PROJETO NÚMEROS – UM SISTEMA PARA CONSTRUÇÃO DE CONSULTAS ENRIQUECIDAS SEMANTICAMENTE

Juliano Julio Arruda da Costa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Jano Moreira de Souza, Ph.D.

Prof. Geraldo Bonorino Xexéo, D.Sc.

Prof. Geraldo Zimbrão da Silva, D.Sc.

Prof. Maria Claudia Reis Cavalcanti, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2006

COSTA, JULIANO JULIO ARRUDA DA

Projeto Números - Um sistema para construção de consultas enriquecidas semanticamente [Rio de Janeiro] 2006

X, 99 p., 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2006)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Consultas por exemplo
2. Consultas semânticas
3. Mineração de conteúdo da Internet

I. COPPE/UFRJ II. Título (série)

À minha mãe.

Agradecimentos

Ao Pai pela vida e oportunidades que tive.

À minha mãe pelo amor incondicional.

À minha tia Judith por sua dedicação e fé.

Ao meu irmão Robson e ao meu amigo João Batista, pelo companheirismo.

À minha companheira Carla Graziela e nossos planos futuros.

Ao professor Geraldo Xexéo, por me aceitar como aluno, ter tido paciência entre as mudanças ao longo de meus estudos e por sua orientação.

Ao professor Jano de Souza pelas suas valiosas contribuições ao meu trabalho, e aos outros professores da COPPE por contribuírem em minha formação.

À fundação COPPETEC onde pude trabalhar e conciliar meus estudos.

Ao Ricardo Barros que contribuiu muito com idéias e revisões.

Aos muitos novos amigos que fiz na cidade maravilhosa, baianos, mineiros, capixabas, fluminenses e paulistas.

Aos velhos amigos que apesar da distancia, sempre estiveram presentes me incentivando e apoiando.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PROJETO NÚMEROS – UM SISTEMA PARA CONSTRUÇÃO DE CONSULTAS ENRIQUECIDAS SEMANTICAMENTE

Juliano Julio Arruda da Costa

Setembro / 2006

Orientadores: Jano Moreira de Souza
Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

A busca e visualização de dados textuais na Internet é um campo consolidado que ganha complexidade e eficiência com o passar dos anos. Empresas cujo nicho inicial restringia-se à busca textual esforçam-se em prover mecanismos de busca sobre dados multimídias e inferência de informações.

A busca por dados numéricos é ainda pouco explorada, obrigando o usuário a realizar diversas consultas, nos sistemas de buscas, e ao fim do processo se deparar com uma série de questões sobre a qualidade dos dados encontrados.

Este trabalho apresenta o Projeto Números, um conjunto de ferramentas e uma arquitetura para buscas com resultados numéricos realizadas em repositórios heterogêneos de dados estruturados ou não.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

NUMBERS PROJECT – A SYSTEM TO BUILD QUERIES SEMANTICALLY ENRICHED

Juliano Julio Arruda da Costa

September / 2006

Advisor: Jano Moreira de Souza
Geraldo Bonorino Xexéo

Department: Computer Science and Systems Engineering

The search and visualization of textual data on Internet is a consolidated subject, which gains complexity and effectiveness along the years. Enterprises which initial focus was textual searches are now providing ways to enable multimedia searches and information inference.

Searches for numerical data aren't a much explored subject, forcing the user to realize a few queries on Web searches engines, and at the end of this process, the user is confronted against questions about data quality.

This work presents the Numbers Project, a set of tools and a framework for numerical data search in heterogeneous repositories of structured data or not.

Índice

Índice de Figuras

Capítulo 1 – Introdução

A busca e visualização de dados textuais na Internet é hoje um campo consolidado que ganha complexidade e eficiência com o passar dos anos. Empresas cujo nicho inicial restringia-se à busca textual esforçam-se em prover mecanismos de busca sobre dados multimídias e inferência de informações. Mas em uma situação onde um pesquisador deseja buscar informações como o Produto Interno Bruto de um determinado país, ou ainda uma consulta mais complexa com o objetivo de colher dados para uma tabela, este pesquisador encontrará dificuldades frente aos atuais mecanismos de busca conhecidos, pois será obrigado a realizar pesquisas exaustivas até que consiga reunir todos os dados, e ainda assim enfrentará o questionamento: os dados estão corretos? Estão atualizados?

Este trabalho apresenta o Projeto Números, um conjunto de ferramentas e uma arquitetura para realizar buscas com resultados numéricos em repositórios heterogêneos de dados estruturados ou não. Após uma revisão sobre *Data Warehouse* apresentamos um sistema capaz de criar e enriquecer semanticamente uma consulta a dados numéricos, utilizando ontologias e informações fornecidas pelo usuário, e propomos uma implementação que trata o armazenamento de resultados.

1.1 – Motivação

Com o advento da Internet o acesso e distribuição de dados se tornam cada vez mais acessível. As instituições de pesquisas estatísticas disponibilizam parte de sua base de dados e surgem então novos desafios como a construção de ferramentas capazes de recuperar e realizar um cruzamento entre esses dados, e ferramentas que encontrem dados entre os diversos repositórios.

Hans Rosling ressalta a necessidade das instituições de pesquisas estatísticas continuarem a disponibilizar seus dados e destaca ainda a necessidade de ferramentas que permitam a manipulação desses dados através de uma representação gráfica, pois assim um maior número de pessoas será beneficiado pela informação ali contida. (RÖNNLUND e ROSLING, 2006; ROSLING, 2006)

Os dados mencionados por Rosling são em sua maioria dados numéricos que são representados de várias formas: bancos de dados; tabelas em diferentes formatos físicos

(como HTML ou planilhas Excel), gráficos em diferentes formas (barra, linha, etc.) e formatos (GIF, JPEG, GNUPLOT, etc.), documentos semi-estruturados (XML, SGML), documentos estruturados e ainda na forma de texto livre.

Nos atuais sistemas de buscas textuais os usuários do sistema são capazes de expressar, utilizando palavras chave, os objetos de suas buscas e se desejarem resultados mais apurados, melhoram suas consultas com mais termos, frases ou expressões regulares. Na busca por dados numéricos, embora a consulta seja descrita de uma forma predominantemente textual, os dados resultantes são basicamente números, muitas vezes rotulados por atributos que representam metadados.

Desta forma em buscas por números a interface com o usuário deve ser capaz de adquirir e processar o máximo de informações possível sobre a busca em questão, bem como toda e qualquer informação semântica que o usuário puder fornecer. Assim, a motivação deste trabalho está no desafio representado pelas consultas com resultados numéricos, em como representá-las e como implementar um sistema para encontrar e armazenar os resultados.

1.2 – Objetivos do trabalho

Este trabalho tem como objetivo definir e apresentar a arquitetura do Projeto Números, um sistema para resolução de consultas que visam resultados numéricos, procurando por dados em repositórios de dados heterogêneos ou não, como a Internet.

A arquitetura proposta por este trabalho terá componentes implementados em outros trabalhos acadêmicos. O presente trabalho apresenta uma proposta de Interface Gráfica para construção e enriquecimento de consultas, permitindo ao usuário expressar sua busca por dados numéricos. Além disso, apresenta parte do processamento da consulta do usuário, destinado à preparação dos repositórios de dados para armazenamento dos resultados das consultas.

1.3 – Organização do trabalho

A elaboração de uma arquitetura requer conhecimento sobre a dimensão desta, e para adquiri-lo é necessário um estudo sobre a representação dos dados que circundam e são utilizados pelo sistema em questão.

Assim o Capítulo 2 faz uma revisão sobre o modelo de dados multidimensional e descreve uma aplicação que recupera dados da Internet para popular um banco de dados baseado nesse modelo. Ao final deste capítulo é traçado um paralelo entre os recursos

apresentados pela ferramenta estudada e o Projeto Números, destacando as contribuições deste trabalho.

O Capítulo 3 apresenta uma breve discussão sobre os dados na Internet, destacando algumas estatísticas sobre os formatos de arquivos e como os dados estão organizados.

O Capítulo 4 descreve a arquitetura do Projeto Números, apresentando os requisitos principais e técnicos, como a utilização de determinada linguagem de programação e ainda a utilização por softwares livres no desenvolvimento. Este capítulo apresenta também uma visão geral das classes que devem compor a arquitetura e suas responsabilidades, definindo o fluxo de dados entre as diversas camadas e macro componentes.

O Capítulo 5 apresenta uma implementação de referência, isto é, um sistema que apesar de demonstrar as funcionalidades esperadas, não deve ser considerado uma versão final do projeto, mas sim um ponto de partida para novos desenvolvedores. Este capítulo descreve ainda as classes utilizadas na implementação da camada de Interface Gráfica com o Usuário e componentes da camada de Dados e Conhecimento.

A primeira parte do capítulo 5 descreve um sistema capaz de auxiliar o usuário na construção de consultas utilizando dados como exemplo, e ainda permite ao usuário expandir e agregar valor semântico à consulta inicial através do uso de Ontologias e uma interface gráfica amigável. A segunda parte do capítulo aborda a parte do servidor do Projeto Números responsável por receber uma consulta e preparar o repositório de dados relacional para armazenar os dados resultantes e também atualizar o modelo de dados do banco de dados multidimensional.

O Capítulo 6 descreve uma simulação de uso do sistema e como o processamento da consulta ocorreria para a recuperação dos dados utilizando mecanismos de buscas na Web.

Por fim, o Capítulo 7 apresenta a conclusão deste trabalho, apontando os objetivos alcançados e indicando as perspectivas para trabalhos futuros.

Capítulo 2 – *Data Warehouse*

Os sistemas de bancos de dados e teoria de representação de dados são tópicos estudados há bastante tempo. Inicialmente estes sistemas visavam atender tanto ao ambiente operacional quanto os ambientes de dados analíticos, necessários para tomadas de decisão ou conhecimento das informações produzidas. Mas o processamento das informações analíticas possui um tempo muito acentuado em relação ao processamento operacional, este podendo levar de 30 minutos até várias horas, tempo inviável em um ambiente operacional.

O alto custo de processamento das informações analíticas nos mesmos repositórios de produção criou a necessidade de alternativas para a aquisição destas informações. A alternativa é a extração dos dados do ambiente operacional para um ambiente analítico exclusivo para a análise coletiva dos dados (INMON, 1997).

A extração dos dados do ambiente operacional tornou-se comum, e percebeu-se claramente uma projeção do ambiente de produção para o ambiente de análise, ou um ambiente de apoio à decisão. Esta mudança de organização dos dados gerou a necessidade de metodologias para organização e ferramentas para manipulação dos dados analíticos. Muitas destas estratégias e técnicas são sintetizadas no ambiente de *Data Warehouse*.

Este capítulo faz uma breve revisão dos principais aspectos de *Data Warehouse* e do desenvolvimento deste ambiente.

2.1 – Arquitetura

Gill e Rao afirmam que o *Data Warehouse* é um processo contínuo, que combina dados de múltiplas fontes heterogêneas incluindo dados históricos e agrupados, segundo alguma política, para apoiar as necessidades de consultas estruturadas e *ad hoc*, relatórios analíticos e apoio à tomada de decisão (GILL e RAO, 1996).

Já Mattison define *Data Warehouse* como uma coleção de dados copiados de outros sistemas e montados em um local diferente disponível para os propósitos do usuário final em atividades de apoio à tomada de decisão e tomada de decisão e coleta de informações (MATTISON, 1997).

Para Inmon "um *Data Warehouse* é um conjunto de dados baseado em assuntos, integrado, não-volúvel, e variável em relação ao tempo, de apoio às decisões gerenciais" (INMON, 1997). Esta definição é obtida a partir da observação das seguintes características:

- Baseado em assuntos – foco na análise e modelagem dos dados, provendo uma visão simples e concisa sobre um assunto em particular, excluindo toda informação desnecessária no processo de tomada de decisões;
- Integrado – os dados são tratados para dar a visão baseada em assunto e a origem pode ser heterogênea (vindo de sistemas legados, ou banco de dados de sistemas atuais);
- Não volúvel – os dados para suporte a decisão não necessitam de atualizações, transações ou controle de concorrência, suas únicas operações são a carga de dados oriundos da base operacional e o acesso aos dados armazenados;
- Variável em relação ao tempo – todo dado no *Data Warehouse* contém explicitamente ou implicitamente uma informação que qualifique temporalmente esse dado, mesmo que na base de dados operacional essa informação não exista.

2.1.1 – Estruturação dos dados

O Modelo Relacional de representação de dados define uma gama de boas práticas e metodologias para organização e representação dos dados buscando a eliminação de anomalias, como a repetição de dados válidos, a impossibilidade de inserção ou remoção de valores e a necessidade de atualizar diversas linhas devido à replicação de informações.

A teoria relacional beneficiou o processamento de transações no ambiente operacional que passaram a ser realizadas de forma mais simples e determinística, mas esta organização não se mostra eficiente para o ambiente analítico, o que exigiu uma diferente organização dos dados para estas consultas.

Desta forma o *Data Warehouse* pode ser descrito como um agrupamento de dados históricos provenientes de bases operacionais heterogêneas ou não. A representação dos dados no modelo *Data Warehouse* é usualmente através da modelagem multidimensional, onde a idéia é avaliar a informação através de diferentes perspectivas, simulando um cubo com *n* **dimensões**, cada uma representando as

diferentes formas que o usuário deseja visualizar os dados, e onde as perspectivas representam um tema central denominado **fato**.

Tabelas de Fatos armazenam medidas numéricas relativas ao negócio em questão, por exemplo, em um *Data Warehouse* de vendas, a tabela de fatos armazenaria: unidades vendidas; valor das vendas. As tabelas de Dimensões armazenam informações textuais sobre informações do negócio em questão, por exemplo, no mesmo *Data Warehouse* de vendas, as dimensões esperadas seriam: nome do produto; marca; tipo do produto. (HAN e KAMBER, 2001)

O modelo de dados de *Data Warehouse* mais popular é o modelo multidimensional. Este modelo pode ser representado de três formas: Estrela, Floco de Neve e Constelação. O modelo Estrela é o mais simples e o mais comum, é caracterizado por uma grande tabela de fatos central e um conjunto de tabelas, uma para cada dimensão. A Figura 1 demonstra essa representação. (HAN e KAMBER, 2001)

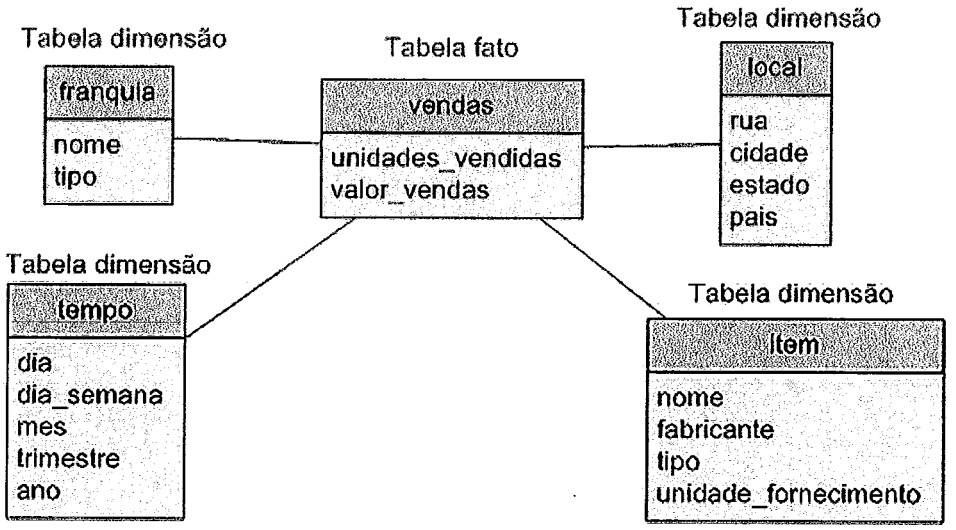


Figura 1 - Representação gráfica do modelo Estrela (Adaptado de: (HAN e KAMBER, 2001))

O modelo Flocos de Neve é uma variação do modelo Estrela, mas com técnicas de normalização nas tabelas de dimensão (preocupação inexistente no modelo Estrela). A Figura 2 demonstra a tabela central fato e as tabelas de dimensões normalizadas, a origem do nome é fruto da aparência desta representação gráfica com um floco de neve.

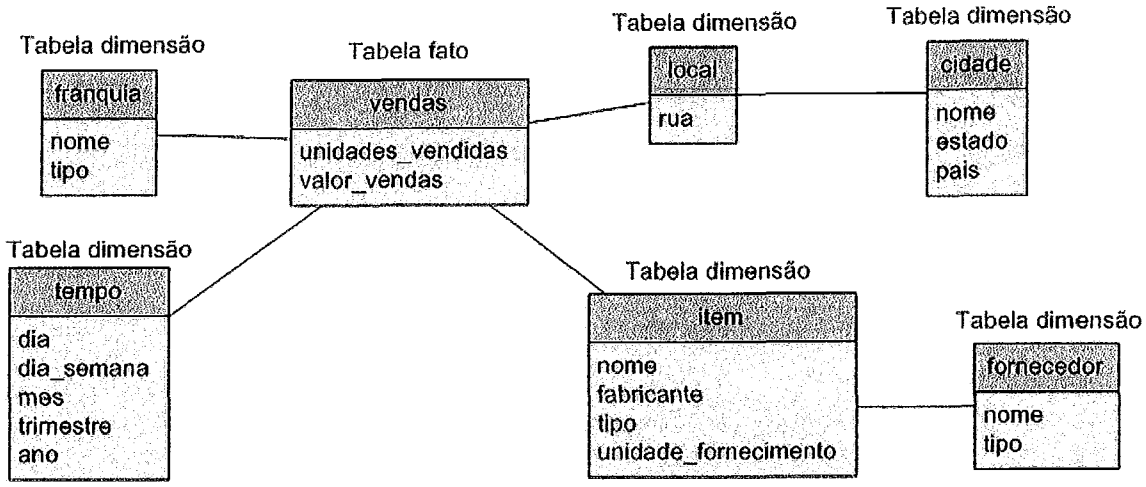


Figura 2 - Representação gráfica do modelo Floco de Neve (Adaptado de: (HAN e KAMBER, 2001))

O modelo Constelação é comum em aplicações mais complexas e que requererem múltiplas tabelas de fatos que compartilham algumas dimensões. A Figura 3 demonstra visualmente essa representação.

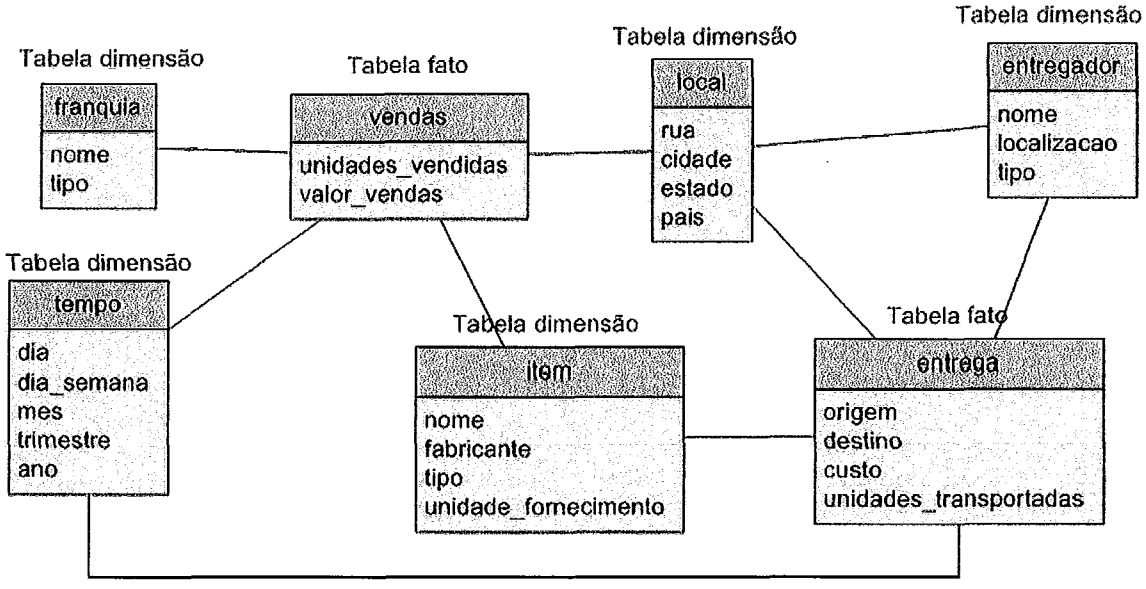


Figura 3 - Representação gráfica do modelo Constelação (Adaptado de: (HAN e KAMBER, 2001))

A modelagem multidimensional permite a visualização dos dados por diferentes pontos de vista, assim é possível utilizar ferramentas OLAP para materializar essas visões.

2.1.2 – Metadados

Metadados são dados capazes de descrever outros dados. Através dos metadados é possível saber onde os dados estão armazenados no ambiente operacional, e assim extraí-los para construção do *Data Warehouse* (INMON, 1997).

Os metadados de um *Data Warehouse* são utilizados como um dicionário e devem armazenar as seguintes informações (KIMBALL, REEVES *et al.*, 1998):

- Origem dos dados – informação da origem ou processo de geração dos dados. Esta informação deve ser única, ou seja, cada dado deve ter uma e somente uma fonte de origem;
- Fluxo do dado – para todo elemento de dado, deve-se identificar o fluxo de transformações, e quais os outros elementos de dados estão envolvidos nesta transformação;
- Formato dos dados – a informação do tamanho e tipo de dado;
- Nome e apelidos – os dados devem ser identificados por um nome ou apelido. Este nome ou apelido deve ser determinado a partir de um padrão, evitando assim ambigüidade;
- Regras de negócio – os dados que compõe as regras de negócios. A manutenção desses dados deve ser feita de forma consistente, de forma que o usuário possa obter facilmente definições para as informações desejadas. Referências a outros metadados que referenciam outros metadados devem ser evitadas, para manter a simplicidade das regras;
- Regras de transformação – são as regras de negócio codificadas, geradas no momento da extração, limpeza e agrupamento dos dados do ambiente operacional. Cada regra de transformação codificada deve estar associada a um metadado. Se mais de uma aplicação contiver a mesma regra de transformação, deverá ser garantido que estas sejam idênticas;
- Atualização dos dados – informação sobre a data de extração do dado do ambiente operacional. Esta informação pode ser utilizada para verificar a atualidade dos dados e a consistência da dimensão tempo do *Data Warehouse*;
- Requisitos de testes – dados sobre os critérios de avaliação de cada dado, como valores possíveis, intervalos de valores, e procedimentos para teste destes dados;

- Indicadores de qualidade dos dados – índices de qualidade, baseados na origem do dado, número de processamentos feito sobre esses dados, valores atômicos ou valores sumarizados e nível de utilização do dado;
- Gatilhos automáticos – informações sobre processos automáticos associados aos metadados definidos;
- Responsabilidade dos dados – informações sobre os dados do *Data Warehouse* e também o responsável pela entrada de metadados;
- Acesso e segurança – definição de usuários ou grupo de usuários bem como seus acessos a leitura, atualização, exclusão ou inserção de dados na base.

2.1.3 – Data *Marts*

Um Data *Mart* é um subconjunto de um *Data Warehouse* destinado a um determinado usuário ou grupo de usuários. Por exemplo: um Data *Mart* financeiro poderia armazenar dados consolidados dia-a-dia para um usuário gerencial e em periodicidades maiores (semana, mês, ano) para um usuário no nível da diretoria. Um Data *Mart* usualmente é organizado como o modelo estrela, e pode ser composto por um ou mais cubos de dados.

Kimball define que um *Data Warehouse* é uma coleção virtual de Data *Marts*. Esta abordagem *bottom-up* aponta a construção de um *Data Warehouse* a partir da elaboração de diversos Data *Marts* especializados (KIMBALL, 1996).

Inmon apresenta uma abordagem *top-down*, apontando o *Data Warehouse* como uma fonte de dados para Data *Marts* especializados (INMON, 1997).

2.2 – Data *Mining*

Data *Mining* (mineração de dados) é o processo automático ou semi-automático de exploração e análise de grandes quantidades de dados, objetivando a descoberta de regras e/ou padrões significativos (BERRY e LINOFF, 1997). Para Deogun esta grande quantidade de dados representa "minas" de conhecimento, onde repousam valiosas informações que podem ser encontradas através de técnicas e algoritmos adequados (DEOGUN, RAGHAVAN *et al.*, 1997).

O Data *Mining* é uma ferramenta útil para o *Data Warehouse*, permitindo a identificação de padrões de difícil percepção nos dados não agregados ou processados,

auxiliando o processo de tomada de decisões e até mesmo de previsão de futuras tendências no negócio em questão.

2.3 – On-*Line Analytical Process* – OLAP

OLTP (*On-Line Transaction Processing*) é um conjunto de ferramentas e tecnologias orientadas ao processamento de transações para entrada, recuperação e manipulação de dados do ambiente operacional. O comportamento e a organização dos dados no ambiente analítico requerem ferramentas específicas para o processamento analítico, denominadas OLAP (*On-line Analytical Processing*).

As ferramentas OLAP apresentam acesso e visualização dos dados de uma maneira analítica e multidimensional, realizando operações de classificação, comparação, taxas multidimensionais e variações percentuais, através de uma interface gráfica, interativa e amigável com o propósito de realizar a descoberta manual de conhecimento a partir dos dados armazenados. (KIMBALL, 1996)

A representação e armazenamento dos dados e derivações (agregações, totais, médias, etc.) em uma estrutura multidimensional justifica o desempenho bastante superior para as consultas que requerem análise dos dados como um todo, voltadas, portanto para o processo de tomada de decisão. OLAP é, portanto, uma ferramenta complementar à área de data *warehousing*, utilizando as mesmas operações básicas do cubo multidimensional onde o usuário de um sistema OLAP pode fazer seleções dos subconjuntos de dados que lhe interessam entre as múltiplas dimensões do cubo, inclusive a temporal (BOULLOSA, 2002; CHRYSAFIS, 2003; DOS SANTOS, 2001).

Existem três importantes categorias de OLAP, que diferem entre si, quanto à estrutura de armazenamento dos dados, mas mantém as mesmas capacidades de manipulação de dados multidimensionais. (CHRYSAFIS, 2003):

- OLAP Multidimensional (MOLAP) – os dados e agregações são armazenados em uma estrutura multidimensional, priorizando o desempenho no processamento dos cubos;
- OLAP Relacional (ROLAP) – os dados são armazenados em uma estrutura relacional ou acessados por consultas SQL. Esta abordagem visa às vantagens da tecnologia de banco de dados relacionais optando pelo processamento do cubo durante a sua utilização;
- OLAP Híbrido (HOLAP) – os dados são armazenados em um SGBD relacional e as agregações em uma estrutura multidimensional;

2.4 – *Web Mining*

Web Mining consiste na aplicação das técnicas de mineração de dados estudadas para *Data Warehouse*, utilizadas na Internet para aquisição de dados, busca de serviços e descoberta de padrões sobre os usuários. As técnicas para aquisição destes dados são divididas em algumas tarefas, são elas: a busca por documentos na Internet relevantes ao assunto; seleção das informações e pré-processamento – consiste na identificação automática e o pré-processamento de determinados dados; generalização – a descoberta automática de padrões em determinados sites da Internet bem como em múltiplos sites; e a análise dos dados e padrões identificados.

A natureza da informação procurada leva a classificação de *Web Mining* em três categorias: *Web Content Mining* – responsável pela descoberta de informações em dados, documentos e conteúdo da Internet; *Web Structure Mining* – responsável por descobrir um modelo representado pelo relacionamento entre as páginas na Internet; e *Web Usage Mining* – responsável por descobrir o fluxo de navegações dos usuários e acesso a dados. (COOLEY, MOBASHER *et al.*, 1997; KOSALA e BLOCKEEL, 2000)

"*Web Structure Mining* cria um modelo a partir da estrutura de relacionamentos (*links*) entre as páginas. Este modelo é usado para categorizar páginas e assim gerar informações sobre similaridades e relacionamentos entre diferentes sites da *Web*." (KOSALA e BLOCKEEL, 2000).

"*Web Usage Mining* tenta identificar padrões entre os fluxos de navegação dos usuários. Enquanto as outras duas categorias utilizam os dados reais e dados primários (conteúdo das paginas, documentos, etc.), esta categoria de mineração utiliza dados de registro de acesso dos servidores *Web*, registros de acesso à servidores de *proxy*, dados dos programas de navegação *Web*, lista de favoritos, cliques e todos os outros dados resultantes de iterações." (KOSALA e BLOCKEEL, 2000).

Web Content Mining descreve a descoberta de informações a partir de dados da *Web*. Na Internet os dados estão disponíveis em diferentes tipos de serviços e diferentes representações, e grande parte desses dados consistem em dados não estruturados como trechos de textos, documentos HTML[1] (KOSALA e BLOCKEEL, 2000).

---

[1] Hyper Text Markup Language – Linguagem de marcação para representação de documentos na Internet.

A sessão seguinte descrever o projeto *e.dot*, uma ferramenta que aplica técnicas de *Web Content Mining* para recuperação de dados da Internet.

2.4.1 – Projeto *e.dot*

Este projeto trata a construção automática de *Data Warehouses* temáticos integrando documentos XML, responsáveis por descrever o domínio da aplicação, a repositórios relacionais utilizando Ontologias.

O objetivo do projeto *e.dot* é enriquecer um repositório de dados existente através de contínuas buscas por dados na Internet. Os resultados das buscas são armazenados em formato XML e deve ser consultado simultaneamente pela interface relacional do *e.dot* e pela interface relacional existente no SGBD (GAGLIARDI, HAEMMERLÉ *et al.*, 2005).

O projeto *e.dot* apresenta dois métodos para integração de dados em uma abordagem relacional. O primeiro método permite integrar dados XML heterogêneos através de visões (*views*) no esquema do banco de dados relacionais. Essas visões não são materializadas, e as consultas aos documentos XML são realizadas por um algoritmo para reescrita das consultas SQL em consultas XQuery (as linguagens SQL e XQuery são abordadas no Apêndice A). Para realizar estas transformações o módulo mediador possui todos os DTD que descrevem os documentos XML envolvidos (*Document Type Definition* – uma linguagem para descrição de esquemas de documentos XML – é abordada no Apêndice A).

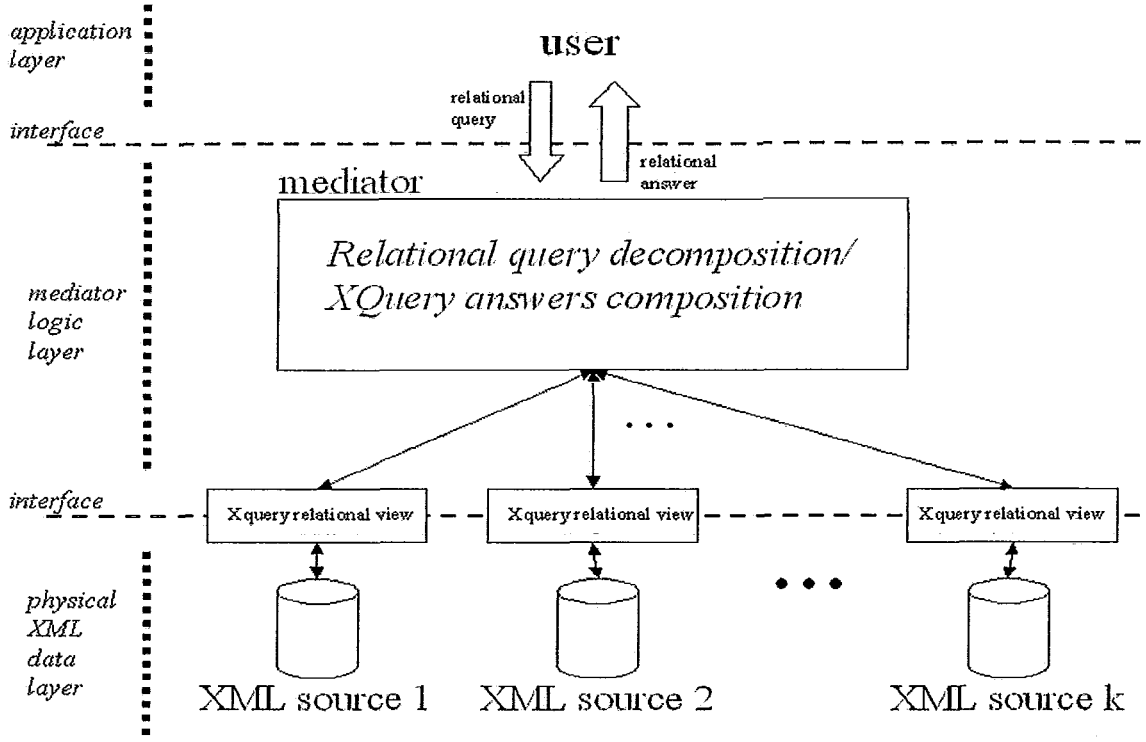


Figura 4 - Esquema do projeto *e.dot* (Fonte: (GAGLIARDI, HAEMMERLÉ *et al.*, 2005))

A Figura 4 ilustra a transformação da consulta relacional em consultas XQuery, que serão executadas em repositórios de documentos XML. Para construir a resposta o módulo mediador (representado na imagem como *mediator logic layer*) realiza o processo inverso, transformando fragmentos de documentos XML obtidos como resposta num formato tabular comum de sistemas relacionais.

O segundo método para integração de dados em uma abordagem relacional concentra-se nas tabelas de dados existentes nos documentos encontrados na *Web*. Este método aproveita a presença de estruturas relacionais para transformações automática desses dados, de modo a fazê-los compatíveis com o esquema do banco de dados atual. Estas tabelas são enriquecidas semanticamente através de marcações e valores oriundos de ontologias da aplicação.

A pesquisa por documentos na Internet é realizada através da WeQueL, uma linguagem declarativa baseada em atributos e valores, que permite a descrição de suas necessidades combinando palavras-chave na consulta. Nesta linguagem, cada propriedade procurada nas páginas pode receber uma lista de palavras-chave. A Figura 5 exibe um exemplo de consulta em WeQueL que representa uma busca por figuras de Formula 1.

Formula one pictures
(page_title = photo gallery 'f1' , schumacher picture , montoya picture . . .) ∧ (URL = photo 'f1' , image 'f1' , coulthard picture , villeneuve picture . . .) ∧ (Incoming_Links = photo formule 1 , image formule 1 , grand prix photo . . .) ∧ (Outgoing_Links = suivant , precedent , previous , back . . .) ∧ (MIME = text/htm)
∨
(URL = f1 photo , image f1 , grand prix photo , gallerie f1 . . .) ∧ (Incoming_Links = formula one pic , f1 picture , formula one gallery . . .) ∧ (MIME = image)
∨
(URL = photo ferrari f1 , photo mac laren f1 , ralf picture . . .) ∧ (title_Incoming_Page = formula one gallery , photo ferrari 'f1' . . .) ∧ (MIME = image , text/htm)

Figura 5 - Consulta em WeQueL por figuras relativas a Formula 1 (Fonte: (MEZAOUR, 2003))

Após a recuperação dos documentos da Internet, o *e.dot* utiliza o subprojeto *e.dot Filter*, capaz de filtrar o conteúdo heterogêneo encontrado extraindo apenas os dados necessários para o *Data Warehouse* temático (MEZAOUR, 2004).

A filtragem de documentos na *Web* parte da análise de documentos fornecidos por especialistas que apresentam dados do domínio da aplicação representada pelo *Data Warehouse* em questão. Estes documentos são arquivos HTML contendo no mínimo uma tabela HTML com o máximo de palavras-chave possível extraídas da Ontologia que descreve o domínio em questão. Com este conjunto de palavras-chave, o programa realiza uma combinação entre os elementos e manualmente, os especialistas removem os elementos irrelevantes. (MEZAOUR, 2004)

Com este conjunto de palavras-chave e termos relevantes ao assunto em questão, o programa é capaz de realizar uma busca por documentos na Internet e classificá-los de acordo com a quantidade de termos encontrados. Após esta classificação, os documentos mais relevantes são escolhidos, levando em consideração a quantidade de palavras-chave encontradas.

2.4.2 – Ontologia para Modelo de Dados

Na Filosofia a palavra Ontologia é associada a um dos ramos da Metafísica, que se destina ao *Estudo da Natureza da Existência* (WIKIPEDIA, 2005a). Neste trabalho a palavra ontologia define um vocabulário comum para pessoas ou sistemas que precisem compartilhar informações em um domínio. Inclui definições interpretáveis por homens e máquinas de conceitos básicos e as relações entre eles. Em uma definição mais formal, ontologia é uma descrição de conceitos em um domínio (conhecido por **classes**), das **propriedades** de cada conceito descrevendo suas funcionalidades, e **restrições** (ou regras) sobre essas propriedades. (NOY e MCGUINNESS, 2001)

As aplicações de ontologias são diversas e atualmente é o principal componente da WEB Semântica, cuja proposta é fazer com que o conteúdo da Internet seja acessado e compreendido por humanos e sistemas computacionais. Podemos encontrar ontologias em sistemas para processamento de linguagem natural, gestão do conhecimento, CSCW, entre outros. (ANTONIOU e VAN HARMELEN, 2004; PASSIN, 2004)

Atualmente algumas linguagens são utilizadas para representação das ontologias, dentre os padrões livres, podemos destacar XML, XML *Schema*, RDF, RDF *Schema* e OWL. OWL é descrição de vocabulários ricos de uma linguagem para descrever propriedades e classes, e suas relações (ANTONIOU e VAN HARMELEN, 2004).

Para criar uma nova ontologia é necessário definir novas classes, organizá-las hierarquicamente, definir e descrever os valores permitidos para cada propriedade e criar instâncias para essa nova ontologia, para tanto é necessário conhecimento sobre os conceitos a serem modelados. Entretanto muitas entidades e organizações já aplicaram esforços no desenvolvimento de ontologias e formaram uma grande biblioteca que está disponível em diversos sites e repositórios, assim, antes de iniciar uma nova ontologia é interessante verificar a existência de alguma que satisfaça a necessidade.

2.5 – Considerações sobre *Data Warehouse*

As ferramentas de *Data Warehouse* são úteis à aquisição de informações de dados. As ferramentas OLAP apresentam a característica descritiva como ponto forte, enquanto na mineração de dados o ponto forte é a descoberta automática de padrões.

As ferramentas de recuperação e processamento de documentos da Internet para a agregação ao repositório de dados de *Data Warehouses* existentes permitem a criação do seguinte quadro comparativo, traçando as suas características, bem como àquelas pertencentes ao Projeto Números.

Tabela 1 - Comparação entre recursos do *e.dot* e Projeto Números

Característica / Projeto	***e.dot* + Subprojetos**	**Projeto Números**
Uso de Ontologias para enriquecer consulta	Sim, mas requer validação manual.	Sim, durante a construção da consulta e para aquisição de dados para os repositórios.

Linguagem para consulta de documentos na *Web*	WeQueL	FormatoComum (enriquecido com Ontologias e marcações Dublin Core)
Assistente para construção da consulta	Não disponível.	Sim
Necessidade de validação dos dados pelo usuário	Alta	Baixa. A construção da consulta determina o escopo do resultado. Há filtros para delimitar a qualidade dos dados e outros atributos.
Esquema de representação dos dados nos repositórios	Deve ser fornecido pelo usuário	Construção automática a partir da Ontologia ou consulta

Capítulo 3 – Dados na Internet

"Apesar de a Internet ser o canal de informações mais recente é também o que apresenta o crescimento mais rápido e o recurso com maior número de usuários". A cada dia uma grande quantidade de dados é produzida nas trocas de e-mails, novas páginas de conteúdo e através dos servidores de conteúdos dinâmicos que expõem dados de bancos de dados. (LYMAN, VARIAN *et al.*, 2003).

Em um estudo sobre a quantidade e tipos de dados na Internet, a Universidade de Berkeley afirma a existência de aproximadamente 530 terabytes de dados originais disponíveis na Internet. Este estudo detalha ainda a composição deste conteúdo entre dados estáticos (como as páginas HTML), dados gerados dinamicamente (proveniente do processamento de informações de banco de dados), dados em e-mails (apenas os originais, descartando cópias) e dados provenientes de programas de mensagens instantâneas. (LYMAN, VARIAN *et al.*, 2003).

Até o ano de 2003, mais de 300 milhões de pesquisas eram realizadas diariamente nos doze sites de buscas mais utilizados na Internet. Atualmente a entidade *Whois.net*, responsável por domínios americanos, registra mais de 90 milhões de domínios ativos.

3.1 – O formato dos dados na Web

Os dados estão disponíveis em diferentes formatos de arquivos cada qual com um objetivo e particularidade. O conhecimento da estrutura desses arquivos é útil para a construção de ferramentas capazes de compreender a organização e principalmente como extrair os dados destes arquivos.

Um estudo realizado em 2003 utilizando uma amostragem de aproximadamente 10 mil domínios da Web, com cerca de 61 milhões de URL e um total de 33.1 Gigabytes decompôs o conteúdo da Web em tipos de arquivos e apresentou o gráfico da Figura 6.

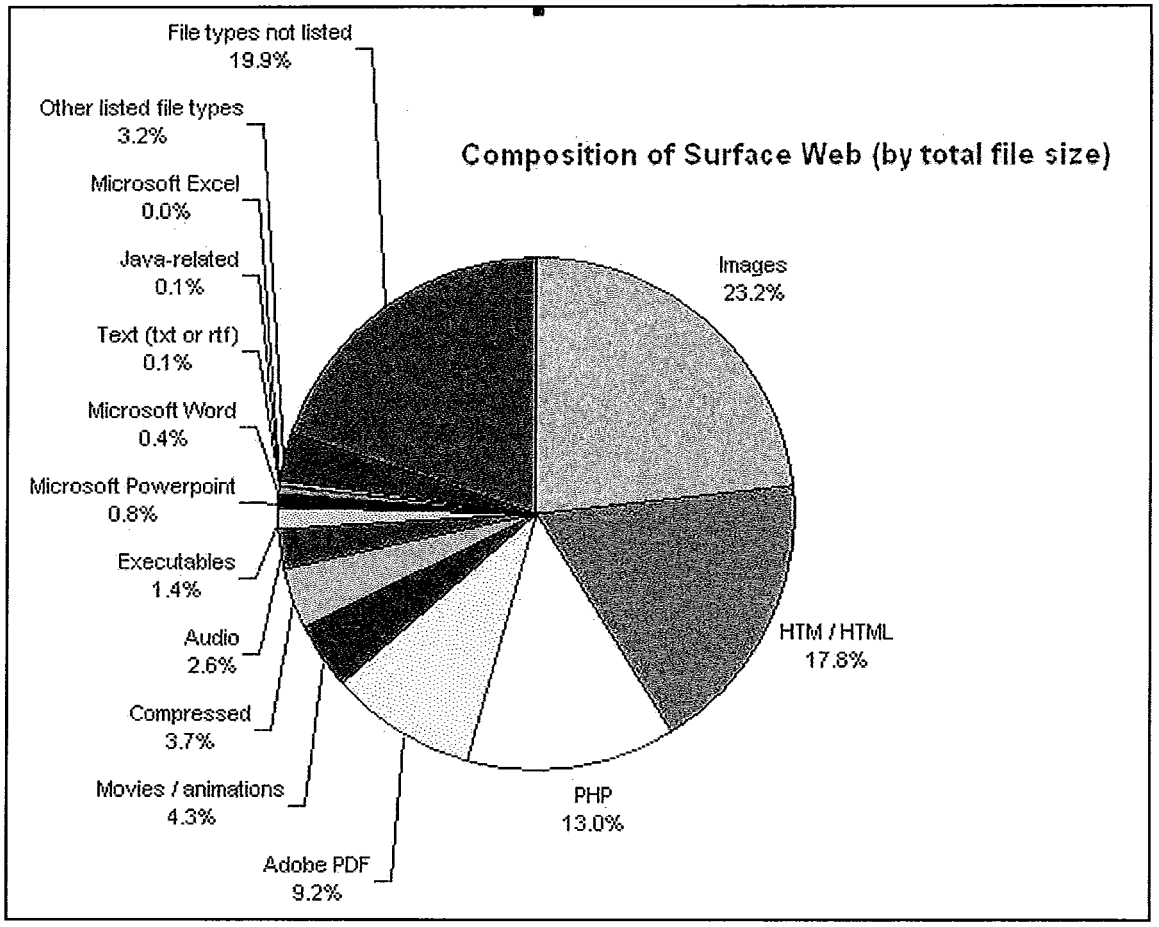


Figura 6 - Composição dos dados na Internet (LYMAN, VARIAN *et al.*, 2003)

A figura anterior exibe uma composição aproximada dos dados da Internet. A pesquisa apresentada pela Universidade de Berkeley classifica a acessibilidade dos dados na Internet em dados de superfície (*Surface Web*), disponíveis de forma estática e dados profundos (*Deep Web*), referindo-se aos dados armazenados em bancos de dados que são expostos apenas através da execução de consultas pelos usuários. (LYMAN, VARIAN *et al.*, 2003)

Os documentos HTML são os principais documentos publicados na Internet, são arquivos de texto semi-estruturados que possuem marcações delimitando o texto a ser exibido e informações de como os navegadores *Web* devem exibir esse texto. Este tipo de documento armazena dados em forma de texto livre com marcações visuais (como negrito, itálico, sublinhado), listagens hierárquicas e tabelas.

A extração de dados de documentos HTML é estudada pelo uso de *Web Mining*, e existem diversas técnicas para a extração, cada uma objetivando um tipo diferente de informação. A extração de dados de uma tabela envolve a identificação das marcações

visuais e avaliação de quais são os rótulos e o que são células. (KOSALA e BLOCKEEL, 2000).

Cerca de 10% dos dados apresentados na pesquisa são dados de formatos proprietários, como documentos Microsoft Word, RTF, planilhas eletrônicas Microsoft Excel e documentos Adobe PDF. As planilhas eletrônicas apresentam os dados organizados em células o que facilita a sua extração, e ainda armazenam informações importantes como fórmulas para calcular o valor de células, textos que servem como rótulos.

Os outros formatos de documentos proprietários citados na avaliação são basicamente documentos de textos, que armazenam dados não estruturados, apresentando tabelas e listagens de forma similar ao HTML.

A pesquisa da Universidade de Berkeley não apresenta estatísticas sobre documentos XML (ver Apêndice A), um padrão de troca de dados entre aplicações, mas hoje é comum que os sistemas *Web* disponibilizem seus dados utilizando este padrão de linguagem de marcação também semi-estruturada.

3.2 – Uma avaliação empírica da disponibilidade de dados numéricos sobre o Brasil na Web

A busca por dados na Internet pode ser uma atividade trabalhosa, esta seção demonstra sites da Internet que armazenam dados estatísticos sobre o Brasil, detalhando como essas informações podem ser encontradas e como os dados são armazenados nos servidores.

A utilização de dados estatísticos nesta avaliação é interessante pela grande quantia de sites que armazenam estes dados. Esta grande diversidade e disponibilidade de dados levam muitas vezes a uma disparidade das informações encontradas, cabendo ao usuário avaliar as fontes mais confiáveis. Os dados são encontrados em diferentes categorias de sites, a seguir estas categorias são descritas e os tipos de dados encontrados são detalhados.

Um ponto de partida na busca por informações estatísticas são os sites de Informações Oficiais Governamentais. No Brasil os sites do governo disponibilizam informações coletadas de agências governamentais específicas para aquisição destas informações, como o Instituto Brasileiro de Geografia e Estatística – IBGE. No IBGE a preocupação é oferecer uma sumarização das informações em níveis como municípios

(ou micro regiões) até a totalização de um determinado dado considerando o país inteiro. Esses dados estão distribuídos em diversos formatos como documentos HTML, planilhas eletrônicas, documentos PDF e páginas dinâmicas, que expõem conteúdo de banco de dados.

Há informações de origem não governamental fruto do trabalho de associações, sindicatos, fundações e das Organizações Não-Governamentais que produzem dados estatísticos e distribuem o fruto de suas pesquisas na Web. Um exemplo é a fundação FGV (http://fgvdados.fgv.br/) que realiza pesquisas econômicas produzindo dados sobre indicadores econômicos brasileiros e disponibilizando em forma de gráficos, sendo possível também a geração de páginas com conteúdo oriundo do processamento de parâmetros fornecidos pelo usuário.

As Entidades Internacionais de pesquisas e informações também são grandes produtoras de dados, onde as informações referentes a outros países são em maioria a totalizações dos dados produzidos pelos próprios países ou através de pesquisas independentes. A entidade americana Central Intelligence Agengy – CIA (https://www.cia.gov/cia/publications/factbook/) reúne informações sobre países disponibilizando-as como páginas HTML e também em PDF.

As grandes distribuidoras de dados e informações são as entidades Jornalísticas, e a Web amplificou este processo de distribuição, tornando os dados ainda mais acessíveis. Os canais jornalísticos da Internet a cada dia divulgam uma grande quantidade de dados, produzidos por entidades de pesquisas, governamentais e também fruto da própria entidade. Observando a página da Folha de São Paulo (http://www.folha.com.br) é possível obter uma série de dados sobre o Brasil, mas em sua maioria estão distribuídos de forma não estruturada como em documentos HTML, imagens e apresentações visuais.

As Entidades de Pesquisa Científica são grandes produtoras de dados, investindo seus esforços na produção e desenvolvimento de informações e dados sobre as mais diversas áreas. No site da Universidade Federal do Rio de Janeiro é possível encontrar nas dissertações e teses, dados estatísticos de forma tabular e também de forma não estruturada.

A produção e distribuição de dados também é fruto de movimentos colaborativos, organizados e mantidos por instituições e pessoas. A Internet permitiu a reunião de diversos serviços de colaboração para a construção de bases de dados, como a WIKIPEDIA (http://wikipedia.org) e movimentos para o compartilhamento livre dos

dados existentes como o Creative Commons (http://creativecommons.org/). Os dados dispostos na WIKIPEDIA contam com o esforço e comprometimento dos usuários para manter os dados fiéis, imparciais e confiáveis. Esta característica colaborativa leva a uma grande disparidade da qualidade dos dados distribuídos e também dos formatos de dados disponíveis.

Dados numéricos também são divulgados em páginas pessoais, principalmente nos Blogs (*Web Logs*), que funcionam como diários eletrônicos. Diversas pessoas, incluindo pesquisadores, professores, jornalistas e pessoas sem formação acadêmica escrevem em seus Blogs pessoais, divulgando dados obtidos dos mais variados meios.

A busca por dados numéricos mostra claramente a grande massa de dados disponíveis, bem como os diversos níveis e critérios para distribuição e qualidade destes dados. A Tabela 1 apresenta um resumo dos sites visitados apontando as características dos dados de cada um deles.

Tabela 2 – Sites produtores e/ou distribuidores de dados numéricos.

Dados / Fontes	Produção de dados / Distribuição	Acessibilidade	Formato dos Dados Disponíveis
Entidades Governamentais (IBGE)	Produção e distribuição, por órgãos destinados à pesquisa.	Simples, os dados estão categorizados e normalmente há um sistema para recuperação.	Documentos HTML, arquivos CSV, sistema para cruzamento de dados.
Entidades não-Governamentais (FGV)	Produzidos e distribuídos como pesquisas encomendadas pela própria entidade.	Os dados estão categorizados e normalmente há um sistema para recuperação. É possível comprar pesquisas mais recentes e planos para acesso a um maior número de dados.	Documentos HTML, Imagens, Gráficos, há também um sistema para cruzamento de dados.
Entidades Internacionais (CIA)	Produção e distribuição.	Dados organizados por países. Acesso simples.	Documentos HTML e páginas dinâmicas recuperando informações de bancos de dados
Entidades Jornalísticas (Folha de São Paulo)	Foco na distribuição de dados.	Em sua grande maioria os dados estão em textos não estruturados, mas há também tabelas e demonstrativos.	Documentos HTML, imagens, gráficos, tabelas e animações.

Entidades de Pesquisa Científica (UFRJ)	Produção e distribuição	Dados disponíveis em dissertações, teses e relatórios técnicos.	Documentos HTML, PDF, documentos de texto e sistemas para recuperação de dados.
Movimentos Colaborativos (WIKIPEDIA)	Foco na distribuição de dados.	Os dados estão disponíveis em textos e tabelas.	Documentos HTML com imagens, tabelas e muitas informações não estruturadas.
Outros (Blogs de jornalistas)	Foco na distribuição de dados.	As ferramentas de construção de blogs permitem o acesso aos textos mais antigos. Os dados normalmente são não estruturados em meio aos textos dos usuários.	Documentos HTML.

Capítulo 4 – Projeto Números

Este capítulo apresenta o Projeto Números, uma arquitetura de um sistema de busca que através do enriquecimento da consulta realiza buscas em bases de dados heterogêneas, como a Internet. As próximas seções descrevem a especificação, a arquitetura, limitações e tecnologias que são empregadas na implementação do projeto.

4.1 – Visão Geral

O Projeto Números é um sistema para construção e enriquecimento semântico de consultas com resultados numéricos. Este trabalho define sua arquitetura e implementa parte deste sistema.

Apresentamos neste trabalho a especificação da arquitetura e uma interface para construção de consultas por exemplo (*Query By Example* – QBE), onde o usuário é capaz de fornecer dados em uma tabela parcialmente preenchida, desejando que o restante da tabela seja preenchida pelo sistema. A Figura 7 exibe uma tela da interface gráfica do Projeto Números, destacando determinadas regiões e seus objetivos.

A consulta do usuário é construída numa linguagem chamada Formato Comum, que armazena além dos dados de exemplo, dados de ontologias associadas à consulta. Esta linguagem é também a linguagem para troca de dados entre as diversas camadas do sistema. O Formato Comum e outros detalhes são abordados no decorrer deste trabalho.

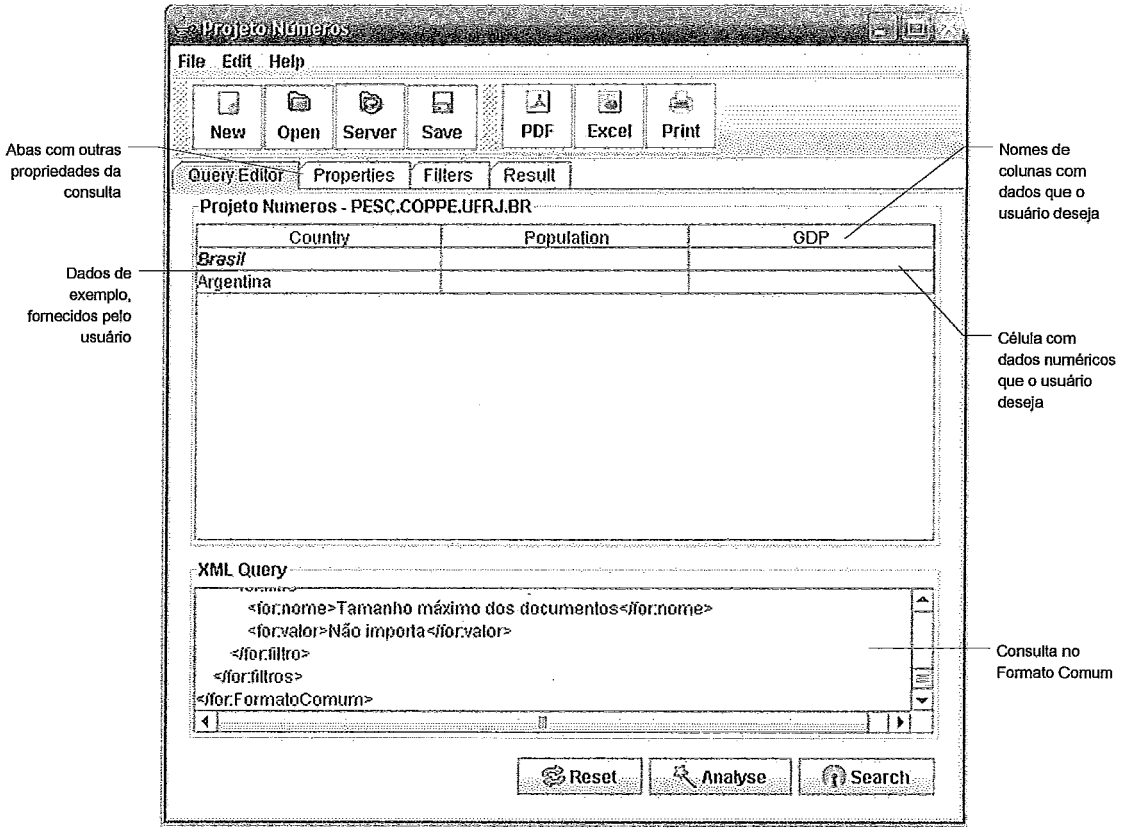


Figura 7 - Tela da Interface gráfica do Projeto Números

4.2 – Especificação

A delimitação de escopo, funcionalidades e requisitos mínimos são fundamentais para a compreensão de um sistema computacional, e através de seu detalhamento é possível determinar quais as melhores soluções para a resolução de problemas já conhecidos ou ainda levantar um questionamento sobre problemas sem respostas satisfatórias.

Esta sessão descreve os requisitos, restrições e funcionalidades impostas ao sistema, de forma que seja possível alcançar os objetivos propostos no Capítulo 1.

4.2.1 – Requisitos básicos

Para alcançar os objetivos, os seguintes requisitos básicos devem ser satisfeitos:

- O usuário deverá ser capaz de construir e manipular visualmente uma tabela com dados e títulos nas colunas que representará a sua a sua busca por dados;

- O sistema deverá permitir a consulta sobre informações numéricas disponíveis em documentos ou outras fontes de informação disponíveis na Internet;
- O sistema deverá permitir a exportação dos resultados da consulta em outros formatos de documentos;
- O sistema deverá realizar as buscas síncrona e assincronamente, de acordo com a dificuldade e duração prevista para a consulta, fornecendo informações, quando necessário, ao usuário;
- O sistema deverá armazenar os dados encontrados na Internet, e seus metadados, em um repositório de dados local, de modo a garantir sua persistência e facilitar consultas posteriores sobre os mesmos dados;
- O sistema deverá encontrar e recuperar dados disponíveis nas forma não-estruturada, semi-estruturada ou estruturada;
- O sistema deverá ser extensível ao ponto de recuperar informações de outros meios ou bases de dados, mediante a implementação de algumas interfaces fornecidas ao desenvolvedor;
- O sistema deverá ser capaz de recuperar, por meio de mediadores ou interfaces especificamente desenvolvidas, de forma *ad hoc*, dados disponíveis de forma não padronizada;
- O sistema deverá manter metadados que permitam avaliar a qualidade dos dados obtidos mediante diferentes procedimentos;
- O sistema deverá apoiar o usuário na construção da consulta, por meio de auxílio inteligente não só ao uso da ferramenta, mas também à exploração da área de aplicação;
- O sistema deverá permitir o cálculo de valores derivados a partir do resultado obtido;
- O sistema deverá possuir mecanismos de recuperação de metadados durante a busca das informações numéricas, com o objetivo de fornecer aos usuários a avaliação da qualidade dos dados que foram recuperados;
- O usuário deverá ser capaz de associar metadados a todo e qualquer campo da consulta construída;
- O usuário deverá ser capaz de especificar o domínio e o tipo dos dados de cada coluna;

- O sistema deve exibir o resultado da consulta em uma estrutura tabular composta por células, similar à consulta e deve apresentar marcações visuais referentes à qualidade dos dados;
- Caso o sistema encontre mais de valor uma mesma célula, deve através de métricas de qualidade, avaliar o melhor dado para ser exibido e inserir uma marcação visual na célula, para que o usuário identifique que aquela célula possui mais de um dado;
- Em células com mais de um dado, o usuário deve ser capaz de visualizar os dados disponíveis e suas diferentes origens, e escolher uma das opções como padrão, o sistema deverá armazenar esta configuração;
- O sistema deverá utilizar ontologias para vincular semântica às colunas;
- O sistema deverá permitir ao usuário escolher qual a folha de uma determinada ontologia que ele considera melhor para a coluna em questão;
- O usuário poderá editar e criar novas ontologias;
- O sistema deverá ser capaz de compartilhar as ontologias editadas pelos usuários e manter um repositório comum para processamento das consultas;
- O usuário deve ser capaz de visualizar a consulta gerada, tanto em forma tabular (representação gráfica) quanto na linguagem de representação interna;
- O sistema deve ser capaz de gerar uma consulta em uma linguagem comum a todos os outros módulos mediadores (*wrappers*);
- O sistema deverá ser capaz de armazenar consultas realizadas como fonte de informações sobre dados e manter um repositório de conhecimento;
- O sistema poderá gerar ontologias a partir de dados contidos nas consultas;
- O sistema deve permitir ao usuário especificar palavras chave que poderão auxiliar no processo de pesquisa;
- O sistema deve suportar o acesso simultâneo de clientes heterogêneos, atendendo suas requisições de forma síncrona ou assíncrona e reutilizando perguntas e respostas de consultas disparadas por outros usuários;

- O sistema deve realizar otimizações e expansões sobre as consultas do usuário buscando eficiência, confiabilidade, atualização e crescimento da base de conhecimento.

4.2.2 – Requisitos da arquitetura

A arquitetura esperada pelo Projeto deve atender aos seguintes requisitos de caráter econômico, ideológico e de propriedade intelectual:

- O desenvolvimento seguirá o paradigma da Orientação a Objetos;
- A linguagem utilizada para o desenvolvimento será a Java 5.0;
- A modelagem deverá ser feita em UML;
- Os componentes de software devem ser considerados implementações de referência, permitindo a extensão de funcionalidades e diferentes implementações;
- Será usado apenas software livre e variações similares (software de código aberto, licença pública, softwares gratuitos, etc.), objetivando ainda a disponibilidade do código fonte e custo zero para aplicações não comerciais.

4.3 – Arquitetura

O Projeto Números é um projeto extenso e que é composto por vários trabalhos acadêmicos. Este trabalho apresenta a arquitetura do projeto, com o objetivo de definir seu escopo e servir como referência aos trabalhos futuros. Até a data de conclusão deste trabalho, o Projeto Números foi dividido em quatro trabalhos acadêmicos (três dissertações e uma tese) com as seguintes responsabilidades:

- Coleta de Dados – trabalho que estuda e implementa os mecanismos de busca, identificação e recuperação de documentos;
- Extração de Dados – trabalho que estuda a identificação e extração de dados dos documentos encontrados pelo trabalho de Coleta de Dados.;
- Qualidade de Dados – trabalho que qualifica os dados encontrados, determinando confiabilidade, atualidade e outros atributos relativos à qualidade do dado;
- Interface de construção de consultas semanticamente ricas e organização dos repositórios de dados – o presente trabalho, que apresenta uma

interface para construção e manipulação de consultas enriquecidas e prepara os repositórios de dados para o armazenamento e reaproveitamento dos dados encontrados.

Os requisitos e funcionalidades esperadas servem como base para a especificação de uma arquitetura que provê compartilhamento de recursos e expansibilidade por todo o sistema e suas extensões.

Diante dos requisitos e das funcionalidades esperadas, o Projeto Números foi dividido em três camadas com responsabilidades distintas buscando a simplicidade para o desenvolvimento e compreensão do sistema. Esta sessão descreve as camadas de Interface com o Usuário, a camada de Armazenamento de Dados e Conhecimento e a Camada de Coleta de Dados, e após a apresentação de cada componente, será apresentado suas entradas e saídas de dados.

A arquitetura aqui apresentada é dividida em camadas, onde cada uma abriga componentes lógicos responsáveis por uma atividade distinta e importante para o funcionamento do sistema. A Figura 8 mostra um esquema com estas camadas e seus componentes e a Figura 9 exibe um diagrama de classes em notação UML desta arquitetura, ambas destacam os componentes abordados neste trabalho.

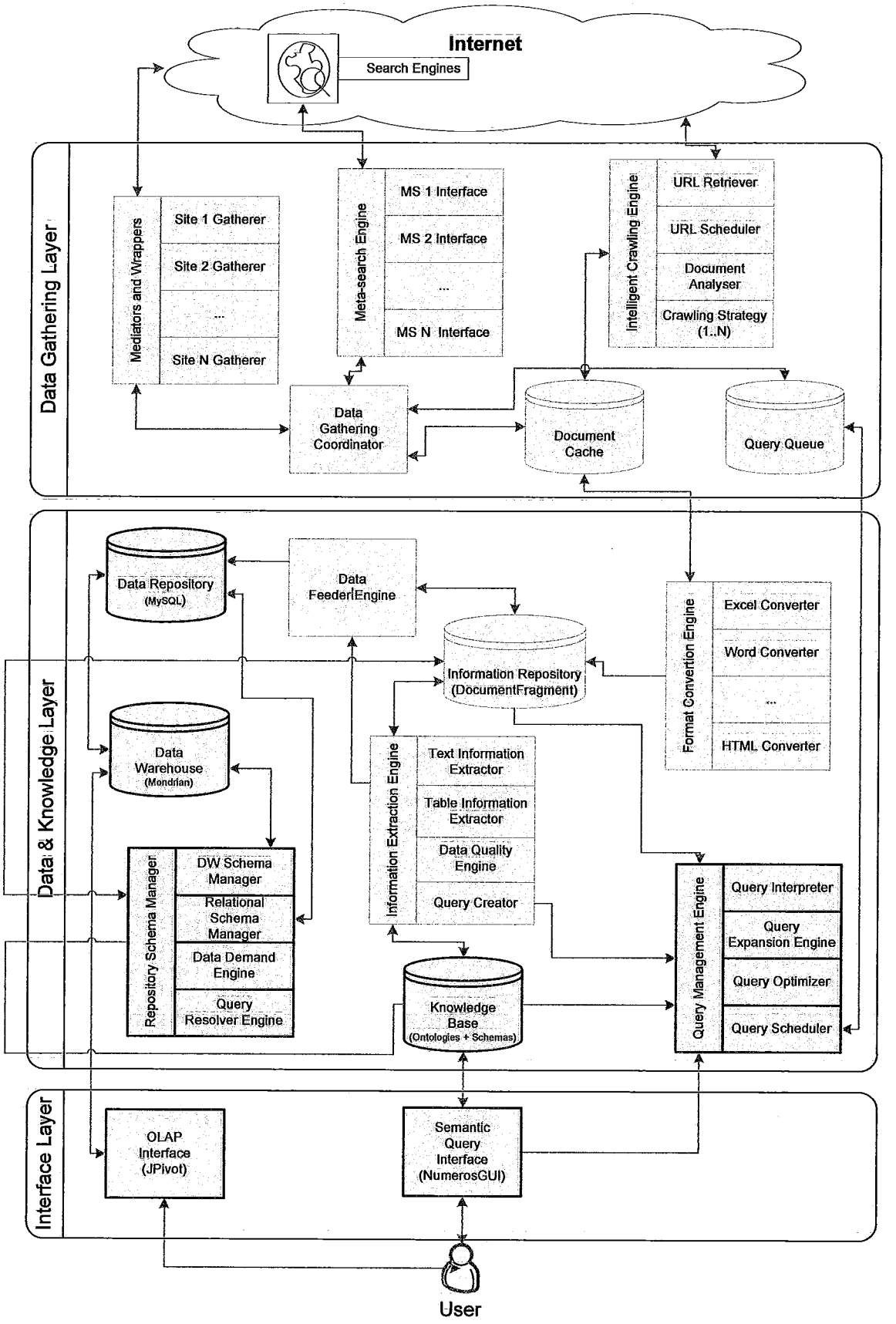


Figura 8 – Representação gráfica das camadas da arquitetura do Projeto Números. Os componentes em destaque representam o foco deste trabalho.

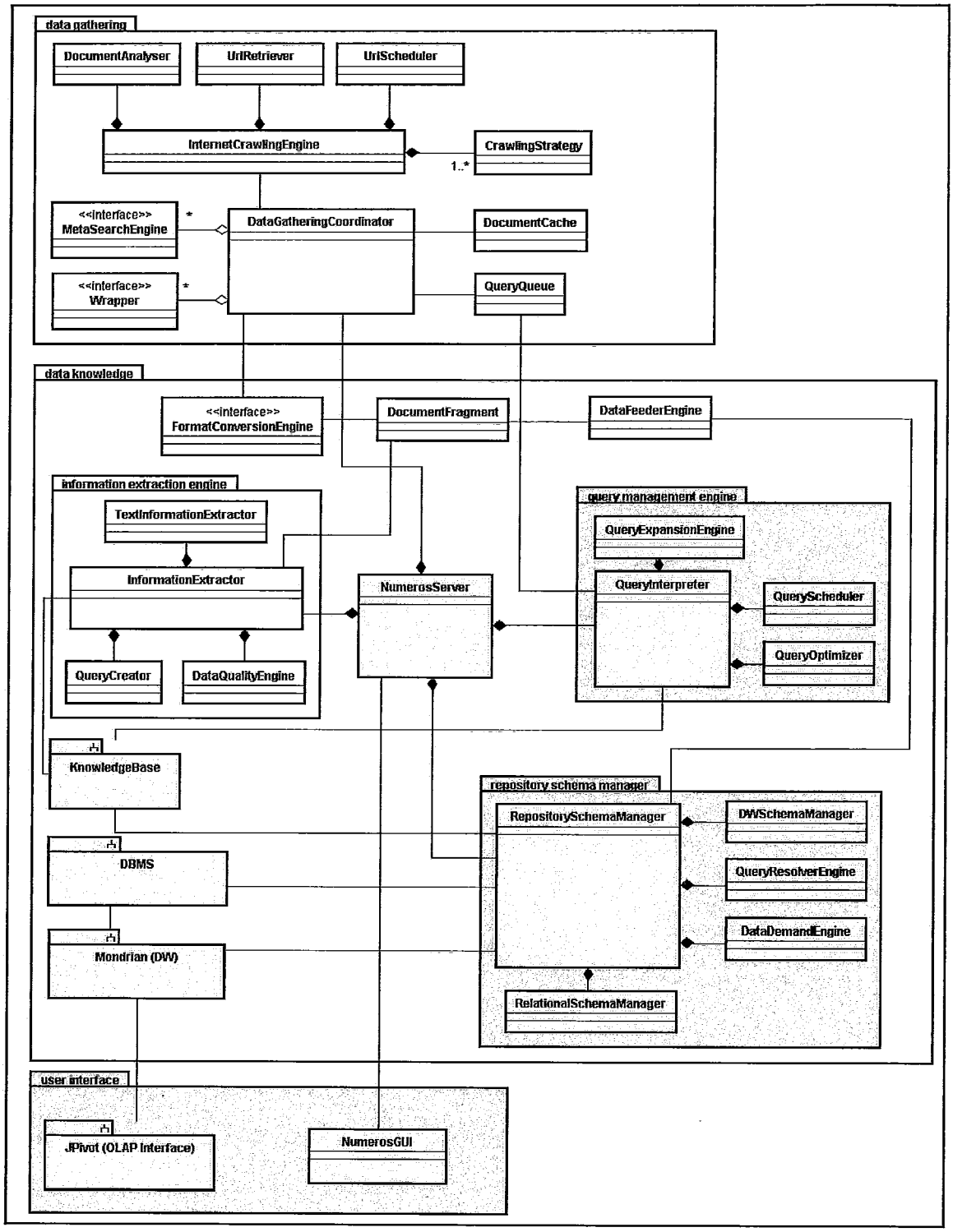


Figura 9 – Diagrama de Classes (UML 1.4) representando pacotes, classes e componentes da arquitetura do Projeto Números, destacando os componentes focados neste trabalho.

4.3.1 – Formato Comum

A comunicação entre os módulos e as camadas do Projeto Números deve ser realizada de forma simples e através de uma linguagem comum que deve ser capaz de

armazenar as informações da consulta do usuário, bem como todas as informações adicionadas pelos módulos onde a consulta for utilizada.

A troca de mensagens entre camadas ou sistemas de computação é um assunto muito discutido e optamos por utilizar uma solução padronizada, ou seja, a troca de documentos XML. Os documentos XML são documentos de textos simples, restritos por um conjunto bem definido de regras de formatação. Para restringir o conteúdo de um documento XML, é necessária a utilização de XML *Schemas* (ver Anexo), solução que foi adotada no projeto para construção do Formato Comum.

O Formato Comum é um documento XML *Schema* que define o formato das mensagens trocadas entre os módulos do Projeto Números, os seguintes requisitos são mínimos para este formato:

- Representar a consulta feita pelo usuário;
- Armazenar todas as informações adicionadas à consulta no decorrer de sua vida no Projeto Números;
- Armazenar informações sobre o usuário;

Além dos requisitos mínimos, o Formato Comum deve atender aos seguintes requisitos para prover uma comunicação funcional entre as camadas e representar efetivamente as consultas:

1. O Formato Comum deve manter os dados e os metadados de uma consulta de forma agregada;
2. O Formato Comum deve utilizar um padrão de metadados como o *Dublin Core Metadata* (DCMI);
3. O Formato Comum deve ser facilmente extensível e buscando manter compatibilidade com versões antigas do Projeto Números;
4. O Formato Comum deve ser mantido, ou facilmente convertido, em um formato legível por humanos;
5. O Formato Comum deve manter a memória completa de cada consulta, referenciando as Ontologias, os metadados, as informações de filtros, os processos de transformação e obtenção utilizados, e as pessoas envolvidas no processo.

4.3.2 – Camada de Interface com o Usuário (*User Interface*)

Esta camada representa a interface gráfica com o usuário e componentes adicionais para a construção destes. É responsável pela principal interface com o usuário, uma interface gráfica rica que permite a construção e a recuperação de resultados. Esta aplicação provê mecanismos de exibição dos metadados referentes à qualidade dos dados encontrados e filtros para atributos dos dados.

A camada de interface com o usuário (*User Interface Layer*) é a porta de entrada do usuário ao sistema. Seus dois componentes representam as interfaces gráficas com o usuário, enquanto a Interface de Consulta Semântica (*Semantic Query Interface*) representa um construtor de consultas, a Interface OLAP (*OLAP Interface*) representa uma interface para consultas multidimensionais *ad-hoc* nos dados existentes no repositório de dados.

A interface gráfica permite a manipulação de consultas no Formato Comum permitindo ao usuário representar a sua consulta visualmente, aplicando o conceito de consultas a partir de um exemplo. Esta interface é capaz de gerenciar Ontologias e também a recuperação do resultado de consultas armazenadas na camada de Dados e Conhecimento.

4.3.2.1 – Interface OLAP (OLAP *Interface*)

Este componente apresenta uma interface gráfica onde o usuário poderá executar consultas ao repositório de dados como em um cubo multidimensional. O usuário poderá escrever consultas OLAP, interagindo com o repositório de dados como um *Data Warehouse* com dados provenientes de todas as consultas realizadas na Interface de Consulta Semântica.

Nesta interface o usuário é capaz de realizar consultas utilizando uma ferramenta de construção de consultas, ou um interpretador de alguma linguagem de consulta multidimensional, como o *Multidimensional Expressions* – MDX. Esta interface provê uma visão multidimensional dos dados armazenados, onde será possível utilizar os campos de ontologias armazenadas como Dimensões e o resultado das consultas no Formato Comum como as tabelas de Fatos.

Responsabilidade: Uma interface gráfica para a construção de consultas multidimensionais e exibição dos resultados.

Saída: Consulta multidimensionais enviadas para o *Data Warehouse* do sistema.

Situação: Implementado pelo JPivot (*vide* 5.4 – JPivot).

4.3.2.2 – Interface de Consulta Semântica (*Semantic Query Interface*)

Este componente representa a interface gráfica entre o usuário e o Projeto Números. A interface de consulta semântica é uma interface gráfica rica que permite ao usuário a construção das consultas no Formato Comum.

A SQI é responsável pela coleta de informações e detalhes que agregam um maior valor semântico sobre a consulta, como por exemplo, o autor e o identificador da consulta, e quais Ontologias estão associadas e os tipos de dados que cada coluna armazena.

A interface de consulta semântica representa visualmente as informações sobre a qualidade dos dados, através de marcações visuais na consulta. Uma célula com mais de um valor deverá ser destacada de alguma forma, para que o usuário seja capaz de identificar rapidamente que existe mais de um valor armazenado.

Outra tarefa exercida por esta interface é a representação de filtros a serem aplicados nos resultados. Os filtros de resultado são baseados nas informações de qualidade de dados existentes no servidor, ou seja, os qualificadores de dados serão utilizados durante a construção da consulta para que a busca por resultados esteja limitada apenas aos valores que atendam aos qualificadores.

Esta interface ainda exercerá o papel de envio e recuperação de consultas e ontologias do servidor. As ontologias poderão ser editadas dentro deste componente, fazendo uso de um editor de ontologias associado ao sistema.

Exemplo:

A Interface de Consulta Semântica converte consultas visuais como exibida na Tabela 3 em consultas no Formato Comum.

Tabela 3 - Exemplo de consulta visual no Formato Comum

Loja	**Funcionários**	**Rendimento**
Filial de Copacabana	15	
Filial de Ipanema		

A Figura 10 exibe a representação da tabela anterior para um documento no Formato Comum.

```
...
<pn:consulta>
    <pn:cabecalho>
      <pn:coluna><pn:nome>Loja</pn:nome></pn:coluna>
      <pn:coluna><pn:nome>Funcionários</pn:nome></pn:coluna>
      <pn:coluna><pn:nome>Rendimento</pn:nome></pn:coluna>
    </pn:cabecalho>
    <pn:linha>
        <pn:celula><pn:valor><pn:texto>Filial de Copacabana</pn:texto></pn:
valor></pn:celula>
        <pn:celula><pn:valor><pn:texto/></pn:valor></pn:celula>
        <pn:celula><pn:valor><pn:texto/></pn:valor></pn:celula>
    </pn:linha>
    <pn:linha>
        <pn:celula><pn:valor><pn:texto>Filial de Ipanema</pn:texto></pn:valor></
pn:celula>
        <pn:celula><pn:valor/></pn:celula>
        <pn:celula><pn:valor/></pn:celula>
    </pn:linha>
</pn:consulta>
...
```

Figura 10 - Consulta convertida no Formato Comum

Responsabilidade: Uma interface gráfica para a construção das consultas no Formato Comum. O usuário deve ser capaz de enviar Ontologias para o servidor, usar estas Ontologias e utilizar recursos gráficos para construção da consulta.

Saída: Documentos no Formato Comum produzido pelo usuário, enriquecido com palavras-chave.

Situação: Implementado.

4.3.3 –Camada de Dados e Conhecimento (*Data and Knowledge Layer*)

O Servidor do Projeto Números é composto pela Camada de Dados e Conhecimento e pela Camada de Coleta de Dados. A Camada de Dados e Conhecimento é responsável pela representação, organização e processamento dos dados nos repositórios locais.

A Camada de Dados e Conhecimento é a camada que aguarda as consultas e requisições da Camada de Interface, respondendo às solicitações de processamento, envio e requisição de dados. Para simplificação e extensibilidade esta camada é representada pelos seguintes componentes:

- Mecanismo de Extração de Informação (*Information Extraction Engine*) – responsável pela extração de informação de documentos;
- Gerente de Esquemas de Repositório (*Repository Schema Manager*) – responsável pela organização e manutenção dos esquemas de

representação de dados locais, como tabelas dos bancos de dados relacionais e esquemas de representação de dados multidimensionais;

- Mecanismo de Conversão de Formatos (*Format Conversion Engine*) – responsável pela conversão de documentos externos (planilhas, documentos de texto) para o formato esperado pelo Mecanismo de Extração de Informação;
- Mecanismo de Gerência de Consultas (*Query Management Engine*) – responsável pela gerência e transformação das consultas, seu papel é enviar a consulta no Formato Comum para a Camada de Coleta de Dados;
- Mecanismo de Alimentação de Dados (*Data Feed Engine*) – responsável pela alimentação dos repositórios de dados locais.

Esta camada apresenta os seguintes repositórios de dados:

- O repositório de dados, um banco de dados no sistema de gerenciamento de banco de dados relacional (SGBD) – responsável por armazenar os resultados das consultas;
- *Data Warehouse* (*Data Warehouse*) – responsável pela visão multidimensional dos dados;
- Repositório de Conhecimentos (*Knowledge Base*) – responsável pelo armazenamento das consultas no Formato Comum e das Ontologias;
- Repositório de Informações (*Information Repository*) – responsável pelo armazenamento dos documentos processados pelo Mecanismo de Conversão de Formatos.

Responsabilidade: Receber consultas no Formato Comum, pré-processar a consulta e repassá-la para a camada de coleta de dados. Este componente é responsável também por armazenar os dados encontrados e responder às solicitações do usuário.

Entrada: Consultas no Formato Comum.

Saída: Documentos no Formato Comum para a camada de Coleta de Dados e; Documentos no Formato Comum com dados de resposta para o usuário.

Situação: Parcialmente implementada.

4.3.3.1 – Mecanismo de Extração de Informações (*Information Extraction Engine*)

O Mecanismo de Extração de Informações é responsável por recuperar os documentos pré-processados do Repositório de Informações, extrair os dados de seu conteúdo e popular as tabelas do SGBD relativas a uma (ou mais) consultas feitas pelo usuário.

Este componente utiliza módulos para processar diferentes formas de representação dos dados. A utilização de componentes possibilita a fácil extensibilidade da capacidade de processamento de novo tipos de representação de dados.

Este componente possui o módulo Criador de Consultas (*Query Creator*), responsável pela análise dos dados e criação de novas consultas a partir do resultado encontrado. Esta funcionalidade é essencial para o crescimento do repositório de Conhecimentos, formando assim um ciclo de um sistema de Gestão de Conhecimento, onde os dados obtidos de passos anteriores são utilizados para inferir novos dados. Assim fazem parte deste Mecanismo de Extração de Informações os seguintes módulos:

- Mecanismo de Qualidade de Dados (*Data Quality Engine*)
- Criador de Consultas (*Query Creator*)
- Extrator de Informações de Tabelas (*Table Extractor Engine*)
- Extrator de Informações de Textos (*Text Extractor Engine*)

Responsabilidade: Extração de dados dos documentos encontrados.

Entrada: Documentos pré-processados, recebidos como objetos `DocumentFragment`.

Saída: Dados encontrados nos documentos pré-processados.

Situação: Proposto.

4.3.3.1.1 – Extrator de Informações de Tabela (*Table Information Extractor*)

Um componente capaz de ler informações representadas por uma tabela nos documentos armazenados no Repositório de Informações. Para extrair informações de tabelas é necessário levar em consideração a disposição dos dados e os metadados que circundam esta tabela, como informações sobre origem, autoria, etc.

Exemplo:

Este componente extrai informações de tabelas de dados, utilizando, por exemplo, o título das colunas. A Tabela 4 exibe uma tabela de exemplo, onde este

componente extrairia, por exemplo, que a Filial de Copacabana possui 15 funcionários e uma renda de R$ 15 mil.

Tabela 4 - Tabela com dados a serem extraídos pelo Extrator de Informações de Tabelas

Loja	Funcionários	Rendimento
Filial de Copacabana	15	R$ 15.000
Filial de Ipanema	10	R$ 12.000

Responsabilidade: Extrair dados armazenados em tabelas pré-processadas.

Entrada: Objeto do tipo `DocumentFragment` com tabelas pré-processadas.

Saída: Dados encontrados nas tabelas pré-processadas.

Situação: Proposto.

4.3.3.1.2 – Extrator de Informações de Textos (*Text Information Extractor*)

Este componente é responsável pela extração de dados e metadados de documentos de texto armazenados no Repositório de Informações. Para a extração destes dados (e metadados) é necessário levar em consideração a formatação do texto, as suas características, como idioma, palavras sinônimas e cabeçalho dos documentos que podem revelar informações essenciais para o processamento da informação.

Exemplo: O Extrator de Informações de Texto utiliza informações situadas ao redor do dado analisado. Por exemplo, o texto "funcionários: 15; rendimento: R$ 15 mil", este componente utilizará o marcador ";" para identificar a separação dos dados e rótulos.

Responsabilidade: Extrair dados armazenados em parágrafos e textos.

Entrada: Objeto do tipo `DocumentFragment` com texto pré-processado, como parágrafos HTML ou trechos de documentos em formato proprietário.

Saída: Dados encontrados nos textos pré-processados

Situação: Proposto.

4.3.3.1.3 – Mecanismo de Qualidade de Dados (*Data Quality Engine*)

O Mecanismo de Qualidade de Dados é o componente responsável por gerenciar uma lista de qualificadores que podem ser aplicados aos dados e responder solicitações sobre os atributos de uma determinada informação considerando os seus qualificadores.

Responsabilidade: Gerenciar uma lista de qualificadores de dados e aplicá-los nos dados recebidos, a fim de determinar a qualidade dos dados. Outra atividade é prover um servidor de classificação de dados usado pela GUI.

Entrada: Documento de resposta ao usuário no Formato Comum.

Saída: Documentos de resposta ao usuário no Formato Comum com informações sobre qualidade dos dados encontrados.

Situação: Proposto.

4.3.3.1.4 – Criador de Consultas (*Query Creator*)

O componente Criador de Consultas é invocado quando: o Mecanismo de Qualidade de Dados não possui dados suficientes para classificar uma determinada informação; quando o Mecanismo de Extração de Informações não extrai dados suficientes para processar uma consulta no Formato Comum e popular o repositório SGBD; ou ainda quando o resultado da consulta no Formato Comum possui poucos dados, neste caso a consulta é reenviada para a Camada de Coleta de Dados com mais metadados e dados de exemplo.

Responsabilidade: Criar consultas quando há uma necessidade por mais dados.

Entrada: Dados de exemplo e um trecho de documento no Formato Comum.

Saída: Uma nova consulta no Formato Comum enviada para o Mecanismo Gerenciador de Consultas.

Situação: Proposto.

4.3.3.2 – Mecanismo de Conversão de Formato (*Format Conversion Engine*)

O componente Mecanismo de Extração de Informação é responsável pela extração dos dados de documentos. Estes documentos podem estar representados em formatos diversos como planilhas eletrônicas, documentos de textos, páginas da Internet, etc. e foram escolhidos pela Camada de Coleta de Dados como documentos que contém dados úteis para a construção da resposta a uma consulta no Formato Comum.

O papel deste componente é então recuperar as informações contidas nos mais diversos formatos, representá-las num único formato e armazená-las no Repositório de Informação, onde esses documentos pré-processados aguardarão o seu processamento final.

Responsabilidade: Pré-processar documentos encontrados na Internet e repositórios de documentos e armazenar como objetos `DocumentFragment`.

Entrada: Documentos com dados para processamento em formato original.

Saída: Objetos `DocumentFragment`.

Situação: Proposto.

4.3.3.3 – Gerente de Esquemas de Repositório (*Repository Schema Manager*)

O Projeto Números utiliza diversos tipos de representação de dados buscando aproveitar algumas características dos modelos de dados utilizados e assim oferecer funcionalidades aos usuários do sistema. O componente Gerente de Esquemas de Repositório é responsável pela manutenção dos esquemas do repositório SGBD e do repositório de multidimensional.

O papel deste componente é coordenar a distribuição de tarefas entre os seguintes módulos após o recebimento de uma Ontologia ou de um consulta no Formato Comum, pré-processada pelo Mecanismo de Gerência de Consultas:

- Gerente de Esquema de *Data Warehouse* – responsável pela transformação de Ontologias em Dimensões no modelo multidimensional. Também gera uma tabela Fato no repositório multidimensional para cada consulta no Formato Comum;
- Gerente de Esquema Relacional – responsável pela transformação das consultas no Formato Comum em tabelas no repositório SGBD;
- Mecanismo de Demanda de Dados – verifica se os dados existentes nos repositórios são suficientes para a construção da resposta;
- Mecanismo de Solução de Consultas – responsável pela construção da resposta de uma consulta no Formato Comum.

Responsabilidade: Manter consistente as estruturas de dados dos repositórios.

Entrada: Documentos no Formato Comum e Ontologias.

Saída: Atualização ou criação de estruturas nos repositórios de dados relacionais e *Data Warehouse.*

Situação: Implementado.

4.3.3.3.1 – Gerente de Esquema *Data Warehouse* (*Data Warehouse Schema Manager*)

O componente Gerente de Esquema do *Data Warehouse* é responsável por garantir a atualização e integridade do modelo de dados multidimensional. Para atingir este objetivo o componente realiza duas transformações, uma sobre as consultas no Formato Comum e outra sobre as Ontologias recebidas.

No Projeto Números o modelo usado é conhecido como Constelação, onde várias tabelas Fatos utilizam as tabelas de Domínio existentes.

A construção das tabelas Fato ocorre com o processamento das consultas no Formato Comum, que ocorre após o seu pré-processamento pelo Mecanismo de Gerência de Consultas. As tabelas do modelo relacional são construídas a partir do processamento das Ontologias enviadas ao sistema, onde cada conceito representado na Ontologia é interpretado como uma representação tabular podendo ter relacionamentos com os outros conceitos (representados através de chaves estrangeiras).

Responsabilidade: Garantir a atualização e integridade do modelo de dados do *Data Warehouse*.

Entrada: Documento no Formato Comum e/ou Ontologia.

Saída: Scripts em SQL para atualização do esquema de dados do *Data Warehouse*.

Situação: Implementado.

4.3.3.3.2 – Gerente de Esquema Relacional (*Relational Schema Manager*)

O Gerente de Esquema Relacional é o componente responsável por garantir a atualização e integridade do esquema de dados relacional representado pelo Repositório de Dados. Este componente realiza dois tipos de processamento, um sobre as Ontologias enviadas e outros sobre as consultas no Formato Comum.

O processamento de Ontologias destina-se a construção de tabelas no modelo relacional onde cada conceito representado na Ontologia é representado como uma tabela e seus relacionamentos interpretados como colunas ou chaves estrangeiras (quando os conceitos são associados a outros conceitos).

Já o processamento de consultas no Formato Comum resulta na construção de tabelas no esquema relacional, onde os dados de exemplo e o resultado são representados em forma tabular, e são essas as tabelas preenchidas pelo Mecanismo de Alimentação de Dados.

Responsabilidade: Garantir a atualização e integridade do modelo de dados do SGBD.

Entrada: Documento no Formato Comum e/ou Ontologia.

Saída: Scripts em SQL para atualização do esquema do modelo relacional.

Situação: Implementado.

4.3.3.3.3 – Mecanismo de Demanda de Dados (*Data Demand Engine*)

O Mecanismo de Demanda de Dados atua em conjunto com o Mecanismo de Solução de Consultas, onde a sua responsabilidade é informar ao Mecanismo de Alimentação de Dados a necessidade de mais dados para a construção de uma resposta. Após processar uma consulta o sistema verifica se o resultado é muito próximo à consulta inicial, isto é, poucos dados foram encontrados para construção da resposta, então o sistema constrói o documento de resposta e imediatamente dispara uma nova consulta no Formato Comum para o mecanismo de gerência de consultas.

Responsabilidade: Solicitar mais dados para a resolução de uma consulta.

Entrada: Documento no Formato Comum construído durante o processamento da consulta.

Saída: Após o processamento de uma consulta, a falta de dados gera uma nova consulta no Formato Comum de prioridade inferior, que tentará suprir a necessidade por dados.

Situação: Proposto.

4.3.3.3.4 – Mecanismo de Solução de Consultas (*Query Resolver Engine*)

Este componente é responsável pela construção das respostas, atua junto ao repositório SGBD construindo a resposta a partir dos dados encontrados, retornando à Interface de Consulta Semântica.

Responsabilidade: Montar o documento de resposta ao usuário.

Entrada: Identificador de uma consulta.

Saída: Documento de resposta no Formato Comum

Situação: Implementado.

4.3.3.4 – Mecanismo de Gerência de Consultas (*Query Management Engine*)

O Mecanismo de Gerência de Consultas é o componente responsável pelo pré-processamento e manutenção das consultas no Formato Comum. O pré-processamento é

uma etapa importante no enriquecimento dos repositórios de dados, uma vez que acrescenta informações à consulta original de forma que a busca por dados seja mais ampla, com a finalidade de preparar o repositório de dados para buscas similares no futuro. Este componente é composto dos seguintes módulos:

- Interpretador de Consultas (*Query Interpreter*) – responsável pelo recebimento da consulta e identificação de seus trechos;
- Mecanismo de Expansão de Consultas (*Query Expansion Engine*) – responsável por expandir as consultas, buscando um preenchimento maior do repositório de dados;
- Otimizador de Consultas (*Query Optimizer*) – responsável por aperfeiçoar a consulta inicial, para gerar uma consulta mais eficiente;
- Escalonador de Consultas (*Query Scheduler*) – responsável por enviar a consulta para a fila de processamento da Camada de Coleta de Dados.

Responsabilidade: Pré-processamento e manutenção das consultas no Formato Comum

Entrada: Documento de consulta no Formato Comum.

Saída: Consulta no Formato Comum com uma versão expandida e otimizada. Este documento resultante é enviado para a fila de processamento.

Situação: Implementado.

4.3.3.4.1 – Interpretador de Consultas (*Query Interpreter*)

O Interpretador de Consultas é responsável pelo recebimento das consultas no Formato Comum oriundas da Camada de Interface com o usuário, seu papel é identificar os componentes da consulta e redirecionar para os outros módulos do Mecanismo de Gerência de Consultas, de forma que seu pré-processamento ocorra adequadamente.

Responsabilidade: Identificar os componentes da consulta e coordenar o processamento pelos outros componentes deste pacote.

Entrada: Consulta gerada pelo usuário no Formato Comum.

Saída: Coordenação das atividades neste pacote. Produz um documento no Formato Comum após processamento pelos componentes deste pacote.

Situação: Implementado.

4.3.3.4.2 – Mecanismo de Expansão de Consultas (*Query Expansion Engine*)

As consultas no Formato Comum criadas pelo usuário são por natureza consultas restritivas, isto é, os usuários esperam que o sistema responda com dados que atendam aos critérios pré-determinados durante a criação da consulta.

O sistema constrói a resposta para a consulta do usuário, mas também deve ser capaz de proporcionar o crescimento do seu repositório de dados local de forma que este esteja preparado para consultas similares. Assim, estes critérios limitadores devem ser aplicados para a construção do resultado do usuário, mas isso não implica que a Camada de Coleta de Dados deve desprezar os dados que estejam fora da faixa esperada pelo usuário. Exemplificando, o usuário pode restringir os resultados de uma determinada consulta deverão trazer dados apenas do ano 2000, mas o sistema encontra dados de vários outros anos, este, deverá utilizar estes dados extras para popular os repositórios locais, de forma que estes fiquem preparados para futuras buscas que envolva outras faixas de valores (anos, no exemplo).

O Mecanismo de Expansão de Consultas tem o papel de ampliar o escopo da consulta inicial de forma que a Camada de Extração e Coleta de dados faça uma pesquisa mais ampla. O sistema deve ser capaz de ser passível ao acoplamento de diversos mecanismos de expansão da consulta, e oferece como exemplo a expansão de valores através das folhas da Ontologia.

Todo documento no Formato Comum é composto por três tabelas que armazenam respectivamente a consulta original criada pelo usuário (ou por outros componentes), a consulta expandida e o resultado da consulta.

Responsabilidade: Expandir a consulta original criada pelo usuário.

Entrada: Consulta escrita pelo usuário no Formato Comum.

Saída: A tabela de consulta expandida no documento Formato Comum.

Situação: Implementado.

4.3.3.4.3 – Otimizador de Consultas (*Query Optimizer*)

O componente Otimizador de Consultas é responsável pela reescrita da consulta inicial, de forma que o resultado seja uma segunda consulta mais eficiente, quanto a utilização de recursos computacionais.

Responsabilidade: Otimizar a consulta original, criando uma consulta com um custo computacional menor.

Entrada: Consulta escrita pelo usuário no Formato Comum após expansão.

Saída: A consulta expandida e otimizada.

Situação: Implementado.

4.3.3.4.4 – Escalonador de Consultas (*Query Scheduler*)

Este componente é responsável por programar o repositório Fila de Consultas definindo a ordem de processamento da consultas pela Camada Extratora de Dados. Os documentos no Formato Comum possuem metadados que identificam propriedades do criador e da consulta, com estas informações é possível avaliar qual documento deve ser priorizado no processamento.

O Escalonador de Consultas deve ser capaz de avaliar elementos em comum num conjunto de consultas e definir-lhes uma prioridade de execução, ou ainda identificar que determinados usuários tem prioridade maior na execução de consultas e priorizá-los.

Responsabilidade: Define a ordem de processamento das consultas.

Entrada: Documento no Formato Comum com a consulta escrita pelo usuário.

Saída: O documento de consulta é enviado para a fila de processamento com um valor de prioridade de processamento.

Situação: Implementado.

4.3.3.5 – Mecanismo Alimentador de Dados (*Data Feeder Engine*)

Após a extração dos dados realizada pelo Mecanismo de Extração de Informação, estes são enviados para o Mecanismo Alimentador de Dados que tem o papel de armazená-los no repositório SGBD. Este mecanismo é então o responsável pela finalização do processamento da consulta semântica, que será marcada como respondida, pronta para ser recuperada pela Camada de Interface.

A consulta finalizada ainda poderá ser utilizada como ponto de partida para novas consultas e será utilizada como fonte de dados para consultas similares ou que faça referencia à mesma Ontologia e/ou conjunto de dados.

Responsabilidade: Armazenar os dados encontrados nos repositórios de dados.

Entrada: Dados encontrados nos documentos recuperados da Web.

Saída: Criação e execução de comandos SQL nos repositórios de dados.

Situação: Proposto.

4.3.3.6 – Sistema de Gerenciamento de Banco de Dados – SGBD

A Figura 8, que ilustra a arquitetura do Projeto Números, apresenta vários repositórios de dados. A utilização de vários repositórios permite ao projeto a utilização de diversas formas de representação dos dados, sobretudo aproveitar os benefícios de cada representação.

Os dados de respostas para as consultas são armazenados no Repositório de Dados. A representação dos dados num mecanismo Relacional é uma escolha baseada na disponibilidade de produtos estáveis, de código aberto e principalmente pela representação dos dados de forma tabular.

4.3.3.7 – *Data Warehouse* (*Data Warehouse*)

Para representar os dados de forma multidimensional, este repositório não precisa armazenar os dados, pode-se fazer o uso do repositório SGBD para tal finalidade, mas deve ser capaz de armazenar um esquema de representação de dados que permita ao Projeto Números a construção de consultas multidimensionais.

Este repositório deve ser manipulado pelo Gerente de Esquemas de Repositório para refletir as evoluções do repositório SGBD e para representar as Ontologias e consultas no Formato Comum.

4.3.3.8 – Repositório de Conhecimento (*Knowledge Base*)

O Repositório de Conhecimentos é responsável pelo armazenamento das consultas no Formato Comum e das Ontologias utilizadas pelo sistema. Seus dados podem ser armazenados, por exemplo, em um SGBD convencional.

Responsabilidade: Armazenar Ontologias ou documentos no Formato Comum.

Entrada: Documentos no Formato Comum, Ontologias, ou seus respectivos identificadores.

Saída: Armazena o objeto enviado ou recupera o objeto do SGBD quando recebe um identificador de um componente.

Situação: Implementado pelo SGDB.

4.3.3.9 – Repositório de Informações (*Information Repository*)

O Repositório de Informações armazena documentos e fragmentos de documentos que foram encontrados na Internet, ou em outros locais pesquisados pela Camada de Coleta de Dados. Estes dados são pré-processados de forma que torne a sua

manipulação mais simples por outros componentes, como o Mecanismo de Extração de Informações.

O componente responsável pela inserção de dados neste repositório é o Mecanismo de Conversão de Formatos, que recebe documentos em seus formatos originais e os pré-processa.

Além dos dados sobre os documentos, este componente armazena também informações sobre o local onde este fora encontrado, seu autor, data de criação e alteração, e quaisquer outros metadados disponíveis.

Responsabilidade: Armazenar documentos e fragmentos de documentos pré-processados pelo Conversor de Formatos.

Entrada: Documentos ou fragmentos de documentos encontrados na Web ou repositórios de dados.

Saída: O armazenamento ou recuperação destes documentos ou fragmentos.

Situação: Proposto.

4.3.4 – Camada de Coleta de Dados (*Data Gathering Layer*)

A Camada de Coleta de Dados é responsável pela aquisição dos dados em diversos meios, seja através de buscas pela Internet, navegação por sistema de arquivos ou ainda a utilização de um sistema legado.

Esta camada possui componentes específicos para acessar distintos locais detentores dos dados, e para realizar esta tarefa, divide-se em quatro módulos:

- Coordenador de Coleta de Dados (*Data Gathering Coordinator*) – responsável pela coordenação e fluxo dos dados entre os outros módulos;
- Mediadores e Adaptadores (*Mediators and Wrappers*) – componentes para coletar dados de determinados sites;
- Mecanismo de meta busca (*Meta-search Engine*) – componentes para realizar buscas em sites de buscas;
- Mecanismo de Crawling Inteligente (*Intelligent Crawler Engine*) – responsável por navegar em diversos sites da Internet em busca da informação desejada;
- Fila de Consultas (*Query Queue*) – representa uma fila de consultas a serem processadas.

Responsabilidade: Aquisição dos dados em diversos meios, a partir do consumo da fila de processamento.

Entrada: Documentos na Fila de Processamento.

Saída: Documentos ou fragmentos de documentos enviados para o Mecanismo Conversor de Formatos. Alteração do estado da consulta no Repositório de Conhecimento.

Situação: Proposto.

4.3.4.1 – Coordenador de Coleta de Dados (*Data Gathering Coordinator*)

O Coordenador de Coleta de Dados é responsável por recuperar a próxima consulta a ser processada do repositório Fila de Consultas e disparar as buscas nos outros componentes.

Após o repasse das consultas, este componente é responsável por armazenar os resultados no Repositório *Cache* de Documentos e enviá-los para o Mecanismo de Conversão de Formatos.

Responsabilidade: Recuperar a próxima consulta a ser processada da Fila de Processamento e invocar as atividades dos outros componentes desta camada.

Entrada: Documentos no Formato Comum na Fila de Processamento.

Saída: Invocação de métodos e procedimentos de outros componentes deste pacote.

Situação: Proposto.

4.3.4.2 – Mediadores e Adaptadores (*Mediators and Wrappers*)

Os componentes Mediadores e Adaptadores são um conjunto de componentes criados especificamente para facilitar a coleta de informações em sites específicos. Cada um dos componentes é responsável por conhecer características de um determinado site ou repositório e deve ser capaz de recuperar seus dados.

Responsabilidade: Conhecer a estrutura de determinados sites e recuperar dados destes.

Entrada: Consulta no Formato Comum.

Saída: Documentos ou fragmentos de um determinado site, informações sobre localização, autores, datas e outras propriedades dos documentos.

Situação: Proposto.

4.3.4.3 – Mecanismo de Meta-busca (*Meta-search Engine*)

Os sistemas de buscas da Internet podem ser ilustrados como um índice de sites e documentos espalhados pela rede. Reutilizar esses mecanismos de busca aumenta e simplifica o processo de busca por documentos e dados que respondam às consultas no Formato Comum. O papel deste componente é, portanto conhecer esses sistemas de buscas e utilizá-los para buscar dados.

Responsabilidade: Conhecer sistemas de buscas da Web e utilizá-los para procurarem dados a partir da consulta criada pelo usuário.

Entrada: Consulta no Formato Comum.

Saída: Endereços de sites na Internet com informações sobre a consulta realizada e os documentos encontrados.

Situação: Proposto.

4.3.4.4 – Mecanismo de *Crawling* Inteligente (*Intelligent Crawling Engine*)

O Mecanismo de *Crawling* Inteligente agrupa um conjunto de componentes capaz de realizar diferentes estratégias de navegação pela Internet, em busca dos dados desejados. Este componente difere-se dos Mediadores e Adaptadores, pois este componente não precisa conhecer o site onde o conteúdo será extraído. Este mecanismo divide-se em:

- Identificador de URL (*URL Retriever*) – responsável pela identificação de URL;
- Escalonador de URL (*URL Scheduler*) – responsável pela ordem de processamento das URLS encontradas
- Analisador de Documentos (*Document Analyzer*) – processador dos documentos encontrados;
- Estratégias de Crawling (*Crawling Strategies*) – estratégias para extração do conteúdo dos documentos encontrados.

Responsabilidade: Implementar estratégias genéricas de navegação e obtenção de dados.

Entrada: Consulta no Formato Comum; Resultado do processamento do Mecanismo de Meta-busca.

Saída: Documentos ou fragmentos de um determinado site, informações sobre localização, autores, datas e outras propriedades dos documentos.

Situação: Proposto.

4.3.4.4.1 – Identificador de URL (*URL Recover*)

O Identificador de URLS é responsável por identificar um conjunto de URLS que aparentemente possuem dados para solucionar uma consulta do usuário.

Responsabilidade: Identificar URLs potenciais.

Entrada: Consulta no Formato Comum.

Saída: URL possivelmente com dados a serem extraídos.

Situação: Proposto.

4.3.4.4.2 – Escalonador de URL (*URL Scheduler*)

O Escalonador de URLS é responsável por receber os endereços oriundos do Mecanismo de *Crawling* Inteligente, que possivelmente possuem dados sobre a consulta e classificá-los para um processamento mais eficiente, evitando que o mesmo documento seja processado duas vezes ou um dado documento seja processado antes de suas dependências.

Responsabilidade: Classificar endereços recebidos de forma a realizar um processamento mais eficiente.

Entrada: Conjunto de URLs que possivelmente armazenam dados para processamento.

Saída: O conjunto de URLs recebida disposto em forma a um processamento mais eficiente e eliminação de elementos duplicados.

Situação: Proposto.

4.3.4.4.3 – Analisador de Documentos (*Document Analyzer*)

O componente Analisador de Documentos é responsável pela escolha dos documentos ou fragmentos de documentos que contenham a provável resposta para a consulta. Estes documentos selecionados serão processados futuramente pelo Mecanismo de Conversão de Formatos.

Responsabilidade: Analisar os documentos e verificar se possuem dados para construção da resposta.

Entrada: Conjunto de URLs que apontam para documentos ou fragmentos.

Saída: Documentos ou fragmentos com dados para construção da resposta a uma consulta no Formato Comum.

Situação: Proposto.

4.3.4.4.4 – Estratégias de Crawling (Crawling Strategies)

Os componentes de Estratégias de *Crawling* são responsáveis por procurar na Internet os endereços candidatos a responderem às consultas do usuário, é seu papel selecionar estes documentos e enviar para o componente Identificador de URLS.

Responsabilidade: Definir estratégia de navegação em páginas HTML em busca de endereços que possuam dados.

Entrada: Consulta no Formato Comum.

Saída: Coleção de URLs candidatas.

Situação: Proposto.

4.3.4.5 – *Cache* de Documentos (*Document Cache*)

O repositório *Cache* de Documentos armazena os documentos recuperados, bem como fragmentos de documentos que foram classificados como candidatos para construção da resposta a uma consulta no Formato Comum. Este repositório armazena também os metadados destes documentos como data de recuperação, autor, origem, etc, pois essas informações serão úteis para qualificar os dados encontrados.

Os componentes de Estratégias de *Crawling* podem utilizar este repositório como ponto de partida em suas buscas por sites que contenham o conteúdo procurado.

Responsabilidade: Armazenar os documentos ou fragmentos encontrados pelos componentes desta camada.

Entrada: Documentos encontrados na Web ou repositórios de dados.

Situação: Proposto.

4.3.4.6 – Fila de Consultas (*Query Queue*)

As consultas no Formato Comum são pré-processadas pelo Mecanismo de Gerência de Consultas e enviadas para o repositório denominado Fila de Consultas. O componente Coordenador de Coleta de Dados recupera as consultas deste repositório e envia solicitações aos outros componentes desta camada.

Responsabilidade: Armazenar as consultas no Formato Comum em ordem de processamento.

Entrada: Consultas no Formato Comum com ordem de processamento definido.

Saída: A próxima consulta a ser processada, de acordo com a sua prioridade.

Situação: Proposto.

Capítulo 5 – Implementação

A implementação de uma boa interface é fundamental para o sucesso ou fracasso de uma aplicação. Neste trabalho há uma grande preocupação na construção de uma interface amigável ao ponto de que usuários com pouco conhecimento da arquitetura envolvida possam utilizar o ambiente proposto.

Este capítulo descreve detalhes e decisões adotadas na implementação do Projeto Números focando nas propostas da Interface Gráfica e do Processamento da Consulta, mostra também as principais características das classes envolvidas e seus relacionamentos com outros componentes do projeto. A primeira seção descreve as propostas, exibindo telas e funcionalidades, as seções posteriores abordam um ponto da arquitetura e descrevem os detalhes de programação utilizados para construí-lo.

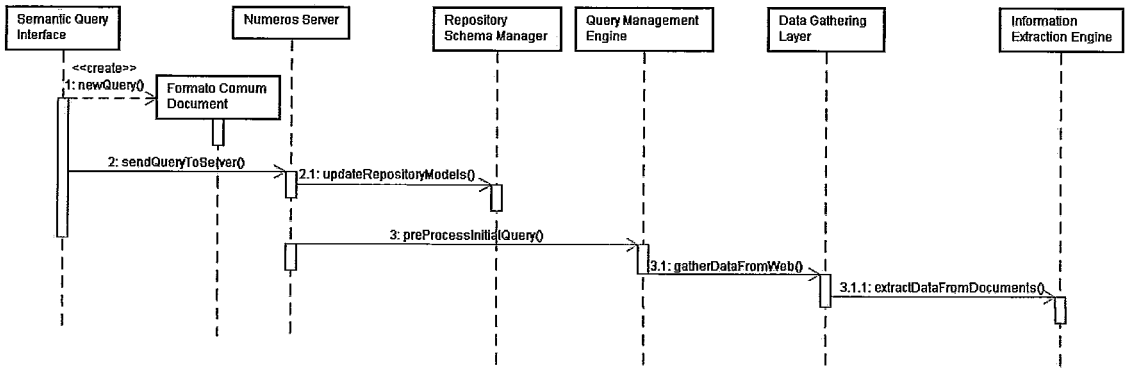


Figura 11 – Diagrama de Seqüência (UML 1.4) demonstrando a interação entre alguns componentes do Projeto Números

A Figura 11 exibe um Diagrama de Seqüência (UML 1.4) demonstrando o processamento de uma consulta no Formato Comum e a interação entre determinados componentes do Projeto Números.

5.1 – Interface Gráfica e Processamento da Consulta

Esta seção exibe telas da interface gráfica, demonstrando as suas funcionalidades e focando na utilização desta ferramenta para a construção de consultas no Formato Comum.

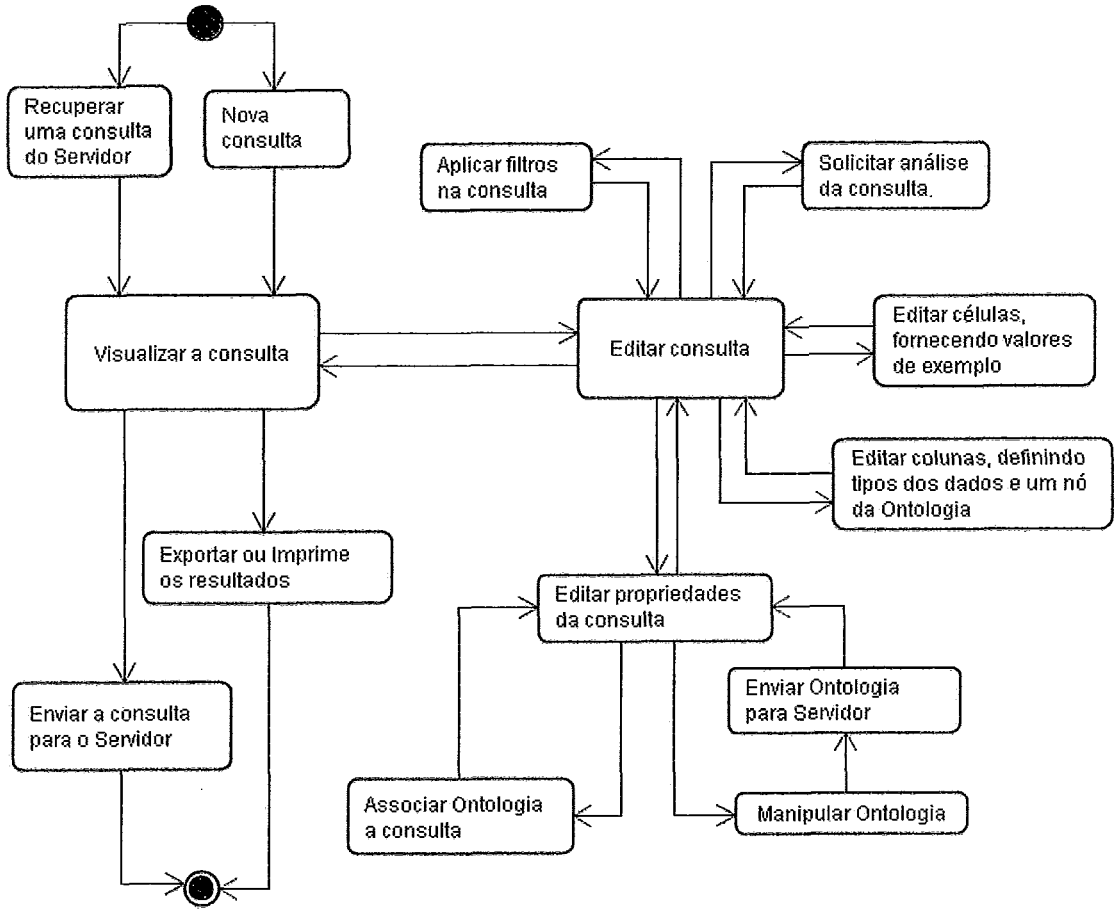


Figura 12 - Diagrama de Atividades (UML 1.4) mostrando o uso típico da interface gráfica

A Figura 12 exibe um Diagrama de Atividades demonstrando a utilização da interface gráfica. A partir deste diagrama as funcionalidades aqui propostas são analisadas através de imagens e uma breve descrição sobre o papel de cada componente.

O diagrama anterior apresenta o fluxo típico de manipulação de um documento no Formato Comum, demonstrando a sua criação e manipulação de um documento já processado.

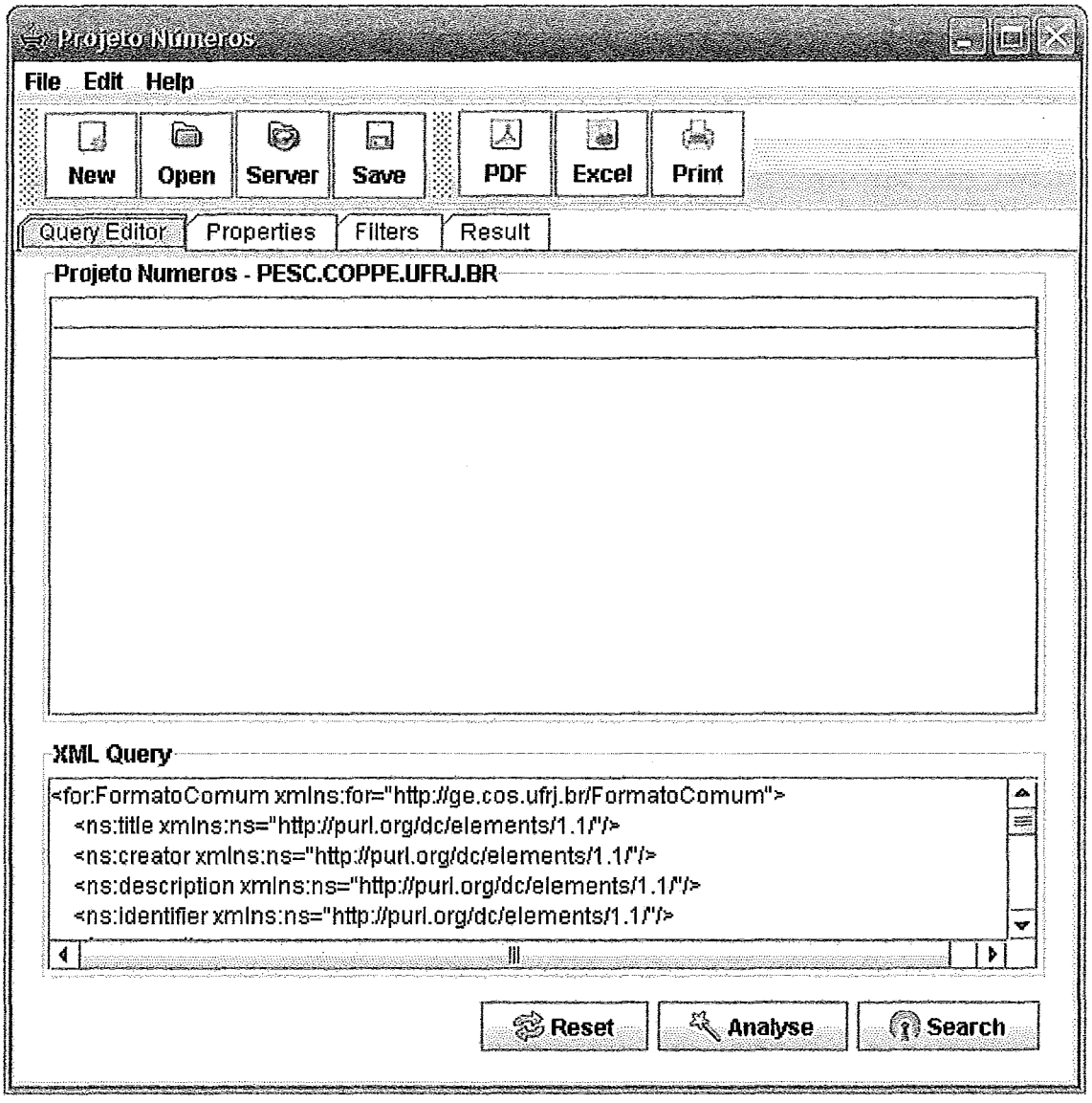


Figura 13 – Tela inicial da interface gráfica do Projeto Números, implementada pela classe `ProjetoNumerosGUI`

A Figura 13 exibe a tela inicial da interface gráfica do Projeto Números, com um consulta no Formato Comum sem dados. É possível observar na figura alguns botões representando ações sobre a consulta em uso pelo sistema, são eles:

- Nova (*New*) – inicia a construção de uma nova consulta;
- Abrir (*Open*) – abre um documento no Formato Comum previamente editado;
- Servidor (*Server*) – recupera uma consulta enviada ao Servidor, com os dados de resposta;
- Salvar (*Save*) – salva a consulta no sistema de arquivos;

- PDF, Excel – exportam o documento atual para documentos respectivamente no formato PDF e Microsoft Excel;
- Imprimir (*Print*) – imprime o documento atual;
- Redefinir (*Reset*) – limpa os dados de exemplo fornecidos para a consulta atual;
- Analisar (*Analyse*) – envia o documento atual para a tela de análise;
- Procurar (*Search*) – envia o documento para processamento.

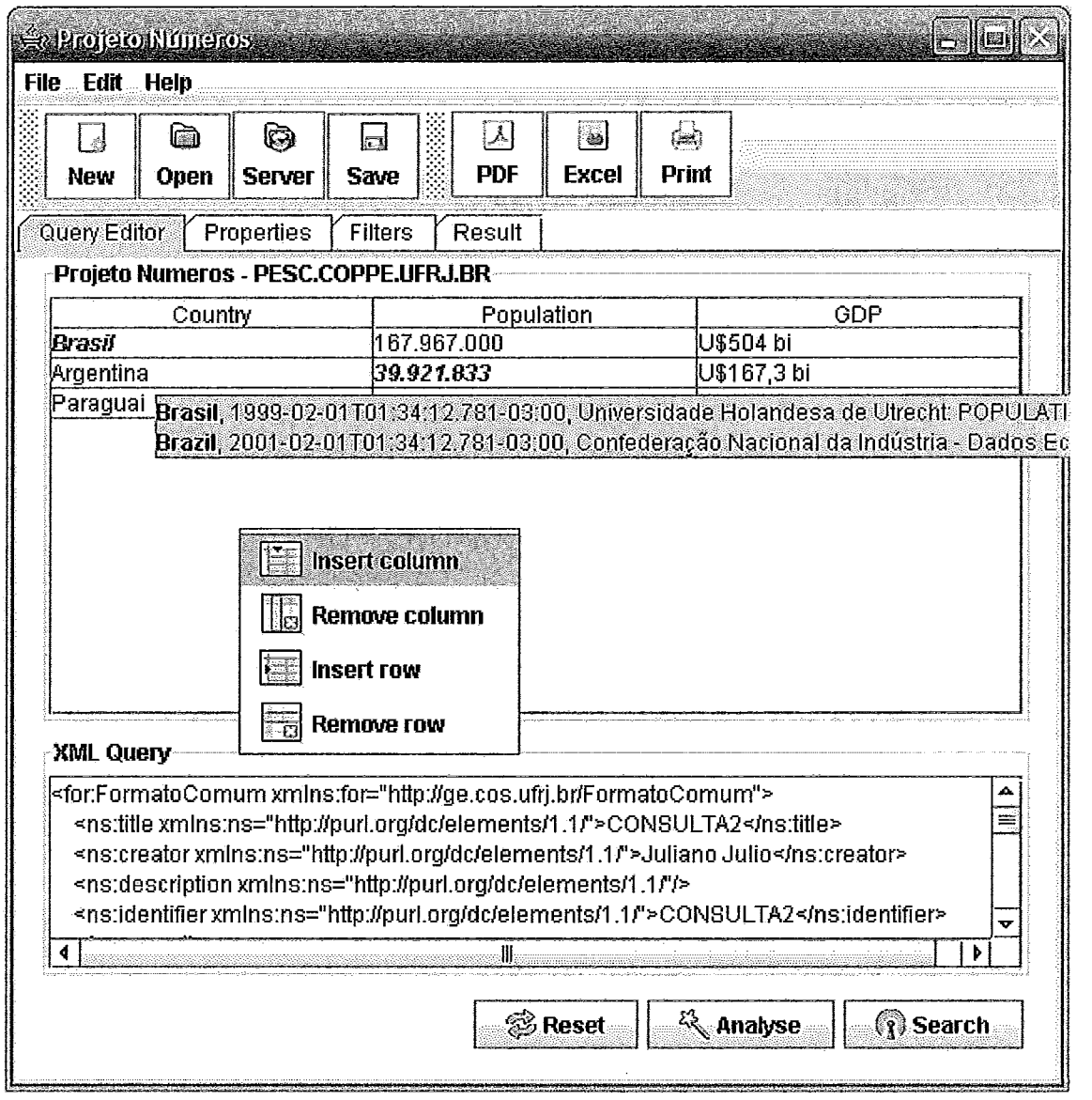


Figura 14 – Tela principal com dados de exemplo e destacando recursos visuais

A Figura 14 mostra uma consulta no Formato Comum criada pela tela principal da interface gráfica. Esta figura mostra alguns recursos visuais da interface, como:

- Mensagens de contexto quando o mouse é posicionado sobre uma célula multivalorada;
- Coloração indicando informações sobre células, onde as células destacadas apresentam mais de um valor;
- Menus para manipulação da consulta, para acrescentar ou remover colunas e linhas.

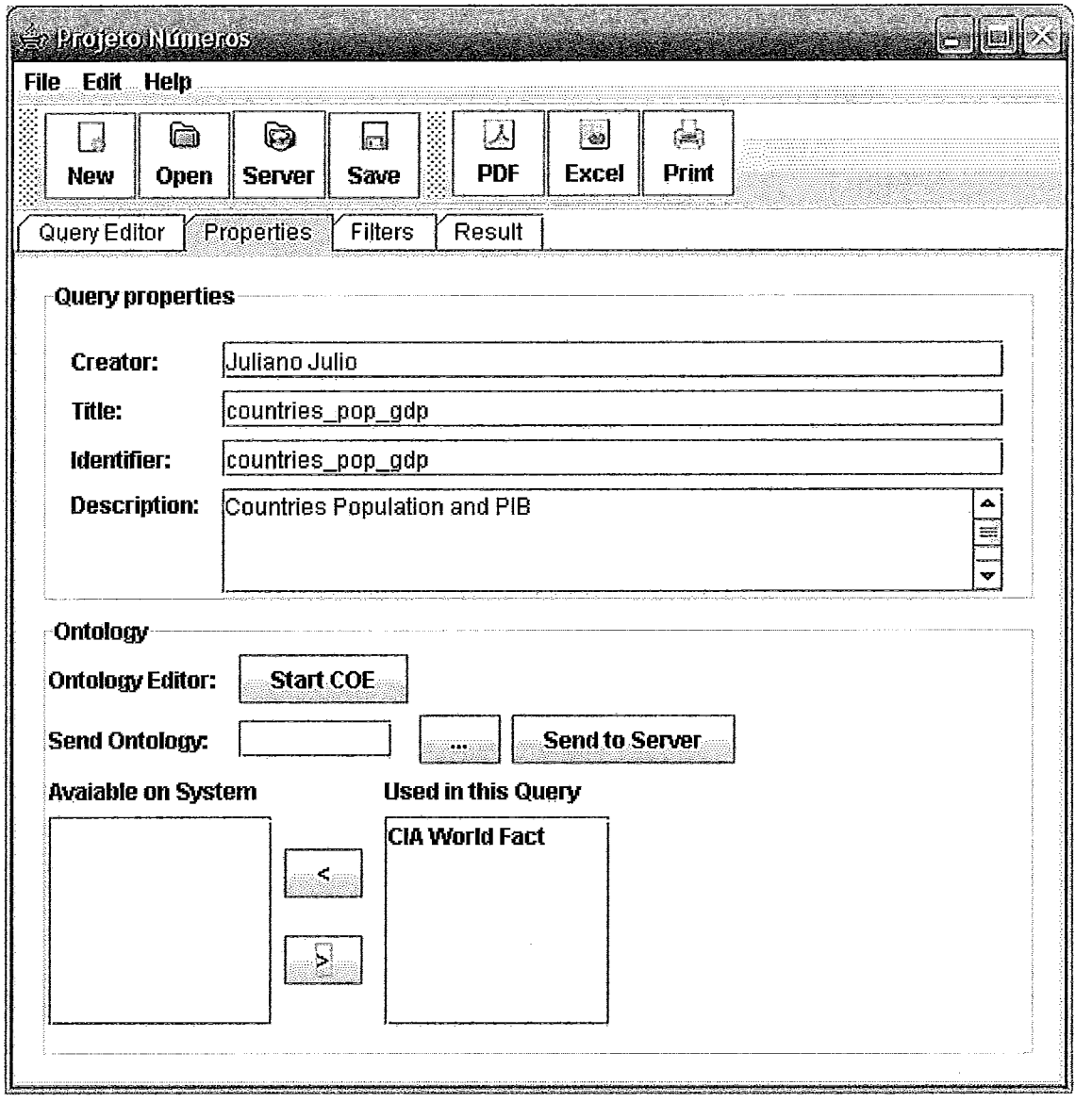


Figura 15 – Editor de Propriedades da Consulta

A Figura 15 mostra o editor de propriedades, onde o usuário pode inserir informações que identificam a consulta, como um identificador, descrição e o nome do autor. Esta tela exibe também a manipulação de Ontologias, que apresenta duas

funcionalidades: a primeira é abrir um editor de Ontologias como o COE e enviar este documento para o Servidor do Projeto Números; e a segunda funcionalidade é a associação de Ontologias ao documento em edição.

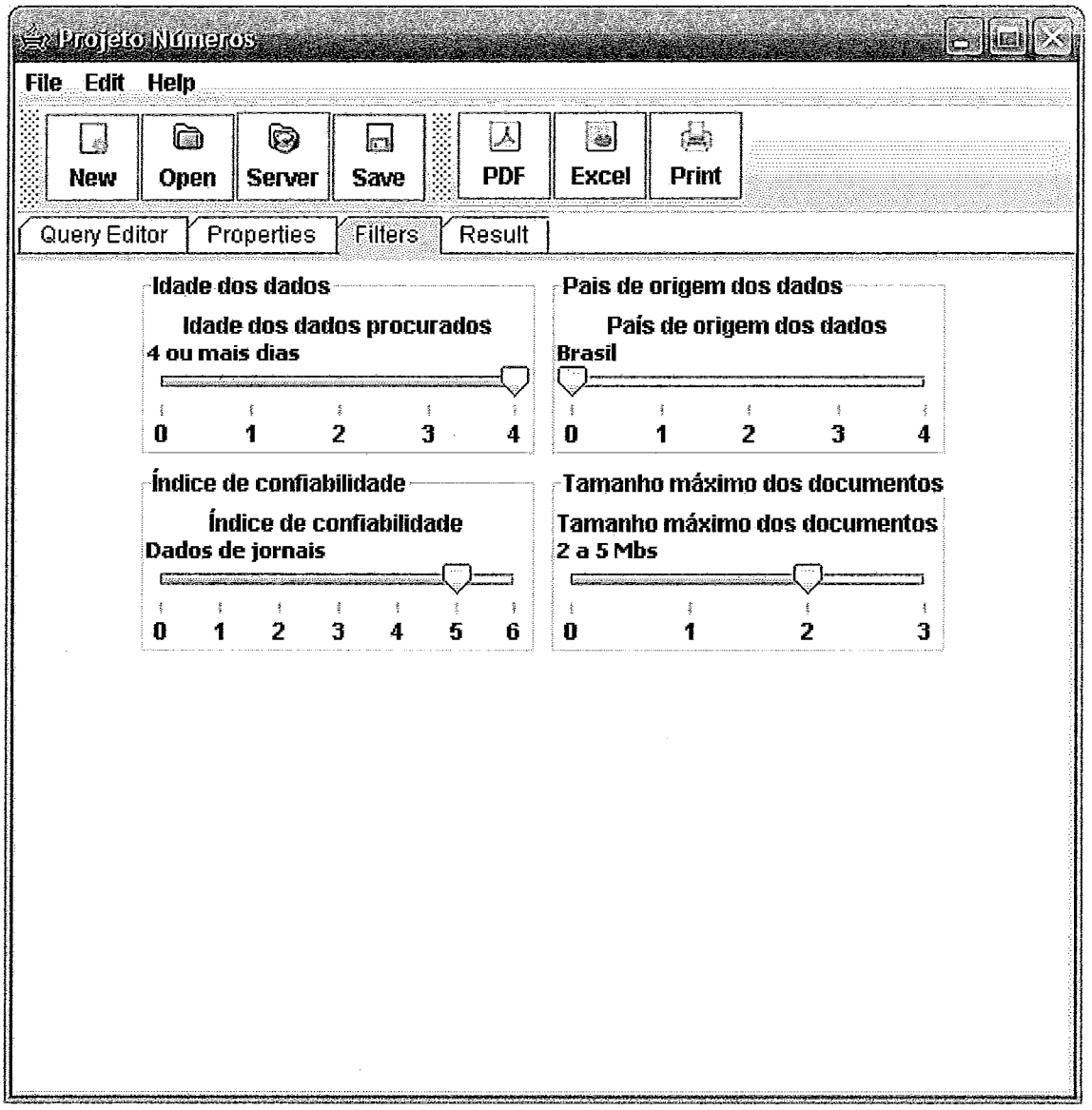


Figura 16 – Tela de filtros de dados para limitar o resultado da busca

O servidor do Projeto Números pode fornecer um conjunto de atributos para a filtragem do resultado. A interface gráfica recupera um conjunto de filtros disponíveis do Servidor e monta a tela exibida na Figura 16, possibilitando ao usuário a filtragem dos dados esperados.

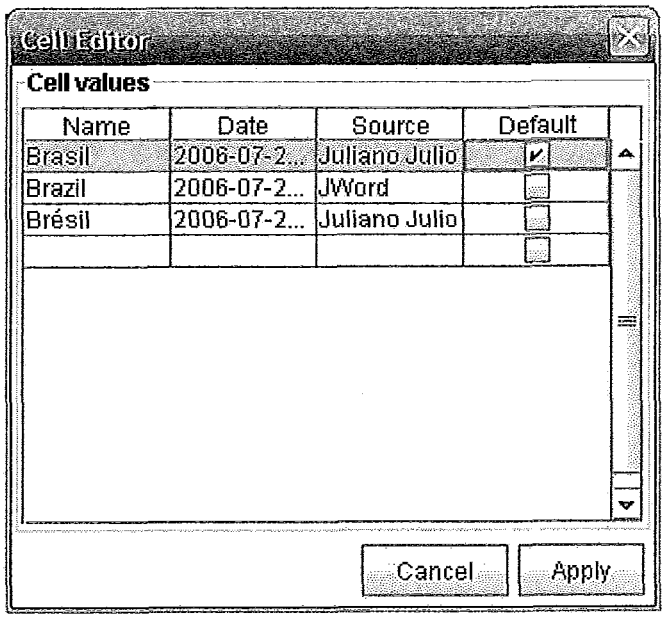


Figura 17 – Editor de células do Formato Comum

O usuário constrói consultas a partir da definição de valores de exemplo, usando o mesmo conceito de consulta por exemplo (*Query by Example*), ele preenche a tabela apresentada na Figura 13 inserindo novas colunas e novas linhas.

Para cada célula que o usuário deseja inserir valores ele deve executar um duplo clique e deve inserir os valores na tela exibida na Figura 17. O usuário pode escolher qual a linha que deverá ser exibida na tela de resultados, para isso ele deve marcar a caixa de seleção da coluna Padrão (*Default*).

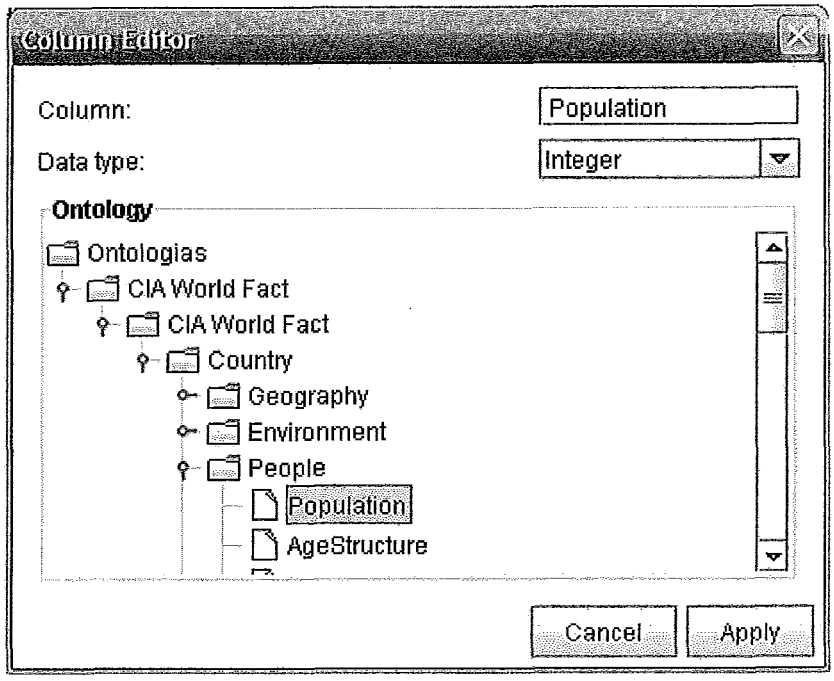


Figura 18 – Editor de Colunas

O usuário deve fornecer informações para cada coluna da sua consulta, ao realizar um duplo clique sobre uma coluna, o sistema exibirá a tela da Figura 18, onde o usuário poderá definir um nome para coluna e o tipo de dados que ela armazenará. Ainda nesta tela o usuário poderá escolher um nó de uma das Ontologias associadas a esta consulta.

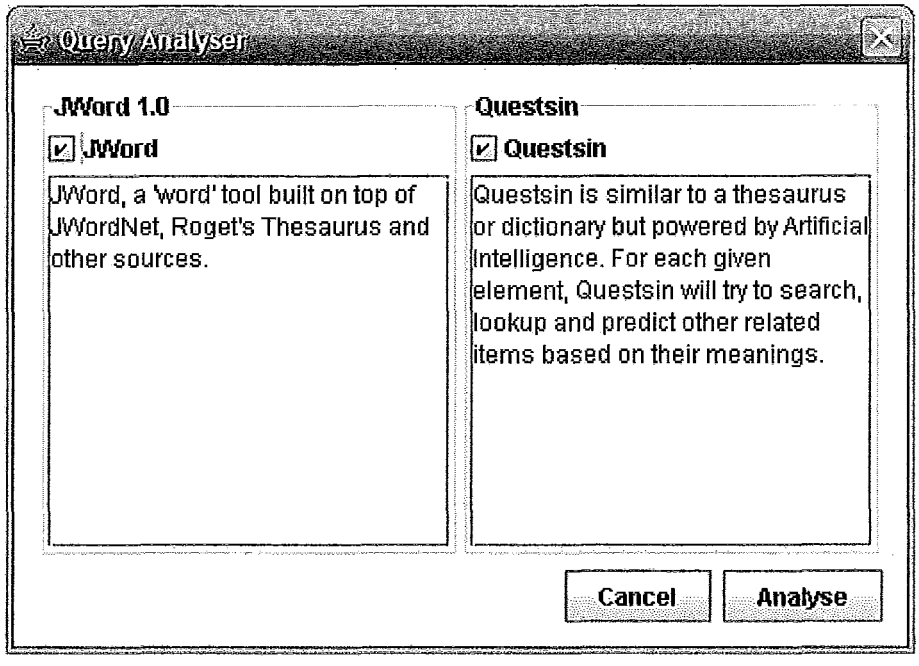


Figura 19 - Analisador de consultas

Após completar as etapas citadas anteriormente, a consulta do usuário está completa, com dados de exemplo e informações sobre os dados a serem procurados. Nesta etapa o usuário pode solicitar a análise de sua consulta, clicando no botão Analisar (*Analyse*) exibido na Figura 13, o sistema exibirá uma janela com os analisadores disponíveis, conforme demonstrado pela Figura 19.

Os analisadores de consulta são utilizados para realizar transformações na consulta quando está em edição, os analisadores implementados acrescentam dados à consulta inicial, expandindo-a usando os dados fornecidos como base. Após a análise da consulta o usuário deverá avaliar quais os dados inseridos acrescentam positivamente para a sua consulta, removendo as entradas indesejadas.

A consulta do usuário está pronta para ser enviada para processamento. O usuário poderá apertar o botão Procurar (*Search*) exibido na Figura 13 e a sua consulta será enviada para processamento.

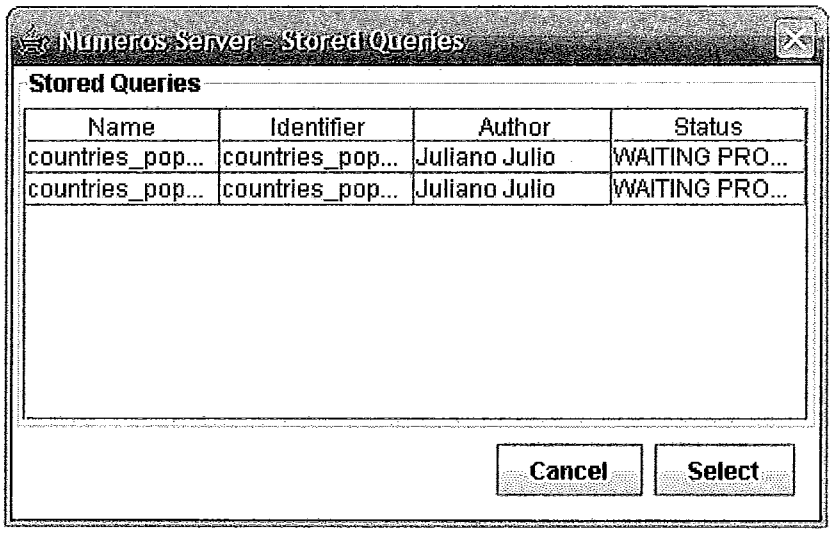


Figura 20 – Tela para recuperação de consulta no servidor

O usuário poderá visualizar as consultas e suas situações clicando no botão Servidor (*Server*) da Figura 13, o sistema exibirá a tela da Figura 20. Caso o usuário deseje visualizar uma das consultas listadas, basta escolher e pressionar o botão Selecionar (*Select*).

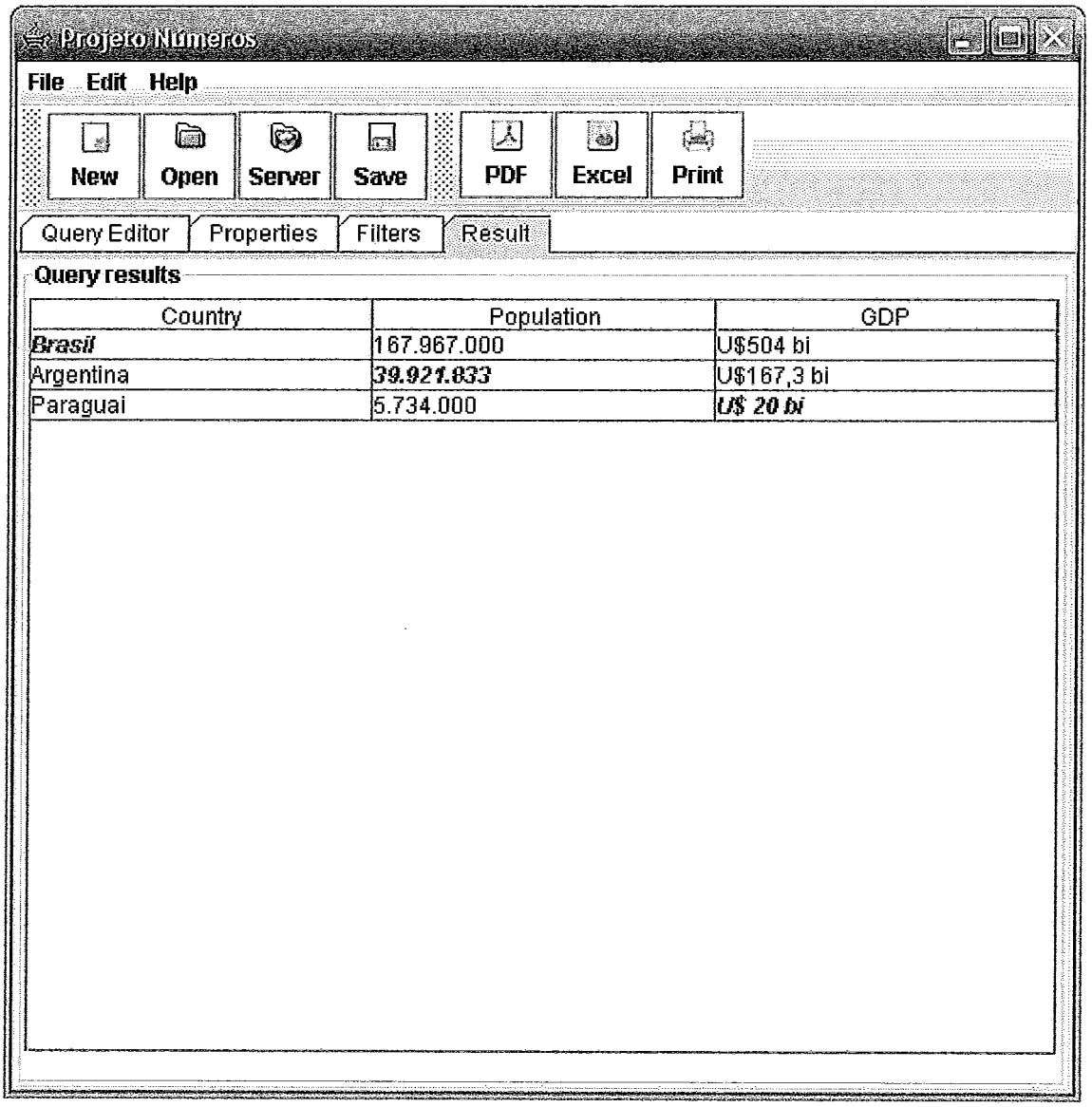


Figura 21 – Tela exibindo o resultado da consulta após processamento

Após recuperar uma consulta o usuário poderá visualizar o resultado conforme a Figura 21.

5.2 – Interface OLAP

As buscas de dados no Projeto Números constroem um repositório de dados similar à consolidação de dados proposta no modelo *Data Warehouse*, e mesmo a consulta no Formato Comum recupera dados de forma analítica ou armazena

informações sobre como consolidá-los. Assim utilizando uma ferramenta OLAP o Projeto Números permite ao usuário manipular esta visão analítica sobre os dados armazenados nos repositórios de dados do Projeto.

Para prover esta visão multidimensional o Projeto Números utiliza um conjunto de ferramentas, composta pelo Mondrian – um servidor OLAP e pelo JPivot – uma ferramenta para executar consultas e visualizar os dados como um cubo multidimensional.

5.3 – Mondrian

O Mondrian é um servidor OLAP escrito em Java que implementa a linguagem de consulta MDX e a especificação JOLAP (Java *Online Analytical Processing* (SUN MICROSYSTEMS, 2006)). Este servidor lê dados de repositórios relacionais (e/ou outras fontes de dados) e agrega os dados em memória (JULIAN HYDE, 2006).

A representação dos dados no Mondrian é feita através de arquivos XML que mapeiam repositórios de dados relacionais provenientes de estruturas multidimensionais. Este mapeamento é feito no PN pela classe `DataWarehouseSchemaManager` que recebe Ontologias e/ou documentos no Formato Comum.

Um documento Mondrian Schema representa um ou mais cubos de dados cada um composto por uma tabela com colunas de medidas e relacionamentos com tabelas de dimensões. A Figura 22 mostra um documento Mondrian Schema representando os principais componentes para a construção dos cubos nesta ferramenta.

```
<Schema>
  <Cube name="Sales">
    <Table name="sales_fact_1997"/>
    <Dimension name="Gender" foreignKey="customer_id">
      <Hierarchy hasAll="true" allMemberName="All Genders" primaryKey="customer_id">
        <Table name="customer"/>
        <Level name="Gender" column="gender" uniqueMembers="true"/>
      </Hierarchy>
    </Dimension>
    <Dimension name="Time" foreignKey="time_id">
      <Hierarchy hasAll="false" primaryKey="time_id">
        <Table name="time_by_day"/>
        <Level name="Year" column="the_year" type="Numeric"
            uniqueMembers="true"/>
        <Level name="Quarter" column="quarter"
            uniqueMembers="false"/>
        <Level name="Month" column="month_of_year" type="Numeric"
            uniqueMembers="false"/>
      </Hierarchy>
    </Dimension>
    <Measure name="Unit Sales" column="unit_sales"
        aggregator="sum" formatString="#,###"/>
    <Measure name="Store Sales" column="store_sales"
        aggregator="sum" formatString="#,###.##"/>
    <CalculatedMember name="Profit" dimension="Measures"
        formula="[Measures].[Store Sales]-[Measures].[Store Cost]">
      <CalculatedMemberProperty name="FORMAT_STRING" value="$#,##0.00"/>
    </CalculatedMember>
  </Cube>
</Schema>
```

Figura 22 – Documento Mondrian Schema exibindo os componentes para construção de cubos

Um documento Mondrian Schema descreve como o Mondrian deve recuperar os dados do repositório relacional e como realizar a agregação. Cada componente deste documento descreve um componente em um sistema multidimensional, ou seja, existem descritores para Cubos, Dimensões e Medidas, como é possível observar na figura anterior.

5.3.1 – Multidimensional Expressions Language – MDX

Multidimensional Expressions Language – MDX é uma linguagem de expressões usada para consultar banco de dados OLAP, criada pela Microsoft em 1997 e passou a ser adotada por diversas aplicações analíticas (PEARSON, 2002).

Uma consulta MDX lembra a estrutura e sintaxe de uma consulta SQL, a Figura 23 exibe uma consulta que busca o valor do PIB (*GDP*) e da população (*population*) no cubo gerado pela consulta exibida na Figura 14. Vale ressaltar que enquanto uma consulta SQL gera uma resposta tabular, isto é, outra tabela, as consultas MDX resultam em outro cubo OLAP.

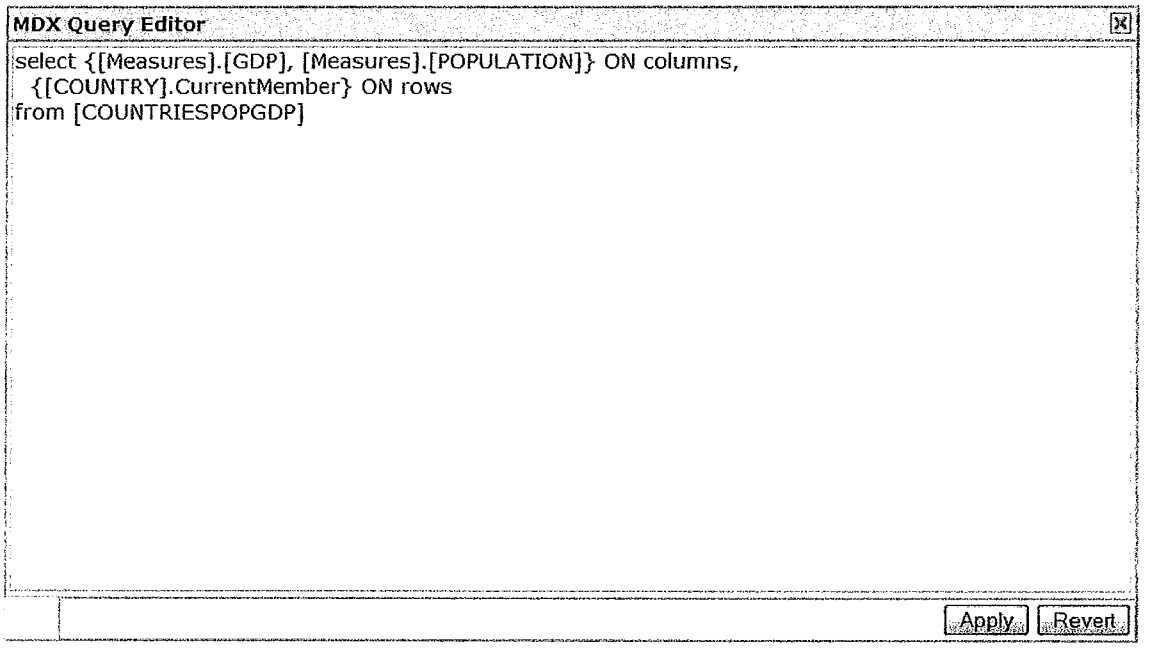
Os componentes de uma consulta típica MDX são os cubos de origem dos dados na cláusula FROM e os eixos nas cláusulas SELECT.

5.4 – JPivot

Para simplificar a visualização e manipulação dos dados do Projeto Números foi escolhido o JPivot, uma biblioteca JSP (Java Server Pages) para renderização e manipulação de tabelas OLAP (JPIVOT, 2006).

A Figura 23 representa o editor de consultas MDX do JPivot. A consulta exibida pela figura é uma projeção sobre um cubo gerado por uma consulta no Projeto Números.

MDX Query Editor



	Measures	
COUNTRY	GDP	POPULATION
-All COUNTRY.COUNTRYs	1258.0	8558.0
Argentina	3.0	3111.0
Brasil	24.0	2224.0
Brazil	1211.0	1112.0
Paraguai	20.0	2111.0

Figura 23 - Captura de tela da aplicação JPivot com o Mondrian como repositório de dados

O JPivot é atualmente distribuído junto ao Mondrian, eliminando assim a necessidade de configurações extras.

5.5 – Formato Comum

A arquitetura do Projeto Números define a troca de mensagens XML como meio padrão de troca de dados entre os módulos que compõe o sistema. Para manter a padronização e ganhar um maior controle sobre o formato das mensagens, definimos o Formato Comum, um documento *XML Schema* que determina a estrutura das mensagens XML transmitidas entre as camadas do Projeto Números. Esta seção descreve o desafio na criação e manutenção desta linguagem, bem como as ferramentas utilizadas para sua manipulação pelas classes da aplicação.

Ao utilizar um documento *XML Schema* para representar o Formato Comum, o desafio foi fornecer um meio prático e transparente para utilização deste a partir das classes da camada de Modelo (e de outras camadas da arquitetura do Projeto Números). A implementação de classes para manipulação através do JDOM foi descartada, pois o Formato Comum poderia sofrer muitas alterações no decorrer do desenvolvimento do projeto, assim, para solucionar esse descasamento foi utilizado o XMLBeans.

O XMLBeans é uma aplicação Java desenvolvida e distribuída pela Apache, que permite a manipulação de documentos XML de forma transparente a partir da transformação de um documento *XML Schema* em classes Java (APACHE GROUP, 2006). Após a execução do XMLBeans sobre o XML Schema do Formato Comum, um conjunto de classes Java é gerado e compactado no pacote `formato-comum.jar`, que contem métodos para manipulação de cada objeto descrito pelo FC, por exemplo: `getConsulta`, `getAuthor`, `setResultado`.

O pacote `formato-comum.jar` é então incorporado no Projeto Números, provendo um meio transparente de manipulação de documentos XML que respeitem o padrão imposto pelo Formato Comum. A subseção "Classe `QueryHandler`" explica em maiores detalhes a utilização das classes geradas.

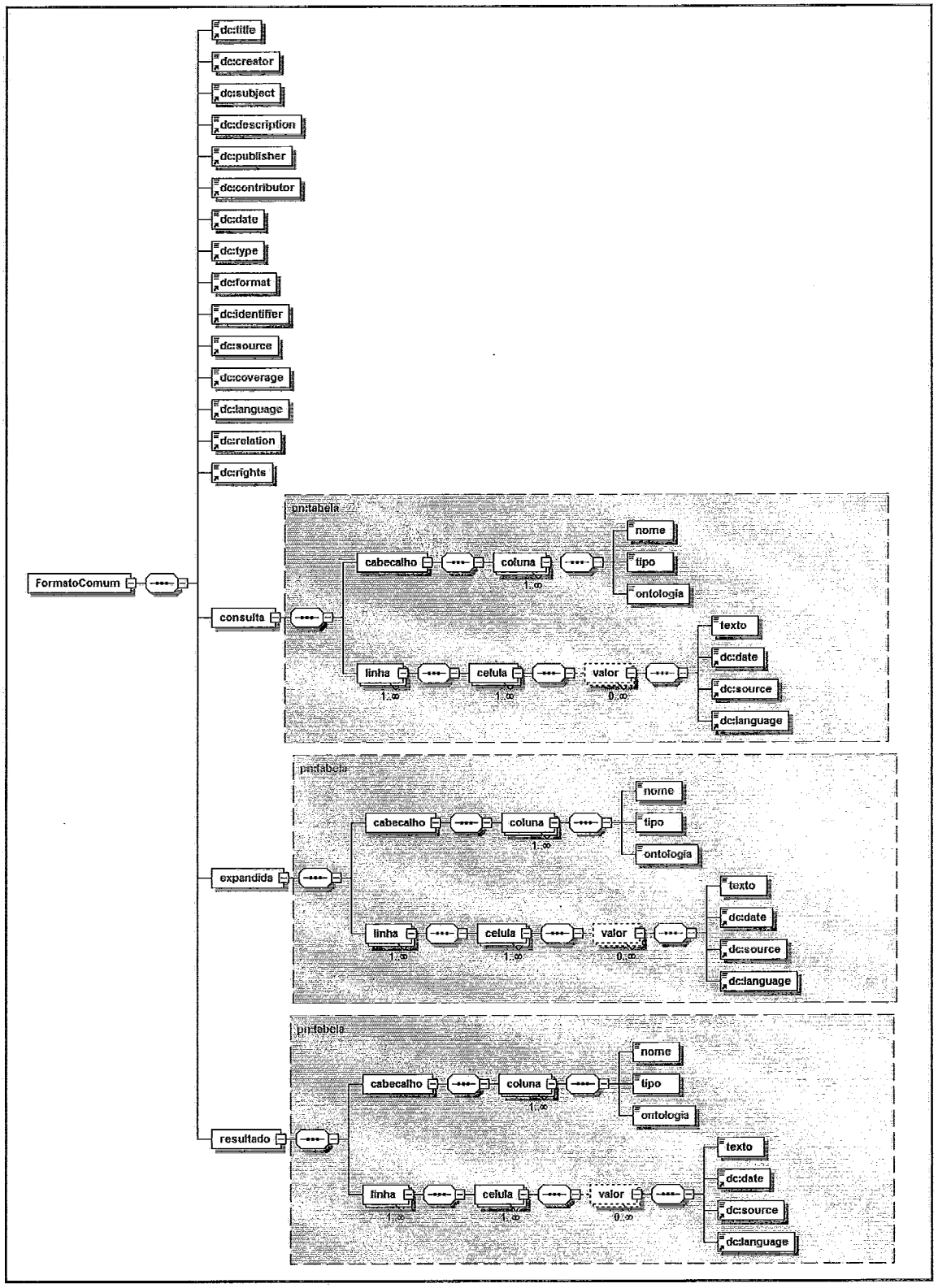


Figura 24 - Diagrama do XML Schema do Formato Comum

A Figura 24 exibe um diagrama com uma árvore de dados do XML Schema, que determina os elementos possíveis nos documentos no Formato Comum. Os elementos Dublin Core são representados pelo prefixo "dc:". O elemento "pn:tabela" representa

uma tabela que armazena a consulta, a consulta expandida ou ainda o resultado para a consulta. O Apêndice B exibe o documento XML *Schema* que define o Formato Comum

5.6 – COE – Editor Colaborativo de Ontologias

A edição de ontologias é uma tarefa bastante explorada no meio acadêmico e está fora do escopo deste trabalho, mas para a implementação de referência proposta neste, escolhemos o COE, uma ferramenta colaborativa para edição e compartilhamento de ontologias. A arquitetura proposta neste trabalho permite a escolha de um editor de ontologias como o COE.

Editor Colaborativo de Ontologias – COE – é uma aplicação que permite a edição cooperativa de ontologias em uma arquitetura Peer2Peer que permite a troca de conhecimento entre os usuários do sistema (XEXÉO, VIVACQUA *et al.*, 2005). A Figura 25 mostra uma imagem do COE com a ontologia *CIA World Facts* em edição.

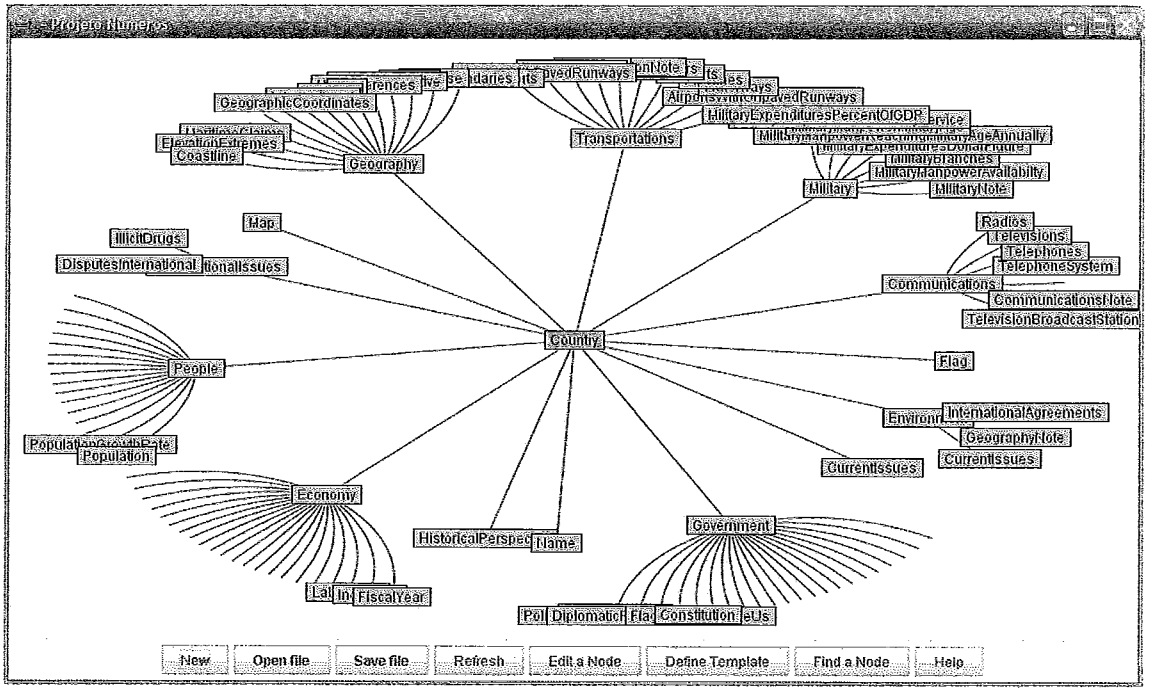


Figura 25 – COE – Editor Colaborativo de Ontologias

5.7 – Camada da Interface Gráfica

A Camada da Interface Gráfica (*Interface Layer*) é representada na arquitetura pela Interface de Consulta Semântica (*Semantic Query Interface*) e pela Interface OLAP

(*OLAP Interface*). Esta seção descreverá a Interface de Consulta Semântica, mostrando as diversas classes que a compõe e como se relacionam. A seção "5.2 – Interface OLAP" descreverá o outro componente desta camada, responsável pela visão OLAP dos dados.

Esta camada foi implementada usando o padrão Modelo, Controle e Visão (MVC), um padrão de Projetos de Engenharia de Software. O MVC é um padrão de projeto que define uma arquitetura em três camadas: Modelo, com as classes do domínio da aplicação, também conhecidas como classes de negócio; Visão, com classes que controlam a interação com o usuário; e Controle, com classes que fazem a ligação entre a camada Modelo e Visão.

O principal objetivo ao utilizar o padrão MVC é manter um nível de independência da camada de visão, que pode ser construída de diferentes maneiras, como uma página de Internet, ou através de um sistema de janelas, como uma aplicação nativa do sistema operacional, dentre outras formas. São requisitos do Projeto Números é a utilização da linguagem Java e a necessidade de fornecer uma Interface Gráfica com ícones, menus e recursos visuais para interação do usuário, assim, este trabalho utiliza o a API Swing, um dos padrões da linguagem Java.

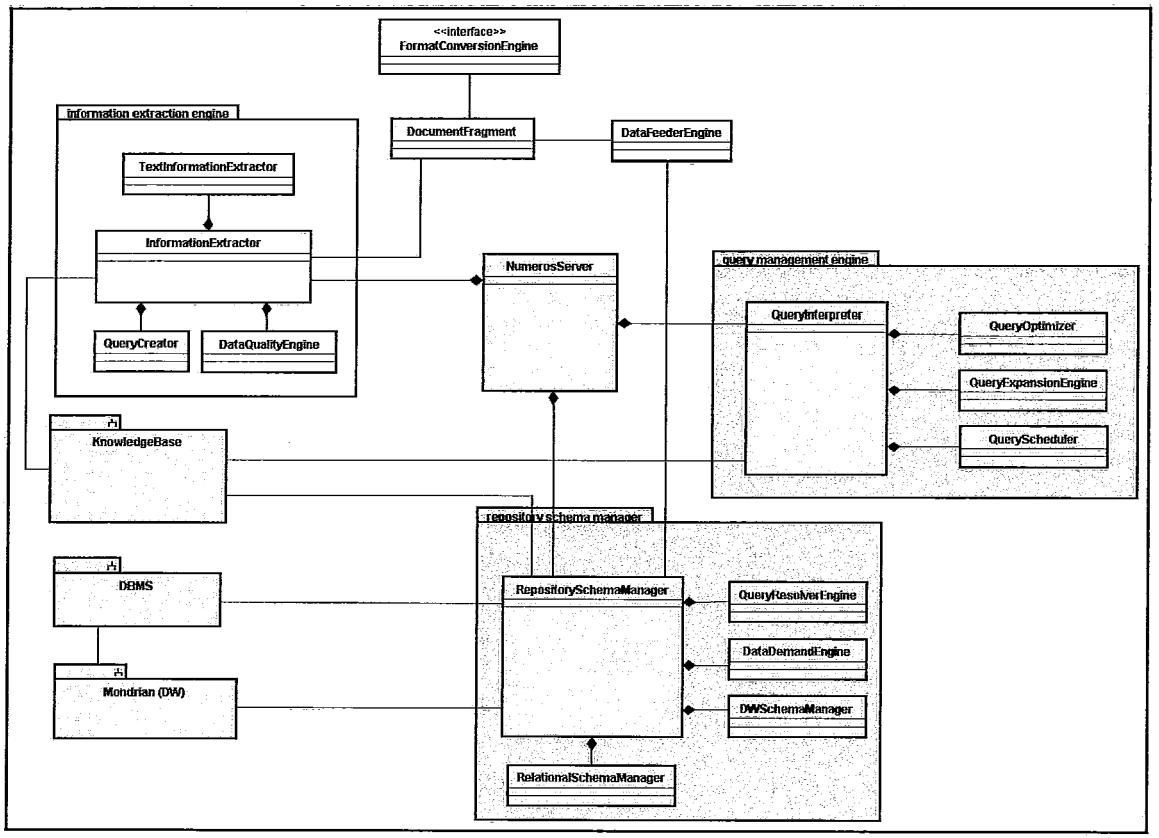


Figura 26 – Diagrama de Classes (UML 1.4) da Camada de Dados e Conhecimento

A Figura 26 apresenta um diagrama UML simplificado da camada de Dados e Conhecimento, destacando os componentes abordados neste trabalho. Este capítulo descreve cada um destes componentes.

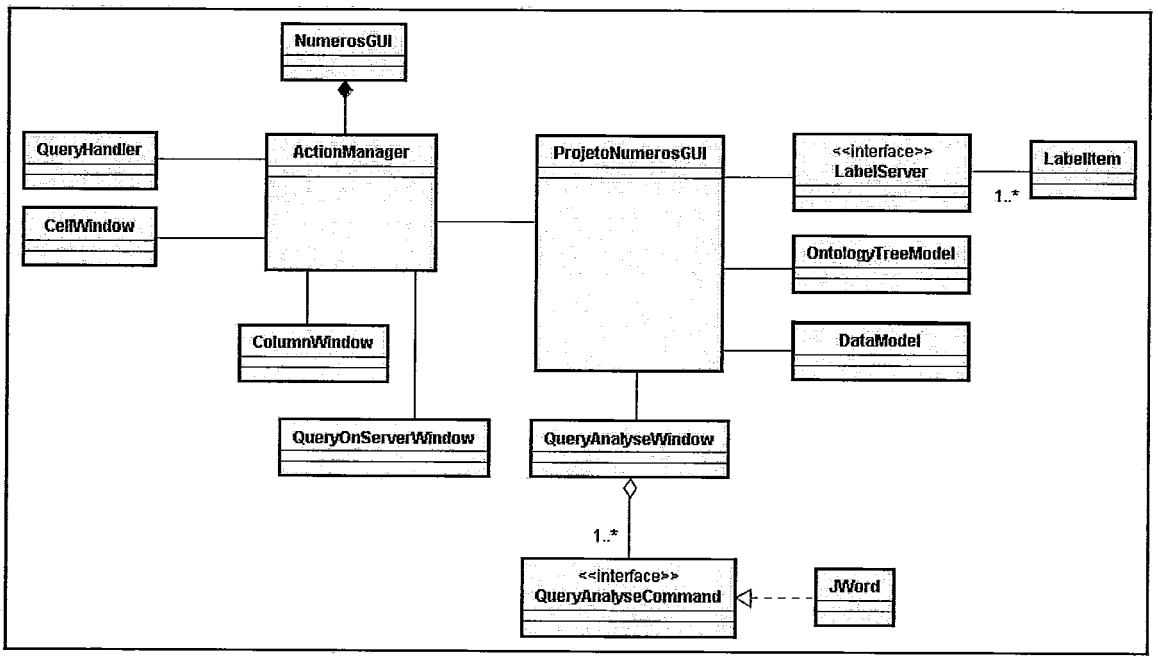


Figura 27 - Diagrama de Classes (UML 1.4) da Camada de Interface com o Usuário

A Figura 27 apresenta um diagrama UML simplificado do Componente Interface de Consulta Semântica (*vide* Figura 8 e Figura 9), apresentando as classes implementadas neste trabalho. Este capítulo descreve em maiores detalhes cada uma destas classes.

5.7.1 – MVC - Camada do Modelo

A camada Modelo no padrão MVC é composta pelas classes de negócio, e representam o domínio da aplicação. Na Camada de Interface Gráfica o domínio da aplicação é representado pelas classes do Formato Comum (vide seção "Formato Comum"), portanto é papel das classes do modelo, manipular instâncias de documentos no Formato Comum.

A arquitetura Swing para desenvolvimento de aplicações visuais em Java também é implementada utilizando o modelo MVC, desta forma as classes que representam componentes visuais necessitam de uma outra classe que represente o Modelo com os dados a serem exibidos. Esta sessão descreve a construção destas classes, tanto para os componentes do Swing quanto para a representação do Formato Comum.

A manipulação de documentos é realizada pela classe `QueryHandler`, e as classes `OntologyModel` e `DataModel` representam os dados para os componentes Swing. As sessões a seguir descrevem o papel destas classes e suas características.

5.7.1.1 – Classe `QueryHandler`

A classe `QueryHandler` é responsável pela manipulação de documentos no Formato Comum. Esta classe possui métodos que recebem documentos em diversos formatos, como arquivos XML ou arquivos de texto armazenados na memória e converte para documentos no Formato Comum. Entre as operações destacam-se ainda a manipulação das consultas, a seleção de trechos do documento, a exclusão e alteração de valores, e a conversão do objeto que representa a consulta para um texto XML.

5.7.1.2 – Classes `OntologyTreeModel` e `DataModel`

A API gráfica do sistema gráfico Swing é feita sobre a arquitetura MVC, assim como o Projeto Números. Alguns componentes de exibição de dados como tabelas,

listas e árvores de hierarquias necessitam que o programador forneça a implementação da classe que represente a camada de Modelo, ou seja, uma classe que forneça os dados para ser exibidos por estes componentes. As classes `OntologyTreeModel` e `DataModel` são responsáveis pela implementação dessas classes da camada de modelo, necessárias para a exibição de dados para os componentes Swing.

A classe `OntologyTreeModel` representa um modelo para o componente `JTree` da API Swing, a Figura 18 mostra esta classe sendo utilizada para representar os dados de uma Ontologia. Os métodos desta classe destinam-se a manipulação de dados e respostas para eventos visuais. Há, por exemplo, métodos para identificar se um determinado objeto selecionado na listagem é a raiz da árvore.

A classe `DataModel` é responsável pelos dados exibidos na janela principal da interface gráfica recebendo uma instância de uma Tabela do Formato Comum e provendo dados para uma `JTable`.

5.7.2 – MVC – Camada da Visão

O papel da Camada de Visão no padrão MVC é interagir com o usuário, recebendo suas solicitações e respondendo a estas com a representação das informações solicitadas. Assim as classes desta sessão são responsáveis pela exibição de dados, pela construção de componentes normalmente visuais que provêem a interação do usuário com o sistema. As subseções a seguir descrevem o papel de cada classe que compõe esta camada na Interface Gráfica do Projeto Números.

5.7.2.1 – Classe `ProjetoNumerosGUI`

A classe `ProjetoNumerosGUI` é a principal tela de interação com o usuário, é responsável pela construção e manipulação das consultas no Formato Comum, bem como a comunicação com o servidos do projeto.

Nesta classe é possível visualizar e manipular a consulta e suas propriedades de forma gráfica e numa interface rica (*Rich Interface*). A Figura 14 mostra a tela principal desta classe, gerada em tempo de execução, onde é possível observar uma consulta de exemplo, no centro da tela em uma representação gráfica e na parte interior no Formato Comum.

Como principal tela da interface gráfica com o usuário a classe `ProjetoNumerosGUI` é responsável por prover os menus e funcionalidades básicas

de uma aplicação gráfica. A Figura 14 exibe menus típicos de aplicações gráficas e os seguintes botões com suas respectivas ações:

1. *New* (novo) – cria uma nova consulta limpando todos os dados que estão na consulta atual;
2. *Open* (abrir) – abre uma consulta do sistema de arquivos;
3. *Server* (servidor) – abre uma consulta através de uma conexão com o servidor e exibe seu resultado. A seção "Classe " descreve detalhadamente a função deste botão;
4. *Save* (salvar) – salva a instância do documento localmente, mesmo que este tenha sido aberto remotamente;
5. PDF – exporta a representação gráfica da consulta (a tabela exibida na Figura 14) para um documento PDF;
6. Excel – exporta a representação gráfica da consulta para um documento Excel;
7. *Print* (imprimir) – imprime a representação gráfica da consulta;
8. *Reset* (restaurar) – limpa a consulta atual, removendo apenas os dados e mantendo as configurações de colunas;
9. *Analyse* (analisar) – envia a consulta para análise. A seção "Classe `QueryAnalyse`" descreve detalhadamente a função deste botão;
10. *Search* (pesquisar) – envia a consulta no Formato Comum para processamento pelo servidor.

Esta classe mostra também quatro abas para melhor compreensão e representação dos dados, são as abas e seus respectivos objetivos:

- *Query Editor* (editor de consulta) – destinada à construção e exibição da consulta, conforma exibido na Figura 14;
- *Properties* (propriedades) – responsável pelas propriedades da consulta, a seção "Propriedades da Consulta" é destinada a esta aba;
- *Filters* (filtros) – aba responsável pela filtragem da qualidade dos dados, a seção "5.7.2.3 –Filtros de qualidade de dados" é destinada a esta aba;
- *Results* (resultados) – a quarta e última aba, é responsável pela exibição do resultado, exibido na Figura 21;

5.7.2.2 – Propriedades da Consulta

Para representar informações sobre os documentos o Formato Comum utiliza o conjunto de marcações do Dublin Core (DCMI, 2006). O Dublin Core – DC – é um padrão para identificação de documentos, representado por um conjunto de marcações que permite que sistemas heterogêneos sejam capazes de identificar e trocar informações.

As diversas camadas do Projeto Números agregam informações e adicionam marcações Dublin Core à consulta, como a data da última atualização. A classe `ProjetoNumerosGUI` permite que o usuário do sistema altere alguns dados sobre a consulta que está criando, isto é, permite que altere algumas marcações DC. A Figura 15 mostra a tela para edição de propriedades na área denominada *Query Properties* (Propriedades da Consulta), onde é possível a edição de informações sobre o criador, título, identificador e descrição da consulta em edição.

A Figura 15 mostra também uma área denominada *Ontology* (Ontologia) que é destinada à edição e associação da ontologia à consulta em edição. Associar uma ontologia à consulta significa restringi-la (a consulta) ao escopo definido pela Ontologia escolhida, ou seja, a busca pelos dados será restrita às definições da ontologia escolhida. A escolha das Ontologias utilizadas por esta consulta restringe o valor para as colunas desta consulta.

Ainda na área de Ontologias é possível enviar uma nova Ontologia para o servidor ou abrir o editor de Ontologias (esta funcionalidade será explicada em maior profundidade na seção "COE – Editor Colaborativo de Ontologia").

5.7.2.3 – Filtros de qualidade de dados

O capítulo anterior descreve um refinamento quali-quantitativo dos dados que o usuário está interessado. Para alcançar esse objetivo, o Projeto Números possui um servidor de qualidade de dados que é responsável entre outras atividades:

- Manter e disponibilizar uma base de características qualitativas sobre dados;
- Responder questões sobre qualidade dos dados qualificando-o de acordo com sua base de características;

Cada característica do servidor de qualidade é um conjunto composto por um título e um coleção de valores possíveis para esta característica, cada conjunto desta estrutura de dados é representada pela classe `LabelItem`. A interface `LabelServer`

representa os serviços mínimos de um servidor de qualidade de dados utilizado pela interface gráfica do Projeto Números, assim toda classe que representa uma conexão com um servidor de qualidade (ou que representa essas características de outra forma) precisa implementá-la, como é o caso da implementação padrão através da classe `LabelServerDefault`. Assim classe `ProjetoNumerosGUI` realiza consultas a esse servidor de qualidade de dados e monta dinamicamente uma interface gráfica para o usuário aplicar filtros sobre sua busca. A Figura 16 mostra a aba de filtros em destaque.

5.7.2.4 – Classe `CellWindow`

A classe CellWindow é responsável pelo editor de células do Formato Comum. Cada célula pode armazenar uma coleção de valores, que diferem um dos outros, na data de sua publicação, na sua origem, na grafia da informação ou ainda no idioma em que foi publicado. Esta classe provê um editor que dispõe essas informações de forma tabular, de modo que o usuário possa visualizar os dados e editá-los como numa planilha eletrônica.

Há uma coluna especial no editor de células, a coluna que marca qual das linhas disponíveis é a linha padrão, isto é, qual a linha de preferência para o usuário naquela consulta ou na resposta da consulta. A informação de linha padrão poderá ser utilizada pelo servidor de qualidade de dados.

A Figura 17 exibe a edição da célula 1x1 da consulta exibida na Figura 14. É possível observar duas grafias para o nome do país Brasil, fato que é justificado pelo idioma diferente entre as linhas.

5.7.2.5 – Classe `ColumnWindow`

A classe `ColumnWindow` é responsável pelo editor de colunas, uma tela que auxilia na construção de consultas onde o usuário acrescenta informações sobre os dados que aquela coluna armazenará.

O usuário poderá informar para cada coluna o nome, o tipo de dado armazenado e quais folhas das Ontologias representam seus dados. A Figura 18 exibe uma captura de tela do editor de colunas.

5.7.2.6 – Classe `QueriesOnServerWindow`

A Classe `QueriesOnServerWindow` é iniciada quando o usuário pressiona o botão *Server* (Servidor) da Figura 14. Esta classe mostra uma listagem com as consultas no servidor e seus respectivos estados de processamento.

Após selecionar uma linha, a classe de Controle solicita ao servidor o documento no Formato Comum escolhido e repassa para a classe Modelo. Com o documento aberto o usuário poderá manipulá-lo como uma nova consulta ou ainda exportar o resultado para diversos formatos de arquivos.

5.7.2.7 – Classe `QueryAnalyse`

A classe `QueryAnalyse` é iniciada quando o usuário pressiona o botão *Analyse* (Analisar) na Figura 14. Esta classe é responsável por executar um conjunto de análises na consulta do usuário com o objetivo de enriquecê-la ou extrair alguma informação para o usuário. A Figura 19 exibe esta tela e seus analisadores de consulta disponíveis.

Esta classe está preparada para a adição de novos componentes de análise de consulta, onde é necessário a implementação da interface *Command* (do pacote "`br.ufrj.numeros.userinterface.view.analyse.Command`"). Este trabalho apresenta uma implementação de análise utilizando o WordNET (através da implementação do JWord 3.0) (JOHAR e SIMHA, 2006; MILLER, 1995) para encontrar sinônimos para os valores fornecidos pelo usuário.

5.7.3 – Camada de Controle

A camada de Controle na arquitetura MVC é responsável pela ligação entre as camadas de visão e modelo. Sua principal atividade é ouvir as solicitações da camada de visão, invocar uma ou mais operações do modelo e retornar os dados, quando necessário, para a camada de visão.

Para implantar este funcionamento, as classes da camada de Visão disparam eventos para cada ação realizada pelo usuário. Estes eventos são esperados e respondidos pela classe `ActionManager` – classe de controle para a camada visual.

A classe `ActionManager` tem um mapeamento para cada ação da interface gráfica, assim quando o usuário pressiona o botão *Open* (abrir) da Figura 14, a seguinte sequência de eventos acontece:

- A classe `ProjetoNumerosGUI` – visão – dispara um evento denominado *"loadQuery"*;
- A classe `ActionManager` – controlador – recebe este evento;
- O controlador abre a janela de seleção de arquivos do sistema operacional;
- Se o usuário escolheu o arquivo, o controlador envia este arquivo para a classe `QueryHandler` – modelo;
- O modelo verifica se o arquivo é válido e o carrega para a memória como um documento no Formato Comum;
- O controlador define que o arquivo da visão é o arquivo que está em uso pelo modelo;
- O foco da aplicação volta para a interface com o usuário.

5.8 – Camada de Dados e Conhecimento

A camada de Dados e Conhecimento é responsável pelo processamento dos documentos no Formato Comum, executando as consultas e montando o resultado. Esta camada atende também às requisições da Interface gráfica.

Assim como a camada de Interface com o usuário, esta camada possui diversas Interfaces que permitem a expansão de funcionalidades e/ou a configuração da forma como os documentos são processados. Esta sessão descreve as classes que compõe esta camada.

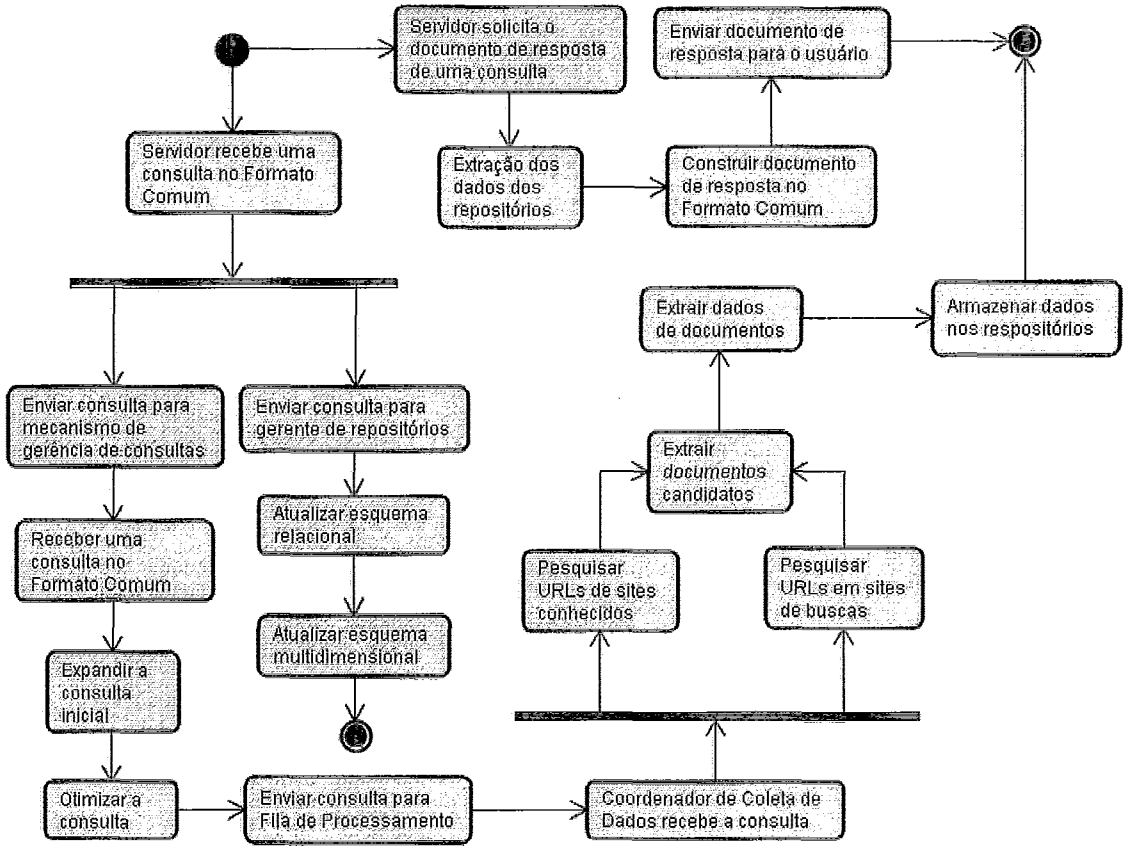


Figura 28 - Diagrama de Atividades (UML 1.4) demonstrando o processamento de dados na Camada de Dados e Conhecimento

A Figura 28 exibe o diagrama de atividades referente ao processamento de dados na Camada de Dados e Conhecimento. Este diagrama apresenta o fluxo típico de processamento de um documento no Formato Comum a partir de sua chegada ao servidor.

5.8.1 –Interface `Servidor`

Define as funcionalidades básicas que o Servidor do Projeto Números deve prover. A escolha de uma interface é uma estratégia que visa à expansibilidade, uma vez que é possível ter diversas implementações para uma interface e assim atuem como um servidor para o projeto.

5.8.2 –Classe `NumerosServer`

A classe `NumerosServer` implementa a interface Servidor, respondendo às solicitações da camada de Interface com o Usuário. Esta classe atua como coordenador

das atividades realizadas pelas outras classes que compõe esta camada, lendo os arquivos de configurações e iniciando os serviços das demais classes.

Existem diversas configurações para o Projeto Números, destinadas a definir os serviços básicos como acesso aos bancos de dados e classes que programam serviços manipulados pelo Projeto, é papel da classe `NumerosServer` ler este arquivo de configuração e instanciá-los como serviços.

5.8.3 – Classe `QueryManangementEngine`

A classe `QueryManangementEngine` representa o mecanismo de gerência de consultas, responsável por receber os documentos no Formato Comum e distribuir entre as classes para processamento.

Esta classe gerencia o fluxo de informações entre as outras classes componentes deste mecanismo, esta sessão descreve em maior profundidade estas classes componentes.

5.8.3.1 – Classe `QueryScheduler`

Esta classe é responsável por organizar a fila de processamento das consultas, nesta implementação a fila é simples, isto é, as consultas são organizadas na ordem de chegada. Desenvolvedores podem determinar outro comportamento para esta classe, como um analisador de consulta que leva em consideração o usuário ou ainda os dados acessados.

5.8.3.2 – Classe `QueryExpansion`

A classe `QueryExpansion` realiza a tarefa de expandir a consulta original, de forma que o processamento da consulta por outras camadas recupere um conjunto maior de dados. A expansão de consultas tem como objetivo popular o repositório de dados, preparando-o assim para responder às consultas similares.

Neste trabalho a implementação do componente de expansão de consultas percorre a ontologia associada a cada coluna e adiciona os sinônimos para os valores, isto é, quando existem.

A consulta expandida é armazenada numa sessão determinada no documento Formato Comum, preservando assim a consulta original que será utilizada para construção do resultado.

5.8.3.3 – Classe `QueryOptimizer`

A otimização das consultas é realizada pela classe `QueryOptimizer`. A otimização é um processo de transformação da consulta original que busca uma melhoria na consulta resultante para uma execução de menor custo computacional.

5.8.4 – Classe `KnowledgeBase`

A classe `KnowledgeBase` representa o repositório de dados que armazena os documentos no Formato Comum e as Ontologias. Esta classe é responsável pela gerencia destes documentos, atendendo as consultas e interagindo com a Camada da Visão, provendo dados para reutilização durante a construção das consultas.

5.8.5 – Classe `RepositorySchemaManager`

O gerente de esquemas de repositórios é implementado pela classe `RepositorySchemaManager`, cujo papel é coordenar a distribuição de trechos de documentos no Formato Comum e Ontologias para as classes `RelationalSchemaManager` e *Data Warehouse*`SchemaManager`. O papel deste mecanismo é manter consistente o esquema de representação de dados dos repositórios, isto é, ao enviar uma nova consulta, o repositório precisa ser adaptado para receber dados provenientes de seu processamento.

As sessões a seguir descrevem em maiores detalhes como os documentos e Ontologias são processados e como cada dado contido será transformado e artefatos dos esquemas Relacional e *DataWarehouse*.

5.8.5.1 – Classe `RelationalSchemaManager`

A classe `RelationalSchemaManager` gerencia o esquema do repositório relacional, é responsável por garantir que os repositórios estejam preparados para armazenar os dados resultantes das consultas.

Esta classe recebe documentos no Formato Comum e Ontologias, e os processam gerando comandos SQL para criação ou atualização de tabelas do esquema relacional. Para alcançar seus objetivos esta classe faz uso do pacote DDLUtils da Apache, que permite a manipulação de esquemas de SGBD como objetos Java (APACHE DB GROUP, 2006).

```
1  <database name="CIAWORLDFACT">
2   <table name="ONTOLOGIA">
3    <column name="ID" primaryKey="true" required="true" type="INTEGER" size="10" autoIncrement="true"/>
4    <column name="NAME" primaryKey="false" required="false" type="VARCHAR" size="254" autoIncrement="false"/>
5    <column name="VALUE" primaryKey="false" required="false" type="CLOB" size="1048576" autoIncrement="false"/>
6   </table>
7   <table name="GEOGRAPHY">
8    <column name="GEOGRAPHY_ID" primaryKey="true" required="true" type="INTEGER" autoIncrement="true"/>
9    <column name="LOCATION" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
10   <column name="GEOGRAPHICCOORDINATES" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
11   <column name="MAPREFERENCES" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
12   <column name="AREA" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
13   <column name="AREACOMPARATIVE" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
14   <column name="LANDBOUNDARIES" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
15   <column name="COASTLINE" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
16   <column name="MARITIMECLAIMS" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
17   <column name="CLIMATE" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
18   <column name="TERRAIN" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
19   <column name="ELEVATIONEXTREMES" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
20   <column name="NATURALRESOURCES" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
21   <column name="LANDUSE" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
22   <column name="IRRIGATEDLAND" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
23   <column name="NATURALHAZARDS" primaryKey="false" required="false" type="VARCHAR" size="50" autoIncrement="false"/>
24  </table>
```

Figura 29 – Representação de esquema relacional com DDLUtils

A Figura 29 exibe um trecho de um documento gerado para representação de um esquema relacional usado pelo DDLUtils. A classe `RelationalSchemaManager` gera tabelas e cria relacionamentos entre tabelas existentes de acordo com o documento recebido.

5.8.5.2 – Classe `DataWarehouseSchemaManager`

A classe `DataWarehouseSchemaManager` é responsável pelo esquema do repositório *DataWarehouse*, processando trechos e documentos no Formato Comum e Ontologias garantindo que a estrutura de organização dos dados seja adequada para a realização e visualização de consultas OLAP.

Esta classe também usa o XMLBeans (*vide* seção 5.5 – Formato Comum) para construção de um pacote de classes para manipulação de documentos de configuração do Mondrian.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Schema name="Numeros" xmlns="mondrian">
3    <Cube name="COUNTRIESPOPPIB">
4      <Dimension name="COUNTRY">
5        <Hierarchy name="COUNTRY" hasAll="true">
6          <Level name="COUNTRY" column="COUNTRY" uniqueMembers="false"/>
7        </Hierarchy>
8      </Dimension>
9      <Dimension name="ANO">
10       <Hierarchy name="ANO" hasAll="true">
11         <Level name="ANO" column="ANO" uniqueMembers="false"/>
12       </Hierarchy>
13     </Dimension>
14     <Dimension name="LINHA">
15       <Hierarchy name="LINHA" hasAll="true">
16         <Level name="LINHA" column="LINHA" uniqueMembers="false"/>
17       </Hierarchy>
18     </Dimension>
19     <Measure name="POPULATION" column="POPULATION" aggregator="sum"/>
20     <Measure name="GDP" column="GDP" aggregator="sum"/>
21   </Cube>
22 </Schema>
```

Figura 30 - Documento representando o esquema multidimensional do Mondrian

A Figura 30 exibe um trecho de um documento Mondrian, representando o esquema multidimensional, é possível observar elementos de *Data Warehouse* neste documento como Cubo (*Cube*), Dimensões (*Dimensions*) e Medidas (*Measures*). Este documento é gerado pela classe `DataWarehouseSchemaManager` através do processamento das consultas no Formato Comum e também das Ontologias.

5.9 – Considerações sobre a implementação

O Projeto Números é uma arquitetura em desenvolvimento utilizada por outros trabalhos acadêmicos, a implementação apresentada neste capítulo é para ser utilizada como referência por outros trabalhos que venham a suplantar as funcionalidades originais apresentadas na arquitetura.

O uso de software livre no desenvolvimento deste trabalho encoraja a livre distribuição desta aplicação, de forma que a comunidade acadêmica possa colaborar com o seu desenvolvimento, bem como voluntários que desejem aprimorar os recursos e funcionalidades aqui propostos.

Tabela 5 - Resumo da situação de implementação dos componentes do Projeto Números

Componente	**Camada**	**Situação**
Formato Comum		Implementado
Interface OLAP	Interface com Usuário	Provido pelo JPivot

Interface de Consulta Semântica – SQI	Interface com Usuário	Implementado
Editor de Consultas no Formato Comum	Interface com Usuário / SQI	Implementado
Analisador de Consultas	Interface com Usuário / SQI	Implementado
Filtro de Atributos dinâmico	Interface com Usuário / SQI	Implementado
Associação de Ontologias à consulta	Interface com Usuário / SQI	Implementado
Comunicação com o Servidor e recuperação de resultados de consultas processadas	Interface com Usuário / SQI	Implementado
Camada de Dados e Conhecimento – CDD		Parcialmente implementado
Mecanismo de Extração de Informações – MEI	CDD	Proposto
Extrator de Informações de Tabela	CDD / MEI	Proposto
Extrator de Informações de Textos	CDD / MEI	Proposto
Mecanismo de Qualidade de Dados	CDD / MEI	Proposto
Criador de Consultas	CDD / MEI	Proposto
Mecanismo de Conversão de Formato	CDD	Proposto
Gerente de Esquemas de Repositório – GER	CDD	Implementado
Gerente de Esquema *Data Warehouse*	CDD / GER	Implementado
Gerente de Esquema Relacional	CDD / GER	Implementado
Mecanismo de Demanda de Dados	CDD / GER	Proposto
Mecanismo de Solução de Consultas	CDD / GER	Implementado
Mecanismo de Gerência de Consultas – MGC	CDD	Parcialmente implementado
Interpretador de Consultas	CDD / MGC	Implementado
Mecanismo de Expansão de Consultas	CDD / MGC	Protótipo
Otimizador de Consultas	CDD / MGC	Protótipo
Escalonador de Consultas	CDD / MGC	Protótipo
Mecanismo Alimentador de Dados	CDD	Proposto
Camada de Coleta de Dados – CCD		Proposto
Coordenador de Coleta de Dados	CCD	Proposto
Mediadores e Adaptadores	CCD	Proposto
Mecanismos de Meta Busca	CCD	Proposto
Mecanismo de Crawling Inteligente	CCD	Proposto
Identificador de URL	CCD	Proposto
Escalonador de URL	CCD	Proposto
Analisador de Documentos	CCD	Proposto
Estratégias de Crawling	CCD	Proposto
Cache de Documentos	CCD	Proposto
Fila de Consultas	CCD	Proposto

Capítulo 6 – Simulação de uso do sistema

O Projeto Números é um projeto extenso, a arquitetura apresentada aqui é implementada por este e outros trabalhos, assim para demonstrar a sua viabilidade, este capítulo simula a busca por dados, resultante do processamento de uma consulta no Formato Comum.

A sessão a seguir descreve uma consulta construída utilizando a implementação de Interface e Servidor demonstrada no capítulo anterior. O objetivo é demonstrar como os dados podem ser obtidos a partir de uma consulta construída com informações coletadas na interface com o usuário.

Para melhor avaliar os resultados as seguintes considerações foram seguidas:

1. O site de busca utilizado foi o *Google*;
2. Para cada consulta realizada, o resultado escolhido está entre os dez primeiros;
3. O critério de seleção do resultado a ser apurado é sempre um site diferente daquele utilizado pela consulta anterior, isto é, se a consulta *x* coletou dados do site *CIA World Facts*, a consulta *x'* não poderá utilizar este mesmo site;
4. As colunas com valores textuais de exemplo determinam o que estamos buscando e, portanto, são utilizadas como pivô na elaboração das consultas na Web;
5. Para o resultado utilizado, uma pequena imagem destacará o dado que estamos buscando e como este é exibido na página de origem. Também é exibido o trecho de código HTML que contém o dado em questão;
6. Apenas documentos HTML foram avaliados nesta simulação, desconsiderando planilhas Microsoft Excel, documentos Microsoft Word, imagens, etc.;

6.1 – A consulta: País, População e PIB.

A consulta por País, População e PIB é a consulta de exemplo utilizada no decorrer deste trabalho, a Figura 31 mostra esta consulta graficamente e a Figura 32 exibe a consulta no Formato Comum. O usuário fornece alguns países, associando cada coluna a uma folha da ontologia *CIA World Facts*, e depois solicita o preenchimento dos valores de população e PIB para cada país fornecido. A partir das informações coletadas, a Camada de Coleta de Dados deve ser capaz de gerar as seguintes consultas e produzindo os seguintes resultados:

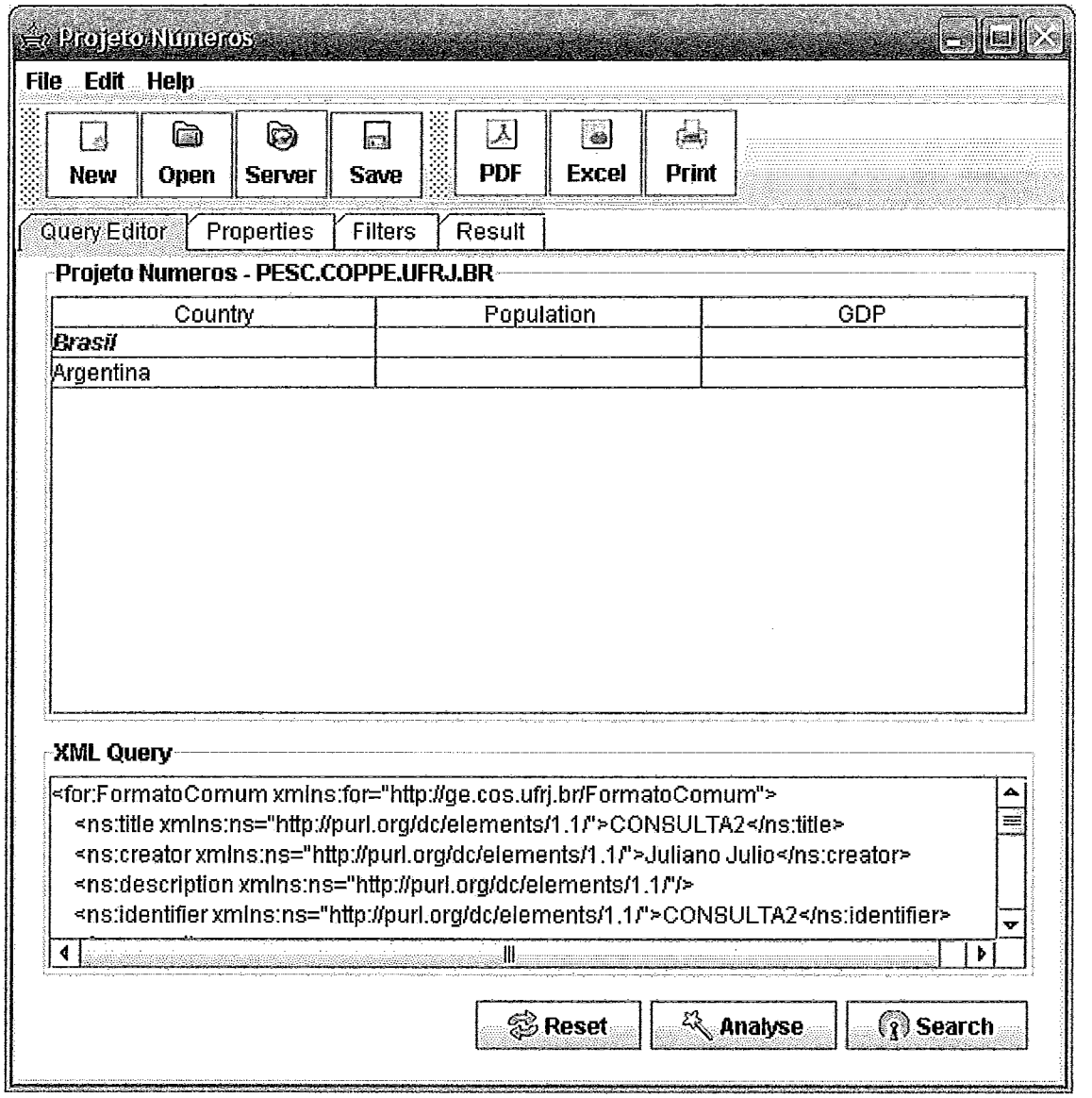


Figura 31 – Consulta por País, População e PIB em edição no editor visual.

```
<for:FormatoComum xmlns:for="http://ge.cos.ufrj.br/FormatoComum">
  <ns:title xmlns:ns="http://purl.org/dc/elements/1.1/"/>
  <ns:creator xmlns:ns="http://purl.org/dc/elements/1.1/"/>
  <ns:description xmlns:ns="http://purl.org/dc/elements/1.1/"/>
  <ns:identifier xmlns:ns="http://purl.org/dc/elements/1.1/"/>
  <for:consulta>
    <for:cabecalho>
      <for:coluna>
        <for:nome>Country</for:nome><for:tipo>String</for:tipo><for:ontologia>/CIA World Fact/CIA World Fact/Country/Name/</for:ontologia>
      </for:coluna>
      <for:coluna>
        <for:nome>Population</for:nome><for:tipo>Float</for:tipo><for:ontologia>/CIA World Fact/CIA World
        Fact/Country/People/Population/</for:ontologia>
      </for:coluna>
      <for:coluna>
        <for:nome>GDP</for:nome><for:tipo>Currency</for:tipo><for:ontologia>/CIA World Fact/CIA World
        Fact/Country/Economy/GDP/</for:ontologia>
      </for:coluna>
    </for:cabecalho>
    <for:linha>
      <for:celula>
        <for:valor>
          <for:texto>Brazil</for:texto>
          <ns:date xmlns:ns="http://purl.org/dc/elements/1.1/">2006-07-10T16:43:41.218-03:00</ns:date>
          <ns:source xmlns:ns="http://purl.org/dc/elements/1.1/"/>
        </for:valor>
        <for:valor>
          <for:texto>Brasil</for:texto>
          <ns:date xmlns:ns="http://purl.org/dc/elements/1.1/">2006-07-10T16:43:41.218-03:00</ns:date>
          <ns:source xmlns:ns="http://purl.org/dc/elements/1.1/"/>
        </for:valor>
      </for:celula>
      <for:celula><for:valor><for:texto/></for:valor></for:celula>
      <for:celula><for:valor><for:texto/></for:valor></for:celula>
    </for:linha>
    <for:linha>
      <for:celula>
        <for:valor>
          <for:texto>Argentina</for:texto>
          <ns:date xmlns:ns="http://purl.org/dc/elements/1.1/">2006-07-10T16:29:05.078-03:00</ns:date>
          <ns:source xmlns:ns="http://purl.org/dc/elements/1.1/"/>
        </for:valor>
      </for:celula>
      <for:celula><for:valor/></for:celula>
      <for:celula><for:valor/></for:celula>
      <for:celula><for:valor/></for:celula>
    </for:linha>
  </for:consulta>
</for:FormatoComum>
```

Figura 32 - Consulta por País, População e PIB, representação no Formato Comum.

- População do Brasil
  - Consulta realizada: "population (Brazil OR Brasil)";
  - Site encontrado: Universidade Holandesa de Utrecht: POPULATION STATISTICS
  - URL: http://www.library.uu.nl/wesp/populstat/Americas/brazilc.htm

BRAZIL

historical demographical data of the whole country

population	year	population	year	population	year	population	year	population	year	population	year
	16XX	9797,0	1870	17984,0	1900	33568,0	1930	70281,0	1960	148477,0	1990
	17XX	9947,0	1871	18392,0	1901	34256,0	1931	72262,0	1961	151152,0	1991
	17XX	10099,0	1872	18782,0	1902	34957,0	1932	74279,0	1962	153850,0	1992
	180X	10289,0	1873	19180,0	1903	35673,0	1933	76352,0	1963	156486,0	1993
	180X	10486,0	1874	19587,0	1904	36404,0	1934	78483,0	1964	159143,0	1994
	181X	10687,0	1875	20003,0	1905	37150,0	1935	80674,0	1965	161374,0	1995
	182X	10891,0	1876	20427,0	1906	37911,0	1936	82926,0	1966	161365,0	1996

Figura 33 – Captura de tela da página de resultado

```
<TD ALIGN="right"><FONT FACE="Arial"><B>167967,0</B></FONT></TD>
<TD ALIGN="left"><FONT FACE="Arial" SIZE=-1>1999</FONT></TD>
</TR>
```

Figura 34 – Trecho do documento HTML com o resultado

- População Argentina
  - Consulta realizada: "population Argentina"
  - Site encontrado: CIA - The World Factbook – Argentina
  - URL: http://www.cia.gov/cia/publications/factbook/geos/ar.html

People — **Argentina**

Population:

39,921,833 (July 2006 est.)

Figura 35 – Captura de tela da página de resultado

```
<td class="FieldLabel" valign="top" width="20%">
        <div align="right">Population:</div>
</td>
<td bgcolor="#ffffff" valign="top" width="80%">


                <a href="../docs/notesanddefs.htn

                <a href="../fields/2119.html"><in

        <a href="../rankorder/2119rank.html"><imc

        <br>

        39,921,833 (July 2006 est.)
        </td>
```

Figura 36 – Trecho do documento HTML com o resultado

- PIB Brasil
  - Consulta realizada: "(gdp OR "Gross Domestic Product" OR "PIB") (Brazil OR Brasil)"
  - Site encontrado: Confederação Nacional da Indústria - Dados Econômicos
  - URL: http://www.cni.org.br/brasil/pib.htm

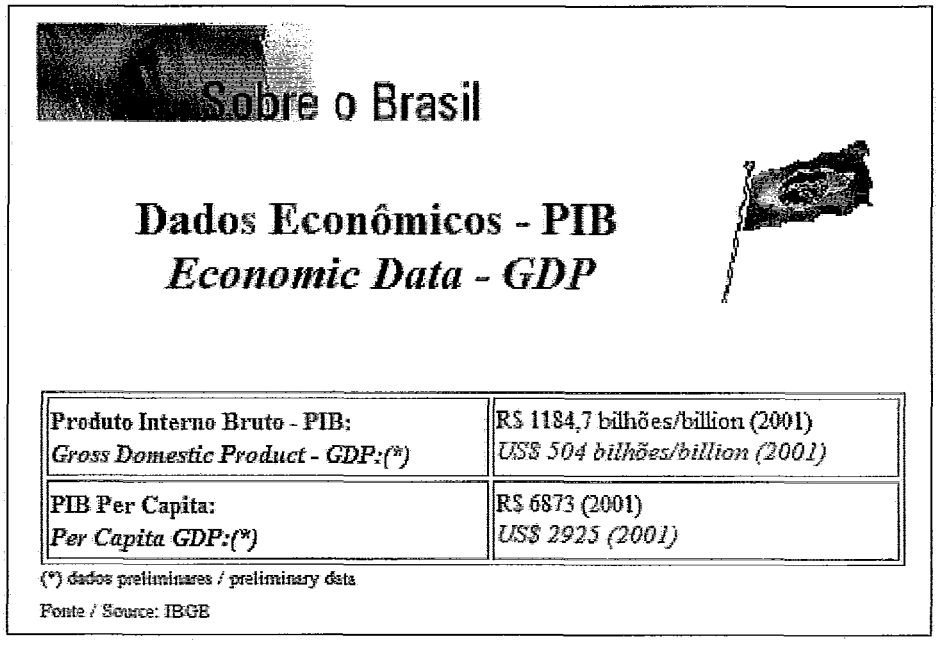


Figura 37 – Captura de tela da página de resultado

```
<td><b><font size="-1">Produto Interno Bruto - PIB:</font></b> 
<br><b><i><font size="-1">Gross Domestic Product - GDP:(*)</font></i></b></td>

        <td><font size="-1">R$ 1184,7 bilhões/billion (2001)</font><br>
          <i><font size="-1">US$ 504 bilhões/billion (2001)</font></i></td>
```

Figura 38 – Trecho do documento HTML com o resultado

- PIB Argentina
  - Consulta realizada: "(gdp OR "Gross Domestic Product" OR "PIB") Argentina"
  - Site encontrado: Energy Information Administration (EIA) - Argentina Country Analysis Brief
  - URL: http://www.eia.doe.gov/emeu/cabs/argentna.html

ECONOMIC OVERVIEW
Minister of the Economy: Roberto Lavagna
Currency: Peso
Financial Exchange Rate (1/11/05): US$1 = 2.96 Argentine Pesos
Gross Domestic Product (2003E): $127.4 billion **(2004E):** $153.1 billion **(2005F):** $167.3 billion

Figura 39 – Captura de tela da página de resultado

```
<p><strong>ECONOMIC OVERVIEW<br>
Minister of the Economy: </strong>Roberto Lavagna<br>
<strong>Currency:</strong> Peso<br>
<strong>Financial Exchange Rate ( 1/11/05):</strong> US$1 = 2.96 Argentine Pesos<br>
<strong>Gross Domestic Product (2003E):</strong> $127.4 billion <strong>(2004E):</strong> $153.1 billion <strong>(200
<strong>Real GDP Growth Rate (2003E):</strong> 8.7% <strong>(2004E):</strong> 8.0% <strong>(2005F):</strong> 5.3%<br>
```

Figura 40 – Trecho do documento HTML com o resultado

A simulação da busca por População e PIB para cada país recuperou um conjunto de dados que deveriam ser processados pela camada de Coleta de Dados. Para cada documento HTML encontrado na simulação, é necessário um processo para a extração dos dados, realizado pelo Mecanismo de Conversão de Formato. A Tabela 6 mostra os dados extraídos e suas respectivas consultas.

Tabela 6 - Resultados extraídos das consultas

Consulta	Origem	Dado extraído	Ano extraído
"population (Brasil OR Brazil)"	Universidade Holandesa de Utrecht: POPULATION STATISTICS	População: 167.967.000	1999
"Population Argentina"	CIA - The World Factbook – Argentina	População: 39.921.833	2006
"(gdp OR "Gross Domestic Product" OR "PIB") (Brazil OR Brasil)"	Confederação Nacional da Indústria - Dados Econômicos	PIB: U$ 504 bilhões	2001
"(gdp OR "Gross Domestic Product" OR "PIB") Argentina"	Energy Information Administration (EIA) - Argentina Country Analysis Brief	PIB: U$ 167,3 bilhões	2005

O processamento da consulta pelo Gerente de Esquemas de Repositórios gera comandos SQL para construção (ou atualização) das estruturas relacionais para armazenamento dos dados encontrados. Após esta etapa os dados são armazenados no SGBD nas estruturas relacionais criada pelo processamento da consulta.

COUNTRY	POPULATION	GDP	ANO	LINHA	ID	SOURCE
Brasil	167.967.000	<null>	1999	1	1	Universidade Holandesa de Utrecht: POPULATION STATISTI...
Brazil	<null>	U$504 bi	2001	1	2	Confederação Nacional da Indústria - Dados Econômicos
Argentina	39.921.833	<null>	2005	2	3	CIA - The World Factbook - Argentina
Argentina	<null>	U$167,3 bi	2005	2	4	Energy Information Administration (EIA) - Argentina Country An...

Figura 41 - Tabela gerada pelo Gerente de Esquema de Repositórios e preenchida pelo Mecanismo de Alimentação de Dados

A Figura 41 mostra uma tabela relacional gerada com o processamento da consulta no Formato Comum e os dados encontrados pelos passos anteriores.

A etapa final deste processo é a solicitação dos dados pelo usuário, etapa que dispara a recuperação dos dados dos repositórios. A Figura 42 exibe o resultado da consulta na interface gráfica do Projeto Números.

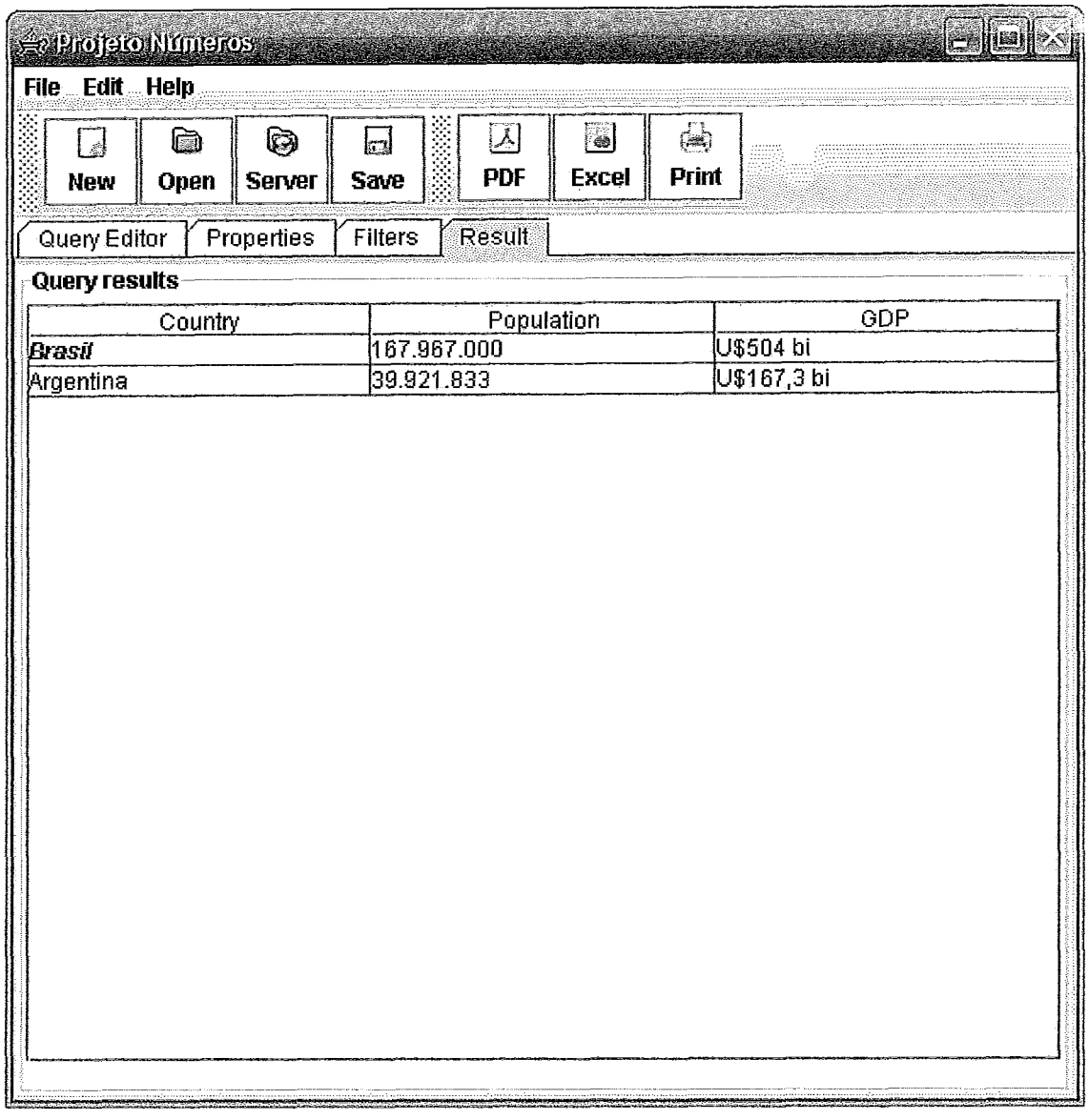


Figura 42 - Tela exibindo resultado da simulação

6.2 – Resultados da simulação

Este capítulo demonstra como os componentes não implementados do Projeto Números recuperariam dados. Parte do processamento da consulta responsável pela estruturação dos repositórios é executada pelos componentes apresentados neste trabalho, assim como a consulta construída na atual interface gráfica.

Capítulo 7 – Conclusão

O presente trabalho apresenta uma arquitetura para a recuperação de dados numéricos de repositórios de dados heterogêneos e uma implementação de referência para um sistema de construção e enriquecimento semântico de consultas.

Foi realizado um estudo sobre o modelo *Data Warehouse* e do projeto *e.dot* que utiliza a Internet como fonte de dados para um banco de dados *Data Warehouse*. Este estudo foi importante para a construção de um sistema capaz de recuperar, armazenar e disponibilizar dados organizados em bases heterogêneas.

Este trabalho especificou e apresentou a arquitetura de um sistema capaz de buscar e recuperar dados e documentos na Internet e de outros repositórios, a fim de responder a consultas de um usuário interessado em dados numéricos. A elaboração desta arquitetura assumiu a utilização de mecanismos de busca e construção de componentes capazes de identificar fragmentos de dados e extraí-los para representações mais formais, onde pudessem ser processadas e utilizadas para responder às requisições do usuário.

A troca de informações entre as camadas da arquitetura foi possível através do uso de documentos (Formato Comum) que representa a consulta do usuário e todos os metadados relativos à mesma.

Foi apresentada uma implementação de referência, demonstrando uma interface gráfica amigável que possibilita a elaboração e enriquecimento de consulta no Formato Comum. Esta interface permite que usuários sem conhecimento da arquitetura sejam capazes de realizar buscas.

A implementação de referência também demonstra determinados componentes da camada de Dados e Conhecimento, estes componentes são capazes de criar e gerenciar esquemas para banco de dados relacionais e OLAP, permitindo que os dados colhidos por outras camadas, possam ser armazenados e posteriormente recuperados pelo usuário através da consulta no Formato Comum, ou pela interface de consultas multidimensionais.

Este trabalho contribui com um sistema de buscas por exemplos, onde os resultados esperados são numéricos e estão armazenados em repositórios heterogêneos. Apesar de demonstrar uma arquitetura completa, não faz parte do escopo deste trabalho

apresentar uma versão completa do sistema que devido a sua vastidão. Apresentamos a especificação completa e detalhada da arquitetura que serviu de base para outros trabalhos que estudam o problema da busca e da recuperação de dados de forma mais abrangente.

Apesar de cumprir os objetivos iniciais, observa-se que há questões a serem tratadas em maior profundidade, como: o aprimoramento da interface cliente permitindo a inclusão de fórmulas para determinadas colunas; a edição colaborativa de consultas; e novos mecanismos de expansão da consulta.

O desenvolvimento deste trabalho leva a conclusão de que é viável o desenvolvimento de sistemas para consultas não convencionais, e o estudo de *Data Warehouse* e formatos dos dados na Internet leva a uma compreensão da organização dos dados de forma a ser possível automatizar o processo de um usuário manipulá-los indiretamente e sem ter conhecimento de sua estrutura. Fica claro que ao desenvolver um trabalho como este, novas possibilidades de pesquisa se abrem e idéias interessantes surgem aumentando as perspectivas de trabalhos a serem realizados.

7.1 – Trabalhos futuros

O Projeto Números é um projeto extenso e que faz parte deste e de outros trabalhos acadêmicos.

Destacamos funcionalidades que ainda não foram implementadas como os métodos para expansão, otimização e escalonamento das consultas no Formato Comum, durante o seu processamento no Mecanismo de Gerência de Consultas.

Entre alguns aperfeiçoamentos possíveis podemos destacar: a edição colaborativa de consultas; a utilização de fórmulas que definem como dados podem ser inferidos a partir de outros dados; novos mecanismos de análise de consultas.

Um trabalho interessante é a aplicação do Projeto Números em organizações onde os dados estão espalhados em diversas fontes e nossa ferramenta poderá propiciar a visão de um *Data Warehouse* virtual dos dados distribuídos.

Referências Bibliográficas

ANTONIOU, G.,VAN HARMELEN, F., 2004, *A Semantic Web Primer*, Cambridge, Massachusetts, Massachusetts Institute of Technology

APACHE DB GROUP, 2006, "DDLUtils". In: http://db.apache.org/ddlutils/, Accessed in 24/07/2006.

APACHE GROUP, 2006, "Apache Software Foundation".

BERRY, M. J. A.,LINOFF, G., 1997, *Data Mining Techniques: For Marketing, Sales, and Customer Support*, New York, USA, Wiley Computer Publishing

BOULLOSA, J. R. D. F., 2002, *Um ambiente para mineração de utilização da WEB* de Mestrado em Eng. de Sistemas e Computação, Universidade Federal do Rio de Janeiro - UFRJ

CHRYSAFIS, T., 2003, *On-Line Analytical Processing*. CITY Liberal Studies an Affiliated Institution of the University of Sheffield. http://www.city.academic.gr/s3ctit03/Abstracts/OLAP.pdf

COOLEY, R., MOBASHER, B., SRIVASTAVA, J., 1997, "Web Mining: Information and Pattern Discovery on the World Wide Web", Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97). http://citeseer.ist.psu.edu/cooley97web.html

DCMI, 2006, "Dublin Core Metadata Initiative". In: http://dublincore.org/, Accessed in 23/07/2006.

DEOGUN, J. S., RAGHAVAN, V. V., SARKAR, A., et al, 1997, "Data Mining: Research Trends, Challenges, and Applications", Boston, MA, USA. http://citeseer.ist.psu.edu/deogun97data.html

DOS SANTOS, A. C. O. G., 2001, *Organização de um Data Warehouse Clínico* de Mestrado em Eng. de Sistemas e Computação, Universidade Federal do Rio de Janeiro - UFRJ

GAGLIARDI, H., HAEMMERLÉ, O., MIGLIORI, D., et al, 2005, *Enriching a relational datawarehouse by integrating XML data : report ont the edot project*, Second Franco Japanese Workshop on Information Search Integration and Personnalization (ISIP)

GILL, H. S.,RAO, P. C., 1996, *The Official Client/Server Computing Guide to Data Warehousing*, USA, Que Corporation

HAN, J.,KAMBER, M., 2001, *Data Mining: Concepts and Techniques*, Academic Press

INMON, W. H., 1997, *Como construir o Data Warehouse*. tradução [da 2. ed. americana], Rio de Janeiro, Editora Campus Ltda.

JOHAR, K.e SIMHA, R., 2006, "JWord". http://www.seas.gwu.edu/~simhaweb/software/jword/

JPIVOT, 2006, "JPivot - JSP custom tag library". In: http://jpivot.sourceforge.net/, Accessed in 07/2006.

JULIAN HYDE, 2006, "Mondrian OLAP Server". In: http://mondrian.sourceforge.net.

KIMBALL, R., 1996, *The Data Warehouse Toolkit*. 1. ed., New York - USA, John Wiley & Sons

KIMBALL, R., REEVES, L., ROSS, M., et al, 1998, *The Data Warehouse Lifecycle Toolkit : Expert Methods for Designing, Developing, and Deploying Data Warehouses*, New York - USA, John Wiley & Sons

KOSALA,BLOCKEEL, 2000, "Web Mining Research: A Survey", *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM*, v. 2. http://citeseer.ist.psu.edu/kosala00web.html

LYMAN, P., VARIAN, H. R., SWEARINGEN, K., et al, 2003, "How Much Information?". In: http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/, Accessed in 03/09/2006.

MATTISON, R., 1997, *Data Warehousing and Data Mining for Telecommunications*, Artech House

MEZAOUR, A.-D., 2003, "Focused Search on the Web using WeQueL", Hamburg, Germany. http://citeseer.ist.psu.edu/663877.html

MEZAOUR, A.-D., 2004, *Filtering Web Documents for eDot, a food risk warehouse*

MILLER, G. A., 1995, "WordNet: a lexical database for English". *ACM Press*, New York, NY, USA. http://doi.acm.org/10.1145/219717.219748

NOY, N. F.,MCGUINNESS, D. L., 2001, *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory e Stanford Medical Informatics. http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html

PASSIN, T. B., 2004, *Explorer's Guide to the Semantic Web*, Greenwich, Manning Publications Co.

PEARSON, W., 2002, "MDX at First Glance: Introduction to SQL Server MDX Essentials". In: http://www.databasejournal.com/features/mssql/article.php/10894_1495511_1, Accessed in 16/07/2006.

RAY, E. T., 2000, *Aprendendo XML*. tradução [Learning XML], Rio de Janeiro, Campus 2001

RÖNNLUND, A. R.e ROSLING, O., 2006, "Free software for a world in motion". http://www.gapminder.org/Experiments/Texts/C5/FreeSoftwareForAWorldInMotion.pdf

ROSLING, H., 2006, "Live presentation, Hans Rosling at TED Conference 2006", TED - Technology Entertainment Design. http://www.ted.com/conference/flashpage.cfm?conferenceKey=2006&CFID=1350108&CFTOKEN=30927997

RUSTY, H. E., 2005, "Managing XML data: Native XML databases". In: http://www-128.ibm.com/developerworks/xml/library/x-mxd4.html?ca=dnt-623, Accessed in 07/10/2005.

SUN MICROSYSTEMS, 2006, "JSR 69: Java OLAP Interface (JOLAP)", Accessed in 11/07/2006.

W3C, 2005, "XML Schema". In: http://www.w3.org/XML/Schema, Accessed in 19/09/2005.

WIKIPEDIA, 2005a, "Ontology". In: http://en.wikipedia.org/wiki/Ontology, Accessed in 09/07/2005a.

WIKIPEDIA, 2005b, "XQuery". In: http://en.wikipedia.org/wiki/XQuery, Accessed in 07/10/2005b.

XEXÉO, G., VIVACQUA, A., DE SOUZA, J. M., et al, 2005, " COE: A collaborative ontology editor based on a peer-to-peer framework", v. Vol. 19, pp. 113-121, Advanced Engineering Informatics.

XML-DB, 2003, "XML:DB Initiative for XML Databases". In: http://xmldb-org.sourceforge.net/, Accessed in 08/10/2005.

XML-DB, 2004, "XUpdate - XML Update Language". In: http://xmldb-org.sourceforge.net/xupdate/, Accessed in 08/10/2005.

Apêndice A – XML e Modelo de Dados XML

XML é uma linguagem de marcação criada pela *World Wide Web Consortium* – W3C, que permite a representação de dados de forma textual e estruturada hierarquicamente. Seu uso inicial era para a dissociação do conteúdo dos documentos das suas características de apresentação ao usuário, mas atualmente sua maior utilização é como linguagem comum entre diferentes sistemas computacionais. (W3C, 2005)

A XML é um subconjunto da SGML[2] que é uma linguagem de marcação para representação de documentos eletrônicos, mas a sua grande flexibilidade e abrangência dificultam a criação de softwares para processá-la, o que limitou o seu uso apenas as grandes empresas que podem pagar pelo alto custo no desenvolvimento e manutenção dos ambientes SGML. (RAY, 2000)

As linguagens de marcações são linguagens que usam de um conjunto de símbolos para demarcar algumas partes de um documento. Um documento XML é apenas um simples arquivo de texto escrito sobre uma política de boas práticas, que garantem a legibilidade para humanos, e algumas regras de formatação:

- Toda marcação deve ser delimitada pelos símbolos "<" e ">";
- Toda marcação iniciada deve ser encerrada;
- Uma marcação é encerrada com o símbolo "/", de duas maneiras "`<elemento></ elemento >`" ou "`<elemento />`";
- Os elementos devem seguir uma estrutura hierárquica;
- Atributos (propriedades) dos elementos devem possuir valores entre aspas;
- Nomes de elementos e atributos devem possuir apenas letras, números e o caractere "_".

Um documento que seguir essas regras de formatação é considerado um documento **bem formado**. Um documento XML é denominado **válido** quando está em conformidade com um documento que restringe seu vocabulário (nomes de elementos e atributos), seu modelo de conteúdo e seus tipos de dados. Para validar um documento

[2] Standard Generalized Markup Language (ISO 8870, 1986)

XML, utilizam-se linguagens como a *Document Type Definition* – DTD e XML *Schema*.

O DTD é uma linguagem de marcação criada para definição de vocabulário para documentos SGML. Para ser utilizada em documentos XML, a DTD foi limitada a um subconjunto da linguagem original, o que torna possível o processamento de documentos XML e DTD por processadores SGML.

O uso de DTD para validação de XML não foi muito bem aceito pela comunidade, pois esta apresentava uma linguagem fortemente baseada em expressões regulares, mas com pouca expressividade para as restrições dos tipos de dados. Mas a maior desvantagem da DTD é o fato de utilizar uma linguagem diferente da XML, resultando em processadores específicos.

A W3C anunciou em 2001 a XML *Schema* e recomendou como alternativa ao DTD. XML *Schema* é um documento XML onde é possível descrever regras para o vocabulário, o modelo de conteúdo e tipos de dados para outros documentos XML (W3C, 2005). Entre outras vantagens, a XML *Schema* é extensível, ou seja, é possível a reutilização em outro documento XML *Schema*, criar tipos de dados derivados dos tipos primitivos e referenciar múltiplos *Schemas* a partir de um único documento.

A simplicidade na representação de dados em documentos XML gerou a necessidade de diversas ferramentas para manipular esses dados. A W3C criou vários grupos para definir padrões que pudessem atender essa necessidade, sem ferir os requisitos do XML. Uma dessas necessidades é capacidade de consultar partes dos documentos XML, bem como os dados armazenados, buscando suprir essa necessidade a W3C criou a XQuery, uma linguagem de consulta a documentos XML similar ao SQL.

Com um conjunto de regras bem definidas e o modelo de dados descrito pelo XML *Schema* a XML rapidamente se tornou o padrão para troca de dados entre aplicações. Essa troca de documentos gerou o custo adicional de armazenar esses documentos e as seguintes estratégias foram adotadas: os documentos inteiros armazenados em banco de dados; os documentos pré-processados e os dados de interesse do usuário armazenados nos banco de dados; os documentos armazenados no sistema de arquivos; ou simplesmente os documentos eram processados e eliminados.

Buscando uma alternativa ao armazenamento de documentos XML, algumas instituições iniciaram pesquisas em banco de dados de documentos XML. Esses bancos de dados XML armazenam coleções de instâncias de documentos XML, os metadados

estão armazenados em XML *Schema*, os dados em árvores de nós e as buscas nesses bancos XML são realizadas com XQuery. Entre os benefícios dos bancos XML estão (RUSTY, 2005; WIKIPEDIA, 2005b):

- Centralização de documentos de forma estruturada em repositórios, passíveis as técnicas de otimização dos sistemas de banco de dados;
- Recuperação de um subconjunto de dados ou ainda a construção de combinações de documentos através dos comandos XQuery;
- Ganho de performance com técnicas de bancos de dados, como criação de índices e organização de dados de melhor forma;
- Manipulação de grandes documentos;
- Extração do documento original e todas as suas propriedades.

A utilização da XQuery é limitada apenas a consultas, pois a linguagem não possui predicados para realizar operações de atualização, inserção ou exclusão de dados. Como alternativa a essa falta de recursos alguns desenvolvedores adotaram linguagens proprietárias para atingir essa funcionalidade, outros buscaram uma alternativa conjunta e criaram a XUpdate (RUSTY, 2005).

A XUpdate é uma linguagem criada pelo XML-DB, um comitê formado por empresas que buscam padronização de linguagem e desenvolvimento de soluções para banco de dados XML (XML-DB, 2003; XML-DB, 2004).

As pesquisas em bancos de dados XML estão buscando resolver os problemas e adicionar funcionalidades e padrões, mas já apresentam benefícios que justificam sua aplicabilidade.

Apêndice B – XML *Schema* do Formato Comum

```
<?xml version="1.0"?>
<xs:schema targetNamespace="http://ge.cos.ufrj.br/FormatoComum"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified" attributeFormDefault="unqualified"
 xmlns="http://ge.cos.ufrj.br/FormatoComum">
 <xs:import namespace="http://purl.org/dc/elements/1.1/"
  schemaLocation="dc.xsd" />
 <xs:element name="FormatoComum">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="dc:title" />
    <xs:element ref="dc:creator" />
    <xs:element ref="dc:subject" />
    <xs:element ref="dc:description" />
    <xs:element ref="dc:publisher" />
    <xs:element ref="dc:contributor" />
    <xs:element ref="dc:date" />
    <xs:element ref="dc:type" />
    <xs:element ref="dc:format" />
    <xs:element ref="dc:identifier" />
    <xs:element ref="dc:source" />
    <xs:element ref="dc:coverage" />
    <xs:element ref="dc:language" />
    <xs:element ref="dc:relation" />
    <xs:element ref="dc:rights" />
    <xs:element name="consulta" type="tabela" />
    <xs:element name="expandida" type="tabela" />
    <xs:element name="resultado" type="tabela" />
    <xs:element name="filtros" type="filtros"></xs:element>
    <xs:element name="ontologia" maxOccurs="unbounded" minOccurs="0">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="nome" type="xs:string"></xs:element>
       <xs:element name="valor" type="xs:string"></xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:complexType name="tabela">
  <xs:sequence>
   <xs:element name="cabecalho" maxOccurs="1" minOccurs="1">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="coluna" maxOccurs="unbounded">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="nome" type="xs:string" maxOccurs="1"
          minOccurs="1" />
         <xs:element name="tipo" type="xs:string" />
         <xs:element name="ontologia" type="xs:string" />
        </xs:sequence>
       </xs:complexType>
      </xs:element>
```

```
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:element name="linha" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="celula" maxOccurs="unbounded">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="valor" minOccurs="0" maxOccurs="unbounded">
           <xs:complexType>
            <xs:sequence>
             <xs:element name="texto" type="xs:string" />
             <xs:element ref="dc:date" />
             <xs:element ref="dc:source" />
             <xs:element ref="dc:language" />
            </xs:sequence>
           </xs:complexType>
          </xs:element>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
  <xs:complexType name="filtros">
   <xs:sequence>
    <xs:element name="filtro" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="nome" type="xs:string"></xs:element>
       <xs:element name="valor" type="xs:string"></xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
</xs:schema>
```