação

A fim de avaliar a solução ExpSOLAP no que diz respeito à performance e desempenho da execução das consultas nas fontes de dados heterogêneas, cada consulta foi executada dez vezes; tendo o tempo de execução registrado e calculado o intervalo de confiança, com fator de 95%. O experimento foi executado em dois computadores com configurações distintas: no primeiro caso o ambiente é caracterizado por um computador com processador Intel Core i7 2.0 GHz, com 8 GB de memória RAM e HD 500 GB 5400 RPM; e o segundo ambiente, caracterizado por uma máquina com processador Intel Core i7 3770 3.4 GHz, com 32 GB de memória RAM e HD 2 TB 7200 RPM.

Na Figura 57 é ilustrado o intervalo de confiança para cada consulta executada no primeiro ambiente. É possível constatar que todas as consultas têm o tempo de execução alto, variando entre 125 e 200 segundos. Entretanto, no contexto de ferramentas analíticas, esse tempo de execução não é considerado elevado. Destaca-se ainda a consulta 3 (C3), que apresenta uma alta variação no tempo de execução. Uma possível justificativa para o alto tempo de execução desta consulta está na fonte de dados estruturada. A consulta manipula um atributo da tabela de fatos (título de filme), e por ter muitos registros na tabela de fatos, possui um tempo de execução maior que as demais consultas.

Na Figura 58 é ilustrado o intervalo de confiança do tempo de execução das consultas executadas no segundo ambiente de configuração. É perceptível o ganho considerável no tempo de execução das consultas, que variaram entre 38 e 99 segundos. Uma outra observação é com relação à pequena variação do tempo de execução das consultas – com exceção da consulta 3 (C3). Isso se deve ao fato de que a máquina utilizada possui uma configuração melhor, sobretudo a memória. O *Framework Apache Jena* executa as consultas em memória, sendo favorecido por essa configuração, que reduz o número de operações de *swapping* e, consequentemente, acesso ao disco rígido.

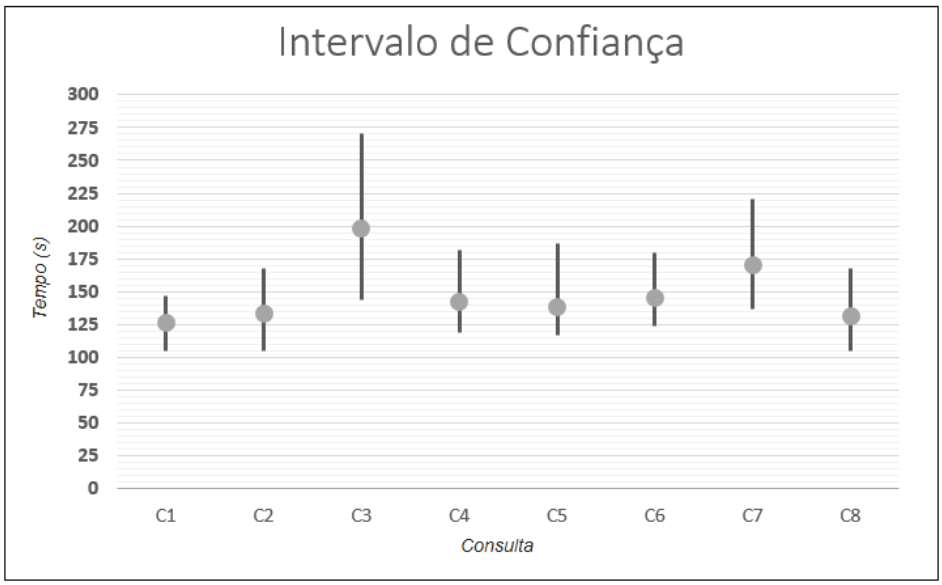

Figura 57: Intervalo de Confiança do Tempo de Execução das Consultas – Configuração 1

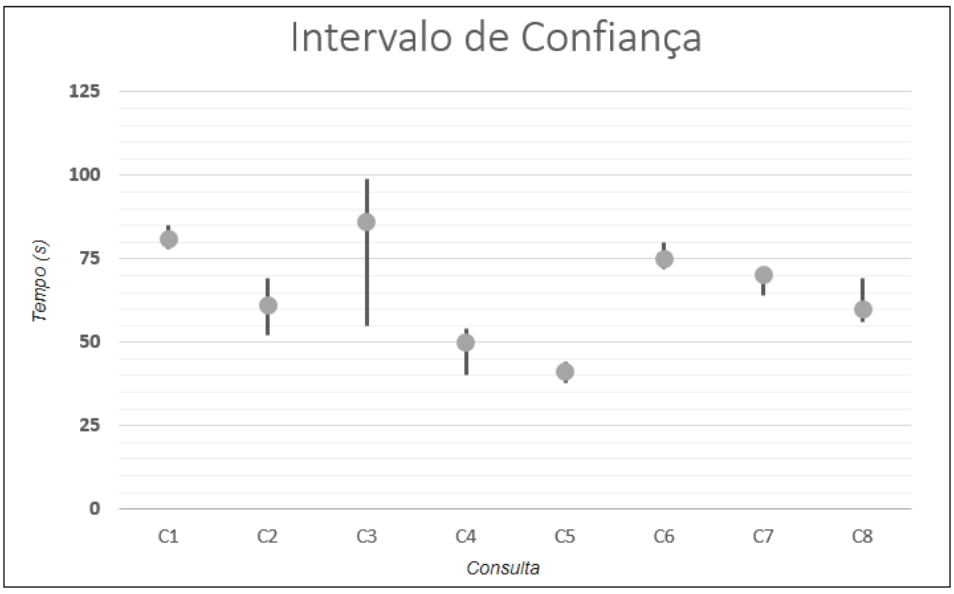

Figura 58: Intervalo de Confiança do Tempo de Execução das Consultas – Configuração 2

Agrupando as consultas em OLAP exploratório (C1, C2 e C3) e SOLAP exploratório (C4, C5, C6, C7 e C8) e calculando-se a média do tempo de execução, é possível comprovar que, por haver interseção entre os intervalos de confiança dos grupos de consultas, não há diferença entre o tempo de execução das consultas. Embora as consultas SOLAP exploratório acessem primeiro o repositório espacial para depois consultar a fonte de dados semântica, o tempo para executar o referido procedimento não afeta demasiadamente o tempo de execução geral da consulta. Na Figura 59, é ilustrado o intervalo de confiança para a média do tempo de execução das consultas OLAP exploratório (ExpOLAP) e SOLAP exploratório (ExpSOLAP) nas duas configurações de ambientes nos quais foram executados o experimento.

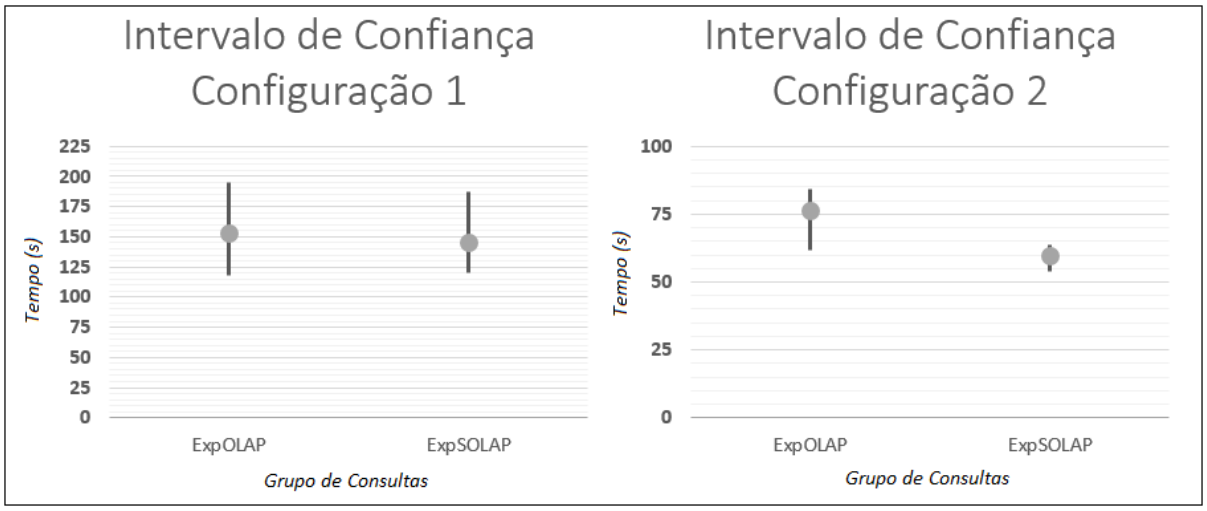

Figura 59: Comparação do Tempo de Execução por Grupo de Consultas

5.3 Considerações Finais

Este capítulo apresentou um exemplo prático utilizando o contexto de Filmes. Ressalta-se que é possível integrar bases de dados de qualquer outro contexto na solução ExpSOLAP; para tanto, faz-se necessário a criação do cubo multidimensional, o cadastro da fonte semântica e a incorporação de ambas as bases na solução, realizando o mapeamento manual necessário.

A exploração espacial das fontes de dados estruturadas e semiestruturadas através da solução ExpSOLAP trouxe resultados satisfatórios, comprovando a capacidade da solução em expandir o domínio de análise ao enriquecer dados internos (estruturado multidimensional) com a incorporação de fontes de dados semânticas, validando, assim, a questão de pesquisa Q2.

Por outro lado, novamente destaca-se que o tempo de processamento das consultas, explorando ambas as bases de dados, é superior comparando-se com uma solução que explora apenas a base multidimensional. Isso porque o Processador Exploratório faz toda a execução em memória, de modo que nada é pré-calculado como na base multidimensional.

No próximo capítulo, serão apresentadas as considerações finais sobre o trabalho desenvolvido nesta pesquisa, suas contribuições e os trabalhos futuros.

Capítulo 6 – Conclusões e Trabalhos Futuros

Recentemente, a comunidade científica tem trabalhado no sentido de propor metodologias para o enriquecimento semântico dos dados internos. Uma das vertentes propostas é que tal enriquecimento seja feito a partir da exploração de fontes de dados externas, disponíveis nos mais variados formatos (estruturadas, semiestruturadas ou não estruturadas). Esse movimento contribuiu para o surgimento das ferramentas *Exploratory* OLAP (Abelló et al., 2015), que oferecem a possibilidade de descobrir, adquirir, integrar e consultar dados, podendo ser facilmente e interligados a várias outras fontes de dados (*Linked Data*). Abelló et al. (2015) também propuseram uma categorização de ferramentas exploratórias OLAP – *Exploratory* OLAP. Neste trabalho, essa categorização foi estendida para ferramentas exploratórias SOLAP – *Exploratory* SOLAP, com a inclusão do critério de dimensionalidade dos dados. Essa nova categorização validou a questão de pesquisa Q1.

A partir do levantamento do estado da arte, apresentado no Capítulo 3, foi possível constatar que, por ser uma área ainda recente, há poucos estudos e ferramentas exploratórias desenvolvidas e ainda existem muitos desafios em aberto, a exemplo da exploração de fontes de dados heterogêneas e da exploração espacial das fontes de dados integradas. Com base nessas lacunas observadas, foi apresentada uma solução exploratória espacial, denominada ExpSOLAP, capaz de realizar análise espacial, simultânea, em duas fontes de dados heterogêneas: dados estruturados - esquema relacional - e dados semiestruturados, através do uso de semântica representada no formato RDF.

A solução ExpSOLAP dispõe de um conjunto de interfaces gráficas com as quais o usuário pode geocodificar membros das dimensões, formular consultas mediante uma linguagem de especificação visual (VQL) e analisar/visualizar os dados através de gráficos, relatórios ou mapas. Torna-se possível, assim, a restrição desses dados através da criação de filtros convencionais ou espaciais.

Para se conectar às fontes de dados, estruturadas ou semânticas, a ExpSOLAP oferece módulos de gerenciamento que possibilitam a criação da conexão para acessar cubos multidimensionais ou a criação de fonte de dados semânticas. Esse último módulo incorpora dados semiestruturados semânticos a partir do cadastro de propriedades de conexão. Essas propriedades referem-se a fonte de dados semânticos, como a localização física dos arquivos

RDF, prefixos e outras informações úteis. Cada fonte de dados semântica cadastrada deve ser associada a um cubo multidimensional através de um mapeamento entre os predicados e sua respectiva dimensão, medida ou nível. Tal mapeamento é primordial para o funcionamento do tradutor desenvolvido, que possibilita que consultas formuladas utilizando VQL sejam traduzidas em consultas SPARQL.

A arquitetura da solução ExpSOLAP mostrou-se satisfatória ao permitir a análise espacial em fontes heterogêneas, uma vez que a arquitetura seguiu a abordagem federada de integração de dados e disponibilizou um serviço de geocodificação que possibilita a análise espacial em fontes puramente convencionais, a exemplo de servidores OLAP. A referida abordagem também contribuiu para a análise espacial nos arquivos RDF, considerando não haver consenso na realização de consultas espaciais utilizando SPARQL.

A realização do exemplo prático no contexto de filmes interligados, comprovou, satisfatoriamente, a capacidade da solução em permitir uma análise exploratória integrada, espacial e simultânea, sobre dados estruturados e semiestruturados semânticos, possibilitando a visualização dos resultados em vários formatos, sejam gráficos, relatórios ou mapas. Nesta perspectiva, conclui-se que o trabalho desenvolvido nesta dissertação alcançou os seus objetivos específicos e também validou a questão de pesquisa Q2, tendo como principal contribuição a solução ExpSOLAP. Da análise do estado da arte, vê-se que esta é a primeira solução ExpSOLAP a ser proposta.

Embora este estudo aborde lacunas identificadas na integração entre as áreas de Web Semântica e BI, foram observados outros pontos de expansão, tratados a seguir, que certamente contribuirão para a evolução da solução aqui proposta.

6.1 Trabalhos Futuros

Nesta seção, são listados os pontos de expansão observados durante o desenvolvimento deste trabalho e que podem ser explorados em estudos futuros. Os pontos de expansão se baseiam nos critérios estendidos de Abelló et al. (2015), quais sejam:

- **Materialização:** a fonte de dados semântica integrada na solução ExpSOLAP é explorada através de sistema de arquivos, o que apresenta como desvantagem a necessidade de materializá-los. Como trabalho futuro, objetivando facilitar a integração de novos dados, sugere-se a integração da fonte semântica virtualmente, de forma que a solução explore dados através de um SPARQL *Endpoint*.

- **Frequência de Integração:** com a incorporação virtual de fontes de dados, não se faz mais necessária a integração em *batch*. Como trabalho futuro atrelado à materialização, propõe-se a investigação da integração de dados sob demanda.
- **Estruturação:** uma considerável quantidade de dados em formato não estruturado, rico em conteúdo, está disponível internamente (relatórios, arquivos de texto, *e-mails*, etc) ou externamente (na web, por exemplo, através das redes sociais). A incorporação de dados não estruturados em ferramentas exploratórias configura-se como uma importante contribuição científica.
- **Dimensionalidade:** a solução ExpSOLAP inova ao integrar dados espaciais e permitir a análise espacial através de operadores topológicos. Visando abranger o leque de possibilidades na análise espacial, tem-se como trabalho futuro a adição de novos operadores espaciais na solução proposta, a exemplo dos operadores *distance*, *buffer*, *difference* e *union*. Bem como, incorporar a dimensão espaço-temporal.
- **Extensibilidade:** no que tange à facilidade para integração de novas fontes de dados, propõem-se como trabalho futuro o desenvolvimento de um modelo semântico, de uma ontologia, para representar o domínio dos dados integrados à solução ExpSOLAP. Assim, facilitaria a incorporação de novas fontes de dados, visto que seria necessário mapear conceitos da nova fonte de dados a ser integrada com a ontologia do domínio, e não mais com a base multidimensional já incorporada à solução ExpSOLAP. Um outro esforço visando facilitar a integração de novas fontes de dados, é a utilização de algoritmos *matchers* (*Schema Matching*) para identificar, automaticamente, as correspondências entre as fontes de dados a serem integradas.

Também pretende-se aprofundar em metodologias de integração dos resultados (*merge*) retornados das fontes de dados heterogêneas, com o intuito de evitar a contagem de dados duplicados ou inexistentes em uma das fontes de dados integradas, além de possibilitar a realização de consultas holísticas (*BottomCount, BottomPercent, TopSum, Rank* entre outras).

Por fim, uma das limitações de sistemas que exploram dados semânticos é seu desempenho. Isso ainda é um problema em aberto na literatura, e a solução ExpSOLAP apresenta um gargalo relacionado ao tempo de processamento de consulta. Essa deficiência pode ser trabalhada com a adoção de uma infraestrutura de Big Data para o gerenciamento de arquivos RDF.

Referências Bibliográficas

Abelló, A.; Romero, O.; Pedersen, T.; Berlang, R.; Nebot, V.; Aramburu, M. J.; Simitsis, A., *Using semantic web Technologies for exploratory OLAP: A survey*. **IEEE Transactions on Knowledge and Data Engineering**, 27(2), p. 571-588, 2015.

Battle, R.; Kolas, D., *Enabling the geospatial Semantic Web with Parliament and GeoSPARQL*. **Semantic Web**, 3(4), p. 355-370, 2012.

Bédard, Y.; Merrett, T.; Han, J., *Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery*, **Vol. Research Monographs in GIS,** Taylor & Francis, chapter 3, p. 53-73, 2001.

Berlanga, R.; Romero, O.; Simitsis, A.; Nebot, V.; Pedersen, T. B.; Abelló, A., *Semantic Web Technologies for Business Intelligence*, **Business Intelligence Applications and the Web: Models, Systems and Technologies**, IGI Global, chapter 14, p. 310-339, 2012.

Berners-Lee, T, **Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web**, HarperBusiness, 2000.

Berners-Lee, T, 2006. *Linked Data*. Disponível em http://www.w3.org/DesignIssues/LinkedData.html, acesso em maio de 2016.

Berners-Lee, T.; Hendler, J.; Lassila, O., 2001. *The Semantic Web*. Disponível em: http://www.scientificamerican.com/article/the-semantic-web/, acesso em abril de 2016.

Bikakis, N.; Tsinaraki, C.; Stavrakantonakis, I.; Gioldasis, N.; Christodoulakis, S, *The SPARQL2XQuery Interoperability Framework*. **World Wide Web**, 18(2), p. 403-490, 2015.

Bimonte, S.; Boucelma, O.; Machabert, O.; Sellami, S., *A new Spatial OLAP approach for the analysis of Volunteered Geographic Information*. **Computers, Environment and Urban Systems**, 48, p. 111-123, 2014.

Bimonte, S.; Tchounikine, A.; Miquel, M., Spatial OLAP: Open Issues and a Web Based Prototype. In: *Proceedings of 10th AGILE International Conference on Geographic Information*. Aalborg, DK, 2007, p. 1-11.

Bizer, C.; Health, T.; Berners-Lee, T., *Linked Data - The Story So Far*, **International Journal on Semantic Web and Information Systems**, 5(3), p. 1-22, 2009.

Both, A.; Garcia-Rojas, A.; Wauer, M.; Hladky, D.; Lehmann, J., The GeoKnow Generator Workbench – An Integrated Tool Supporting the Linked Data Lifecycle for Enterprise Usage. In: *Proceedings of SEMANTiCS (Posters & Demos)*, Viena, AU, p. 92-95, 2015.

Brito, K. S.; Costa, M. A.; Garcia, V. C.; Meira, S. R., Experiences Integrating Heterogeneous Government Open Data Sources to Deliver Services and Promote Transparency in Brazil. In: *Proceedings of 38th Annual International Computers, Software and Applications Conference (COMPSAC),* Vasteras, SE, p. 606-607, 2014.

Chaudhuri, S.; Dayal, U.; Narasayya, V. (2011). *An overview of business intelligence technology*. **Communications of the ACM**, 54(8), p. 88-98, 2011.

Clark, L, SPARQL Views: A Visual SPARQL Query Builder for Drupal. In: *Proceedings of the International Semantic Web Conference (Posters & Demos)*, Xangai, CH, p. 161-164, 2010.

Etcheverry, L.; Vaisman, A., QB4OLAP: A Vocabulary for OLAP Cubes on the Semantic Web. In: *Proceedings of the Third International Workshop on Consuming Linked Data*, Boston, EUA, p. 114-132, 2012.

Etcheverry, L.; Vaisman, A.; Zimányi, E., Modeling and Querying Data Warehouses on the Semantic Web Using QB4OLAP. In: *Proceedings of Data Warehousing and Knowledge Discovery - DaWaK*, Munique, AL, p. 45-56, 2014.

Franklin, C., An introduction to geographic information systems: linking maps to databases. **Journal Database**, 15(2), p. 15-21, 1992.

Furtado, P.; Nadal, S.; Peralta, V.; Djedaini, M.; Labroche, N.; Marcel, P., Materializing Baseline Views for Deviation Detection Exploratory OLAP. In: *Proceedings of the 17th International Conference on Big Data Analytics and Knowledge Discovery – DaWaK*, Valência, ES, p. 243-254, 2015.

Gallinucci, E.; Golfarelli, M.; Rizzi, S., Meta-Starts: Multidimensional Modeling for Social Business Intelligence. In: *Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP - DOLAP*, São Francisco, EUA, p. 11-18, 2013.

Goodchild, M. F., Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0. **International Journal of Spatial Data Infrastructures Research - IJSDIR**, 2, p. 24-32, 2007.

Gür, N.; Hose, K.; Pedersen, B. P.; Zimányi, E., Modeling and Querying Spatial Data Warehouse on the Semantic Web. In: *Proceedings of the 5th Joint International Semantic Technology Conference*, Yichang, CH, p. 3-22, 2015.

Haag, F.; Lohmann, S.; Thomas, E., *SparqlFilterFlow: A SPARQL Query Composition for Everyone*. **The Semantic Web - ESWC**, Springer, p. 362-367, 2014.

Hannachi, L.; Benblidia, N.; Fadila, B.; Boussaid, O., Social microblogging cube. In: *Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP - DOLAP*, São Francisco, EUA, p. 19-26, 2013.

Ibragimov, D.; Hose, K.; Pedersen, T. B.; Zimányi, E., Towards Exploratory OLAP Over Linked Open Data - A Case Study. In: *Proceedings of Enabling Real-Time Business Intelligence*, Hangzhou, CH, p. 114-132, 2014.

Inmon, W. H.; Strauss, D.; Neushloss, G., **DW 2.0: The Architecture for the Next Generation of Data Warehousing**, Morgan Kaufmann, 2008.

Jensen, C. S.; Snodgrass, R. T., *Temporal Data Management*. **IEEE Transactions on Knowledge and Data Engineering**, 11(1), p. 36-44, 1999.

Jensen, C. S.; Pedersen, T. B.; Thomsen, C., **Multidimensional Databases and Data Warehousing**, Morgan & Claypool Publishers, 2010.

Kämpgen, B.; Harth, A., Transforming Statistical Linked Data for Use in OLAP Systems. In: *Proceedings of the 7th International Conference on Semantic Systems*, Graz, AU, p. 33-40, 2011.

Kämpgen, B.; O'Riain, S.; Harth, A., Interacting with Statistical Linked Data via OLAP Operations. In: *Proceedings of The Semantic Web: ESWC 2012*, Creta, GR, p. 87-101, 2012.

Kettouch, M. S.; Luca, C.; Hobbs, M.; Fatima, A., Data integration approach for semi-structured and structured data (Linked Data). In: *Proceedings of the IEEE 13th International Conference on Industrial Informatics*, Cambrige, UK, p. 820-825. 2015.

Kimball, R.; Ross, M.; Thornthwaite, W.; Mundy, J., **The Data Warehouse Lifecycle Toolkit**, Wiley, 2013.

Kimoto, T.; Morita, T.; Ishii, T.; Suga, H.; Sagawara, Y.; Beppu, T.; Yamaguchi, T., Integrating Heterogeneous Data Sources for Planning Road Reconstruction. In: *Proceedings of 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems*, p. 1720-1727, 2015.

Lassila, O.; Swick, R., 1999. *Resource Description Framework (RDF) Model and Syntax Specification*. Disponível em: https:// www.w3.org/TR/REC-rdf-syntax/, acesso em abril de 2016.

Malinowski, E, *GeoBI Architecture Based on Free Software: Experience and Review*, **Geographic Information Systems: Trends and Technologies**, CRC/Taylor & Francis Group, chapter 9, p. 244-286, 2014.

MICROSOFT, 2014. *Conceitos básicos de consulta MDX*. Disponível em https://msdn.microsoft.com/pt-br/library/ms145514%28v=sql.120%29.aspx, acesso em maio de 2016.

MICROSOFT, 2014. *Conceitos XMLA*. Disponível em https://msdn.microsoft.com/pt-br/library/bb522619%28v=sql.120%29.aspx, acesso em maio de 2016.

Miles, R.; Hamilton, K., **Learning UML 2.0**. O'Reilly, 2006.

Nebot, V.; Berlang, R., Building Data Warehouses with Semantic Data. In: *Proceedings of the International Conference on Extending Database Technology*, Bordo, FR, p. 150-157, 2010.

Neumayr, B.; Anderlik, S.; Schrefl., Towards Ontology-based OLAP: Datalog-based Reasoning over Multidimensional Ontologies. In: *Proceedings of the Fifteenth International Workshop on Data Warehousing and OLAP - DOLAP*, Havai, EUA, p. 41-48, 2012.

Neumayr, B.; Schütz, C.; Schrefl, M., Semantic enrichment of OLAP cubes: Multi-dimensional ontologies and their representation in SQL and OWL. In: *Proceedings of On the move to Meaningful Internet Systems*, Graz, AU, p. 624-641, 2013.

Niinimäki, M.; Niemi, T., *An ETL Process for OLAP Using RDF/OWL Ontologies*. **Journal on Data Semantic XIII**, 13, p. 97-119, 2009.

Noy, N. F.; McGuinness, D. L., 2001. *Ontology Development 101: A Guide to Creating Your First Ontology*. Disponível em: http://protegewiki.stanford.edu/wiki/Ontology101, acesso em março de 2016.

Open Source Geospatial Foundation (OSGeo). *OpenLayers: Free Maps for the Web*. Disponível em: http://openlayers.org/, acesso em março de 2016.

ORACLE, *Oracle and Java | Technologies | Oracle*. Disponível em: http://www.oracle.com/us/technologies/java/overview/index.html, acesso em junho de 2016.

Oukid, L.; Asfari, O.; Bentayeb, F.; Benblidia, N.; Boussaid, O., CXT-Cube: Contextual Text Cube Model and Aggregation Operator for Text OLAP. In: *Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP*, São Francisco, EUA, p. 27-32, 2013.

Park, B.-K.; Song, I.-Y., Toward total business intelligence incorporating structured and unstructured data. In: *Proceedings of the 2nd International Workshop on Business Intelligence and the WEB*, Uppsala, SE, p. 12-19, 2011.

Percival, G.; & Singh, R., 2012. *Geospatial Business Intelligence (GeoBI)*. Disponível em: http://www.opengeospatial.org/pressroom/papers, acesso em dezembro de 2014.

Prat, N.; Megdiche, I.; Akoka, J., Multidimensional models meet the semantic web. In: *Proceedings of the Fifteenth International Workshop on Data Warehousing and OLAP* - DOLAP, Havai, EUA, p. 17-24, 2014.

Rehman, N. U.; Weiler, A.; Scholl, M. H., OLAPing Social Media: The case of Twitter. In: *Proceedings of Advances in Social Networks Analysis and Mining - ASONAM*, Canada, p. 1139-1146, 2013.

Rivest, S.; Bédard, Y.; Proulx, M.-J.; Nadeau, M.; Hubert, F.; Pastor, J. *SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data*. **ISPRS Journal of Photogrammetry and Remote Sensing**, 60 (1), p. 17-33, 2005.

Romero, O.; Abelló, A. *A framework for multidimensional design of data warehouses from ontologies*. **Journal Data & Knowledge Engineering**, 69(11), p. 1138-1157, 2010.

Scoth, P.; Parmanto, B. 05). SOVAT: Spatial OLAP Visualization and Analysis Tool. In: *Proceedings of 38th Hawaii International Conference on System Sciences*, Havai, EUA, p. 142-148, 2005.

Seitz, S. M.; Baker, S., Filter Flow. In: *Proceedings of the 12th International Conference on Computer Vision*, Kyoto, JP, p. 143-150, 2009.

Shekhar, S. **Spatial Databases: A Tour**. Pearson, 2003.

Silva, J.; Oliveira, A. G.; Fidalgo, R. N.; Salgado, A. C.; Times, V. C. *Modeling and Querying Geographical Data Warehouses*. **Journal Information Systems**, 35 (5), p. 592-614, 2010.

Silva, T. E. *Um Framework para a análise espacial de fonte de dados multidimensionais*. 2013. 123 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Campina Grande, Campina Grande, Paraíba, 2013.

Stolte, C.; Hanrahan, P., Polaris: A System for Query, Analysis and Visualization of Multi-Dimensional Relational Databases. In: *Proceedings of the IEEE Symposium on Information Visualization*, Utah, EUA, p.5-14, 2000.

Tao, F., Lei, K. H., Han, J., Zhai, C., Cheng, X., Danilevsky, M.; Desai, N.; Ding, B., Ge, J.; Ji, H.; Kanade, R.; Kao, A.; Li, Q.; Li, Y.; Lin, C. X.; Liu, J.; Oza, N. C.; Srivastava, A. N.; Tjoelker, R.; Wang, C.; Zhang, D.; Zhao, B., EventCube: multi-dimensional search

and mining of structured and text data. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, EUA, p. 1494-1497, 2013.

W3C, 2004. *RDF Schema 1.1*. Disponível em: https://www.w3.org/TR/rdf-schema/, acesso em junho de 2016.

W3C, 2008. *SPARQL Query Language for RDF*. Disponível em: https://www.w3.org/TR/rdf-sparql-query/, acesso em junho de 2016.

Wang, C.-J.; Ku, W.-S.; Chen, H., Geo-Store: A Spatially-Augmented SPARQL Query Evaluation System. In: *Proceedings of International Conference on Advances in Geographic Information Systems - SIGSPATIAL*, California, EUA, p. 562-565, 2012.

Watson, M. **Web 3.0 Information Gathering and Processing**. Apress, 2009.

Wiegand, N.; Grove, R.; Wilson, J.; Kolas, D., Querying Geospatial Data over the web: a GeoSPARQL Interface. In: *Proceedings of Workshop on Managing and Mining Enriched Geo-Spatial Data*, Utah, EUA, p. 4-6, 2014.

Wolfson, O.; Xu, B.; Chamberlain, S.; Jiang, L., Moving Object Databases: Issues and Solutions. In: *Proceedings of the 10th International Conference on Scientific and Statistical Database Management*, Capri, IT, p. 111-122, 1998.

Zaveri, A.; Rula, A.; Maurino, A.; Pietrobon, R.; Lehmann, J.; & Auer, S. *Quality Assessment for Linked Data: A Survey*. **Semantic Web Journal**, 7(1), p. 63-93, 2016.

Zhai, X.; Huang, L.; Xiao, Z., Geo-spatial Query Based on Extended SPARQL. In: *Proceedings of 18th International Conference on Geoinformatics: GIScience in Change*, Pequim, CH, p. 863-866, 2010.

Zou, L.; Özsu, M. T.; L., C.; Schen, X.; Huang, R.; Zhao, D. *gStore: a graph-based SPARQL query engine*. **The International Journal on Very Large Data Bases - The VLDB Journal**, 23(4), p. 565-590, 2014.

Apêndice A – Ontologia movie.owl

```
1   <?xml version="1.0"?>
2      <!DOCTYPE rdf:RDF [<!ENTITY owl "http://www.w3.org/2002/07/owl#">
3      <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
4      <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
5      <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >]>
6
7    <rdf:RDF
8      xmlns="http://www.semanticweb.org/daniel/ontologies/2015/9/untitle
       d-ontology-8#"
9      xmlns:base="http://www.semanticweb.org/daniel/ontologies/2015/9/un
       titled-ontology-8"
10     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
11     xmlns:owl="http://www.w3.org/2002/07/owl#"
12     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
13     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
14
15     <owl:Ontology rdf:about="http://150.165.75.163/2015/10/movie"/>
16
17     <!- Data properties -->
18
19     <!- http://150.165.75.163/2015/10/movie#box_office -->
20     <owl:DatatypeProperty
       rdf:about="http://150.165.75.163/2015/10/movie#box_office">
21         <rdfs:domain
           rdf:resource="http://150.165.75.163/2015/10/movie#Movie"/>
22         <rdfs:range rdf:resource="&xsd;decimal"/>
23     </owl:DatatypeProperty>
24
25     <!- http://150.165.75.163/2015/10/movie#oscars -->
26     <owl:DatatypeProperty
       rdf:about="http://150.165.75.163/2015/10/movie#oscars">
27         <rdfs:domain
           rdf:resource="http://150.165.75.163/2015/10/movie#Movie"/>
28         <rdfs:range rdf:resource="&xsd;integer"/>
29     </owl:DatatypeProperty>
30
31     <!- http://150.165.75.163/2015/10/movie#rating -->
32     <owl:DatatypeProperty
       rdf:about="http://150.165.75.163/2015/10/movie#rating">
33         <rdfs:domain
           rdf:resource="http://150.165.75.163/2015/10/movie#Movie"/>
34         <rdfs:range rdf:resource="&xsd;decimal"/>
35     </owl:DatatypeProperty>
36
37     <!- Classes -->
38     <owl:Class rdf:about="http://150.165.75.163/2015/10/movie#Movie">
39       <rdfs:subClassOf>
```

```
40          <owl:Restriction>
41            <owl:onProperty
              rdf:resource="http://150.165.75.163/2015/10/movie#oscars"/>
42            <owl:maxCardinality rdf:datatype=
              "&xsd;nonNegativeInteger">1</owl:maxCardinality>
43          </owl:Restriction>
44        </rdfs:subClassOf>
45
46        <rdfs:subClassOf>
47          <owl:Restriction>
48            <owl:onProperty
              rdf:resource="http://150.165.75.163/2015/10/movie#rating"/>
49            <owl:maxCardinality rdf:datatype=
              "&xsd;nonNegativeInteger">1</owl:maxCardinality>
50          </owl:Restriction>
51        </rdfs:subClassOf>
52
53        <rdfs:subClassOf>
54          <owl:Restriction>
55            <owl:onProperty
              rdf:resource="http://150.165.75.163/2015/10/movie#box_offic
              e"/>
56            <owl:maxCardinality rdf:datatype=
              "&xsd;nonNegativeInteger">1</owl:maxCardinality>
57          </owl:Restriction>
58        </rdfs:subClassOf>
59
60      </owl:Class>
61    </rdf:RDF>
```

Apêndice B – Ontologia time.owl

```
1   <?xml version="1.0"?>
2     <!DOCTYPE rdf:RDF [<!ENTITY owl "http://www.w3.org/2002/07/owl#">
3     <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
4     <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
5     <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >]>
6
7   <rdf:RDF
8     xmlns="http://www.w3.org/2002/07/owl#"
9     xmlns:base="http://www.w3.org/2002/07/owl"
10    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
11    xmlns:owl="http://www.w3.org/2002/07/owl#"
12    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
13    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
14
15    <Ontology rdf:about="http://150.165.75.163/2015/10/time">
16      <imports rdf:resource="http://www.w3.org/2001/XMLSchema#">
17      <imports rdf:resource="http://www.w3.org/2006/time">
18    </Ontology>
19
20    <!– Datatypes –->
21
22    <!– http://150.165.75.163/2015/10/time#gBimester –>
23    <rdfs:Datatype
      rdf:about="http://150.165.75.163/2015/10/time#gBimester">
24
25    <!– http://150.165.75.163/2015/10/time#gQuarter –>
26    <rdfs:Datatype
      rdf:about="http://150.165.75.163/2015/10/time#gQuarter">
27
28    <!– http://150.165.75.163/2015/10/time#gSemester –>
29    <rdfs:Datatype
      rdf:about="http://150.165.75.163/2015/10/time#gSemester">
30
31    <!– Data properties –>
32
33    <!– http://150.165.75.163/2015/10/time#bimester –>
34    <DatatypeProperty
       rdf:about="http://150.165.75.163/2015/10/time#bimester">
35      <rdfs:domain
        rdf:resource="http://www.w3.org/2006/time#DateTimeDescription
        "/>
36      <rdfs:range
        rdf:resource="http://150.165.75.163/2015/10/time#gBimester"/>
37    </DatatypeProperty>
38
39    <!– http://150.165.75.163/2015/10/time#quarter –>
```

```
40      <DatatypeProperty
         rdf:about="http://150.165.75.163/2015/10/time#quarter">
41         <rdfs:domain
           rdf:resource="http://www.w3.org/2006/time#DateTimeDescription
           "/>
42         <rdfs:range
           rdf:resource="http://150.165.75.163/2015/10/time#gQuarter"/>
43      </DatatypeProperty>
44
45      <!– http://150.165.75.163/2015/10/time#semester –->
46      <DatatypeProperty
         rdf:about="http://150.165.75.163/2015/10/time#semester">
47      <rdfs:domain
        rdf:resource="http://www.w3.org/2006/time#DateTimeDescription"/>
48      <rdfs:range
        rdf:resource="http://150.165.75.163/2015/10/time#gSemester"/>
49      </DatatypeProperty>
50      <!– http://150.165.75.163/2015/10/time#bimester –->
51      <DatatypeProperty
         rdf:about="http://150.165.75.163/2015/10/time#bimester">
52
53      <!– Classes –->
54
55      <!–– http://150.165.75.163/2015/10/time#BimesterOfYear –->
56      <Class
        rdf:about="http://150.165.75.163/2015/10/time#BimesterOfYear">
57        <equivalentClass>
58          <Class>
59            <oneOf rdf:parseType="Collection">
60              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#B3"/>
61              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#B4"/>
62              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#B5"/>
63              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#B6"/>
64              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#B1"/>
65              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#B2"/>
66            </oneOf>
67          </Class>
68        </equivalentClass>
69      </Class>
70
71      <!–– http://150.165.75.163/2015/10/time#QuarterOfYear –->
72      <Class
        rdf:about="http://150.165.75.163/2015/10/time#QuarterOfYear">
73        <equivalentClass>
74          <Class>
75            <oneOf rdf:parseType="Collection">
76              <rdf:Description
```

```
                rdf:about="http://150.165.75.163/2015/10/time#Q2"/>
77              <rdf:Description
                rdf:about="http:/150.165.75.163/2015/10/time#Q3"/>
78              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#Q4"/>
79              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#Q1"/>
80            </oneOf>
81          </Class>
82        </equivalentClass>
83      </Class>
84
85      <!-- http://150.165.75.163/2015/10/time#SemesterOfYear -->
86      <Class
        rdf:about="http://150.165.75.163/2015/10/time#SemesterOfYear">
87        <equivalentClass>
88          <Class>
89            <oneOf rdf:parseType="Collection">
90              <rdf:Description
                rdf:about="http://150.165.75.163/2015/10/time#S1"/>
91              <rdf:Description
                rdf:about="http:/150.165.75.163/2015/10/time#S2"/>
92            </oneOf>
93          </Class>
94        </equivalentClass>
95      </Class>
96
97      <!-- http://www.w3.org/2006/time#DateTimeDescription -->
98      <rdf:Description
        rdf:about="http://www.w3.org/2006/time#DateTimeDescription">
99        <rdfs:subClassOf>
100         <Restriction>
101           <onProperty
              rdf:resource="http://150.165.75.163/2015/10/time#quarter"/
              >
102           <maxCardinality
              rdf:datatype="&xsd;nonNegativeInteger">1</maxCardinality>
103         </Restriction>
104       </rdfs:subClassOf>
105       <rdfs:subClassOf>
106         <Restriction>
107           <onProperty
              rdf:resource="http://150.165.75.163/2015/10/time#semester"
              />
108           <maxCardinality
              rdf:datatype="&xsd;nonNegativeInteger">1</maxCardinality>
109         </Restriction>
110       </rdfs:subClassOf>
111       <rdfs:subClassOf>
112         <Restriction>
113           <onProperty
              rdf:resource="http://150.165.75.163/2015/10/time#bimester"
              />
```

```
114             <maxCardinality
                rdf:datatype="&xsd;nonNegativeInteger">1</maxCardinality>
115           </Restriction>
116         </rdfs:subClassOf>
117       </rdf:Description>
118
119       <!-- Individuals -->
120
121       <!-- http://150.165.75.163/2015/10/time#B1 -->
122       <NamedIndividual
          rdf:about="http://150.165.75.163/2015/10/time#B1">
123         <rdf:type
            rdf:resource="http://150.165.75.163/2015/10/time#BimesterOfYea
            r"/>
124       </NamedIndividual>
125
126       <!-- http://150.165.75.163/2015/10/time#B2 -->
127       <NamedIndividual
          rdf:about="http://150.165.75.163/2015/10/time#B2">
128         <rdf:type
            rdf:resource="http://150.165.75.163/2015/10/time#BimesterOfYea
            r"/>
129       </NamedIndividual>
130
131       <!-- http://150.165.75.163/2015/10/time#B3 -->
132       <NamedIndividual
          rdf:about="http://150.165.75.163/2015/10/time#B3">
133         <rdf:type
            rdf:resource="http://150.165.75.163/2015/10/time#BimesterOfYea
            r"/>
134       </NamedIndividual>
135
136       <!-- http://150.165.75.163/2015/10/time#B4 -->
137       <NamedIndividual
          rdf:about="http://150.165.75.163/2015/10/time#B4">
138         <rdf:type
            rdf:resource="http://150.165.75.163/2015/10/time#BimesterOfYea
            r"/>
139       </NamedIndividual>
140
141       <!-- http://150.165.75.163/2015/10/time#B5 -->
142       <NamedIndividual
          rdf:about="http://150.165.75.163/2015/10/time#B5">
143         <rdf:type
            rdf:resource="http://150.165.75.163/2015/10/time#BimesterOfYea
            r"/>
144       </NamedIndividual>
145
146       <!-- http://150.165.75.163/2015/10/time#B6 -->
147       <NamedIndividual
          rdf:about="http://150.165.75.163/2015/10/time#B6">
148         <rdf:type
            rdf:resource="http://150.165.75.163/2015/10/time#BimesterOfYea
```

```
        r"/>
149     </NamedIndividual>
150
151     <!-- http://150.165.75.163/2015/10/time#Q1 -->
152     <NamedIndividual
        rdf:about="http://150.165.75.163/2015/10/time#Q1">
153       <rdf:type
          rdf:resource="http://150.165.75.163/2015/10/time#QuarterOfYear
          "/>
154     </NamedIndividual>
155
156     <!-- http://150.165.75.163/2015/10/time#Q2 -->
157     <NamedIndividual
        rdf:about="http://150.165.75.163/2015/10/time#Q2">
158       <rdf:type
          rdf:resource="http://150.165.75.163/2015/10/time#QuarterOfYear
          "/>
159     </NamedIndividual>
160
161     <!-- http://150.165.75.163/2015/10/time#Q3 -->
162     <NamedIndividual
        rdf:about="http://150.165.75.163/2015/10/time#Q3">
163       <rdf:type
          rdf:resource="http://150.165.75.163/2015/10/time#QuarterOfYear
          "/>
164     </NamedIndividual>
165
166     <!-- http://150.165.75.163/2015/10/time#Q4 -->
167     <NamedIndividual
        rdf:about="http://150.165.75.163/2015/10/time#Q4">
168       <rdf:type
          rdf:resource="http://150.165.75.163/2015/10/time#QuarterOfYear
          "/>
169     </NamedIndividual>
170
171     <!-- http://150.165.75.163/2015/10/time#S1 -->
172     <NamedIndividual
        rdf:about="http://150.165.75.163/2015/10/time#S1">
173       <rdf:type
          rdf:resource="http://150.165.75.163/2015/10/time#SemesterOfYea
          r"/>
174     </NamedIndividual>
175
176     <!-- http://150.165.75.163/2015/10/time#S2 -->
177     <NamedIndividual
        rdf:about="http://150.165.75.163/2015/10/time#S2">
178       <rdf:type
          rdf:resource="http://150.165.75.163/2015/10/time#SemesterOfYea
          r"/>
179     </NamedIndividual>
180   </rdf:RDF>
```