

DESCOBERTA COLABORATIVA DE CONHECIMENTO APLICADA À BASE DE DADOS DO PORTAL DE TRANSPARÊNCIA DO GOVERNO FEDERAL

Tiago Wiedemann

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
tiagowrs@gmail.com

Francisco Assis Moreira do Nascimento

Professor Orientador
Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
assis@faccat.br

Resumo

Este artigo descreve o desenvolvimento e funcionalidades do portal da Transparência Participativa, que utiliza técnicas de *business intelligence* e colaboração para oferecer ao cidadão uma ferramenta útil na fiscalização e análise das contas públicas. Os *softwares* desenvolvidos extraem os dados de documentos de despesas diárias da página *web* do Portal da Transparência do Governo Federal e os armazenam em um *Data Warehouse* utilizando o conceito de modelo estrela proposto por Kimball. É apresentada uma interface *OLAP* que contempla funcionalidades específicas para a análise destes dados além de oferecer um ambiente de colaboração onde os cidadãos podem compartilhar suas análises entre si além de discutir as aplicações as quais o Governo Federal está destinando os recursos públicos. Por fim, são apresentadas sugestões para desenvolvimento de futuras funcionalidades.

Palavras-chave: e-Gov, Transparência Pública, Business Intelligence, Trabalho Colaborativo, Web Scraping.

Abstract

This paper presents the development process and main features of Transparência Participativa – a portal that adopts Business Intelligence tools and collaboration techniques in order to help people to analyze the brazilian government spendings. The developed softwares extract data from the web page of brazilian government's Portal da Transparência and put into a data warehouse, which was modeled using the star schema proposed by Kimball. It is also presented an OLAP web interface, which was developed to support the specific features of this kind of application and provides a virtual environment where people can discuss about the allocation and expenditure of the public resources by brazilian federal government.

Keywords: e-Gov, Government Transparency, Business Intelligence, Collaborative Work, Web Scraping.

1. Introdução

A civilização humana vive hoje em dia um momento único em sua história. Nunca antes a sociedade produziu tanta informação e esta nunca foi tão disseminada quanto nos dias atuais. As tecnologias para publicação, armazenamento e disseminação de informação evoluíram mais nos

Novembro de 2011, FACCAT, Taquara - RS

últimos 50 anos do que nos séculos anteriores. Além disto, pessoas, instituições, governos e empresas estão cada vez mais dependentes de informações para operarem eficientemente no mundo moderno.

Diversas tecnologias e processos foram criados para coleta de dados de diversas fontes como por exemplo, cartões de crédito, prestações de contas por meio eletrônico, documentos eletrônicos, dentre outras.

Outras tecnologias foram desenvolvidas com o intuito de disseminar e compartilhar estas informações, seja para pessoas através de páginas *web*, por exemplo, ou entre aplicativos através de *Web Services*.

Estes avanços porém trazem um problema. Como extrair conhecimento útil a partir de bases de dados gigantescas, de forma rápida e eficiente? A própria tecnologia se encarrega de criar respostas a esta pergunta. Através de técnicas de *Data Warehouse* e Mineração de Dados tornou-se possível automatizar em parte a busca por conhecimentos ocultos nas bases de dados disponíveis.

Um outro efeito deste novo contexto social é o de ter feito com que pessoas passem a buscar informações de forma mais ativa. E a relação do público com o governo foi uma das áreas influenciadas por este efeito.

Com a publicação de casos de corrupção, desvios de verbas públicas e ineficiências no controle de gastos, a população vêm se conscientizando de que deve fiscalizar de forma mais ativa as operações do governo. Neste contexto surgiu a ideia do Portal da Transparência, da Controladoria Geral da União – CGU, onde diariamente são publicados os dados referentes aos gastos do Governo Federal.

O presente trabalho têm como objetivo aperfeiçoar o serviço que o portal de transparência já presta, oferecendo à população técnicas mais sofisticadas de análise dos dados oferecidos. Além disso, visa criar uma comunidade virtual onde os cidadãos poderão compartilhar e discutir suas análises dos gastos públicos para, de forma colaborativa, aperfeiçoar o controle sobre estes gastos além de incentivar o espírito de cidadania e responsabilidade social.

2. Visão geral do Portal de Transparência do Governo Federal

O Portal da Transparência do Governo Federal foi criado pela Controladoria Geral da União (CGU), em novembro de 2004, com o objetivo de permitir ao cidadão acompanhar a maneira como o governo está gastando o dinheiro público.

2.1 Portal da Transparência

O Portal da Transparência está disponível para acesso via *web* através do endereço <http://www.portaltransparencia.gov.br> e contém dados diários e mensais sobre gastos diretos do

Novembro de 2011, FACCAT, Taquara - RS
governo, a transferência de recursos à estados e municípios, servidores públicos e também sobre o CEIS (Cadastro Nacional de Empresas Inidôneas e Suspensas). A Figura 1 a seguir mostra a página de entrada do portal.



Figura 1: Página inicial do portal da transparência

Fonte: <http://www.portaltransparencia.gov.br>

Como pode ser visto nesta figura, as consultas de despesas do Governo Federal são divididas em duas sessões, chamadas de informações diárias e informações mensais. Os dados extraídos para o *software* desenvolvido são aqueles exibidos na sessão informações diárias, que contém o detalhamento por documentos emitidos diariamente nos sistemas financeiros do Governo Federal.

2.2 Disposições legais a respeito da transparência governamental

A regulamentação dos gastos públicos é determinada através da Lei nº 4.320 de 17 de março de 1964 (PRESIDÊNCIA DA REPÚBLICA, 2011), que estabelece as três fases da despesa orçamentária:

- Empenho:** Regulamentado pelo artigo 58 da lei 4.320/64, é contabilizado através de um documento chamado Nota de Empenho no momento da contratação do serviço, aquisição de bem ou material de consumo ou amortização da dívida. Segundo informações do Portal da Transparência, o tipo do empenho pode ser **ordinário**, para despesas de valor fixo e previamente determinado, cujo pagamento ocorrerá apenas uma vez; **estimativo**, para

Novembro de 2011, FACCAT, Taquara - RS

despesas cujo montante não é possível determinar previamente, como por exemplo, água e energia elétrica; ou **global** para despesas contratuais ou de valor determinado, sujeitas a parcelamentos, como por exemplo aluguéis;

- b) **Liquidação:** Regulamentado pelo artigo 63 da lei 4.320/64 e processado pelas entidades executoras, ocorre no momento do recebimento do bem ou serviço. Os dois principais documentos envolvidos nesta fase são a Nota de Sistema e a Nota de Lançamento;
- c) **Pagamento:** Regulamentado pelo artigo 64 da lei 4.320/64 e processado pelas unidades gestoras executoras através da emissão principalmente do documento chamado Ordem Bancária e documentos relativos a retenção de tributos conforme o caso. Além deste, podem ser envolvidos os documentos DF (Documento de Arrecadação de Receitas Federais - DARF), DR (Documento de Arrecadação Financeira – DAR), GR (Guia de Recolhimento da União) e NL (Nota de Lançamento) em casos específicos.

O Governo Federal vem demonstrando preocupação crescente com a transparência dos gastos públicos e a responsabilidade na gestão fiscal como formas de combater a corrupção. Prova disto é a Lei Complementar nº 101 de 04 de maio de 2000, que estabelece as normas voltadas para a responsabilidade fiscal. O artigo 48 desta Lei determina que seja dada ampla divulgação, inclusive em meios eletrônicos de acesso público, aos instrumentos de transparência da gestão fiscal, como por exemplo as prestações de contas. A redação deste artigo foi alterada pela Lei Complementar nº 131 de 27 de maio de 2009, ficando sua redação atualmente em vigor:

Art. 48. São instrumentos de transparência da gestão fiscal, aos quais será dada ampla divulgação, inclusive em meios eletrônicos de acesso público: os planos, orçamentos e leis de diretrizes orçamentárias; as prestações de contas e o respectivo parecer prévio; o Relatório Resumido da Execução Orçamentária e o Relatório de Gestão Fiscal; e as versões simplificadas desses documentos.

Parágrafo Único. A transparência será assegurada também mediante:

I – incentivo à participação popular e realização de audiências públicas, durante os processos de elaboração e discussão dos planos, lei de diretrizes orçamentárias e orçamentos;

II – liberação ao pleno conhecimento e acompanhamento da sociedade, em tempo real, de informações pormenorizadas sobre a execução orçamentária e financeira, em meios eletrônicos de acesso público;

III – adoção de sistema integrado de administração financeira e controle, que atenda a padrão mínimo de qualidade estabelecido pelo Poder Executivo da União e ao disposto no art. 48-A.” (NR)

Especialmente em cumprimento ao inciso II, o Portal da Transparência passou a disponibilizar a partir de maio de 2010 os dados referentes ao detalhamento diário das despesas, que correspondem aos documentos emitidos diariamente nos sistemas integrados do Governo Federal relativos as três fases da despesa orçamentária determinadas pela lei 4.320/64.

2.3 Consulta de despesas diárias

Esta é a funcionalidade do portal da transparência que oferece a possibilidade de listar os documentos emitidos pelas unidades gestoras do Governo Federal. O usuário têm a opção de listar os documentos por período, fase, órgão superior, órgão ou entidade vinculada, unidade gestora, elemento da despesa e documento de identificação do favorecido. Através desta listagem é possível acompanhar diariamente para onde estão sendo destinados os recursos públicos.

É possível obter dados detalhados dos documentos clicando-se no número do mesmo. Os campos exibidos variam em função da fase da despesa a qual o documento se refere, mas cada documento possui pelo menos os seguintes campos: Número único de identificação; Tipo conforme definido pela legislação; Data de emissão; Órgão superior do poder executivo; Órgão ou entidade vinculada ao órgão superior; Unidade gestora responsável direta pela despesa; Identificação do gestor da unidade; Nome e documento de identificação do Favorecido e Valor em moeda corrente. A figura 2 a seguir exemplifica um documento típico da fase de empenho.

Detalhamento do documento: 2011NE000779

DADOS BÁSICOS			
Fase:	Empenho		
Documento:	2011NE000779	Tipo de Documento:	Nota de Empenho (NE)
Data:	30/09/2011		
Tipo de Empenho:	ESTIMATIVO	Espécie de Empenho:	Original
Órgão Superior:	25000 - MINISTERIO DA FAZENDA		
Órgão / Entidade Vinculada:	25101 - MINISTERIO DA FAZENDA		
Unidade Gestora Emitente:	170600 - COORD.GERAL DE CONTROLE DA DIVIDA PUBLICA		
Gestão:	00001 - TESOURO NACIONAL		
Favorecido:	17060000001 - COORD.GERAL DE CONTROLE DA DIVIDA PUBLICA		
Valor:	R\$ 10.000.000.000,00		
DADOS DETALHADOS			
Observação do Documento:	BACEN-JUROS-0152		
Esfera:	1 - ORÇAMENTO FISCAL	Tipo de Crédito:	A - INICIAL (LOA)
Fonte de Recursos:	52 - RESULTADO DO BACEN		
Unidade Orçamentária:	71101 - RECURSOS SOB SUPERVISAO DO MF - EFU		
Funcional Programática			
Função:	28 - ENCARGOS ESPECIAIS		
Subfunção:	843 - SERVICO DA DIVIDA INTERNA		
Programa:	0905 - OPERAÇÕES ESPECIAIS: SERVIÇO DA DÍVIDA INTERNA (JUROS E AMOR)		
Ação:	0455 - DIVIDA PUBLICA MOBILIARIA FEDERAL INTERNA		
Subtítulo (localizador):	0001 - DIVIDA PUBLICA MOBILIARIA FEDERAL INTERNA - NACIONAL		
Linguagem Cidadã:			

Figura 2: Exemplo parcial de uma Nota de Empenho (NE)

Fonte: <http://www.portaltransparencia.gov.br/despesasdiarias/empenho?documento=170600000012011NE000779>

Os campos exibidos na página demonstrada pela Figura 2 detalham o documento e permitem identificar onde, quando e como os recursos públicos foram aplicados em cada ato do poder executivo.

Novembro de 2011, FACCAT, Taquara - RS

O portal armazena atualmente um grande volume de informações históricas a respeito dos gastos federais. Segundo a página de Origem das Despesas do Portal da Transparência, disponível no endereço <http://www.portaltransparencia.gov.br/sobre/OrigemDespesas.asp>, em 30 de setembro de 2011 o portal armazenava 23.012.144 (vinte e três milhões doze mil cento e quarenta e quatro) documentos somente na parte de despesas diárias.

Este grande volume de documentos acaba por dificultar a análise, pois torna-se inviável analisar cada documento individualmente sem ter uma visão global das despesas. Por este motivo, o portal disponibiliza as informações mensais, que contém dados sumarizados a partir de 2004, permitindo assim que as despesas de cada exercício sejam analisadas de forma sintética. Por exemplo, ao selecionar o exercício de 2010 e fazer uma busca “por Ação”, o usuário obtém uma lista com os valores gastos por ação de governo no exercício de 2010. Esta consulta está demonstrada na Figura 3 a seguir.

GASTOS DIRETOS POR AÇÃO GOVERNAMENTAL

baixar mais dados  imprimir 

Total destinado pelo Governo Federal em âmbito nacional em 2010 - Aplicações Diretas: R\$ **1.045.039.943.071,43**

Selecione o(a) "Ação Governamental" para obter o detalhamento do valor Caso queira outra classificação, clique no título da coluna correspondente.

Função	Ação Governamental	Linguagem Cidadã	Total no Ano (R\$)
Administração2003	Ações de Informática	Tecnologia da Informação	40.392.979,84
Administração2B13	Ações de Prevenção à Corrupção e Transparência Governamental		3.445.289,88
Administração2000	Administração da Unidade	Administração de unidade	449.479.775,28
Administração2069	Análise dos Reajustes de Preços e Tarifas de Serviços Públicos		40.369,93
Administração10HQ	Aperfeiçoamento do Sistema de Suporte à Gestão		1.859.600,00
Administração10UW	Apoio à Modernização do Sistema de Controle Externo dos Estados, Distrito Federal e Municípios Brasileiros - PROMOEX		260.824,12
Administração005E	Apoio a Projetos de Desenvolvimento Sustentável Local Integrado		786.229,67
Administração8236	Apoio a Projetos de Interesse do Sistema Nacional de Políticas sobre Drogas		3.157.551,37
Administração4914	Apoio ao Desenvolvimento de Sistemas em Software Aberto		145.014,70
Administração2C11	Apoio Técnico e Administrativo à Equipe de Transição de Governo		76.628,94
Administração2244	Apuração, Inscrição e Execução da Dívida Ativa da União		20.722.377,56
Administração2A37	Arrecadação e Cobrança Administrativa de Créditos Patrimoniais		1.627.302,16
Administração2238	Arrecadação Tributária e Aduaneira		81.580.858,14
Administração20CW	Assistência Médica aos Servidores e Empregados - Exames Periódicos		631.796,86
Administração2004	Assistência Médica e Odontológica aos Servidores, Empregados e seus Dependentes	Assistência Médica e Odontológica	9.465.902,57

Figura 3: Exemplo de consulta mensal no Portal da Transparência: Gastos por ação de governo no exercício de 2010

Fonte: Portal da Transparência (2011)

Ainda na consulta demonstrada na Figura 3, ao se clicar em “Ações de Informática”, o usuário recebe uma lista com os favorecidos no período de 2010, referentes a ação clicada, podendo ordenar por nome ou valor da despesa.

A sequência das consultas é determinada pelo portal, não havendo até a data do presente artigo nenhuma funcionalidade que permitisse cruzar livremente os dados das despesas e/ou compartilhar estas análises com outros usuários de forma a incentivar a discussão a respeito das contas públicas.

3. Transparência Participativa

O software – Transparência Participativa - desenvolvido no contexto do presente trabalho une técnicas e ferramentas de *Business Intelligence* com os conceitos de colaboração em ambiente *web* para criar um ambiente de análise multi-dimensional das despesas governamentais.

3.1 Modelo ER do Data Warehouse

Segundo Kimball (1998), um *Data Warehouse* é um banco de dados relacional que é modelado para facilitar as consultas, permitindo ao usuário visualizar uma grande massa de dados sob diferentes ângulos. Para atingir este objetivo, Kimball propõe um modelo de dados chamado de modelo estrela, onde os valores mensuráveis de um determinado domínio são armazenados em uma tabela chamada de **fato**, que por sua vez faz referência a outras tabelas que armazenam as **dimensões** que descrevem o fato em diversos ângulos.

Seguindo este conceito, os fatos e dimensões do Portal da Transparência podem ser extraídos a partir dos respectivos documentos. Uma vez que toda despesa do governo deve ser contabilizada em um documento, pode-se criar o modelo estrela, considerando-se o valor de cada documento como sendo um fato, e os seus demais campos como sendo dimensões. A Figura 4 abaixo apresenta o diagrama ER do modelo criado.

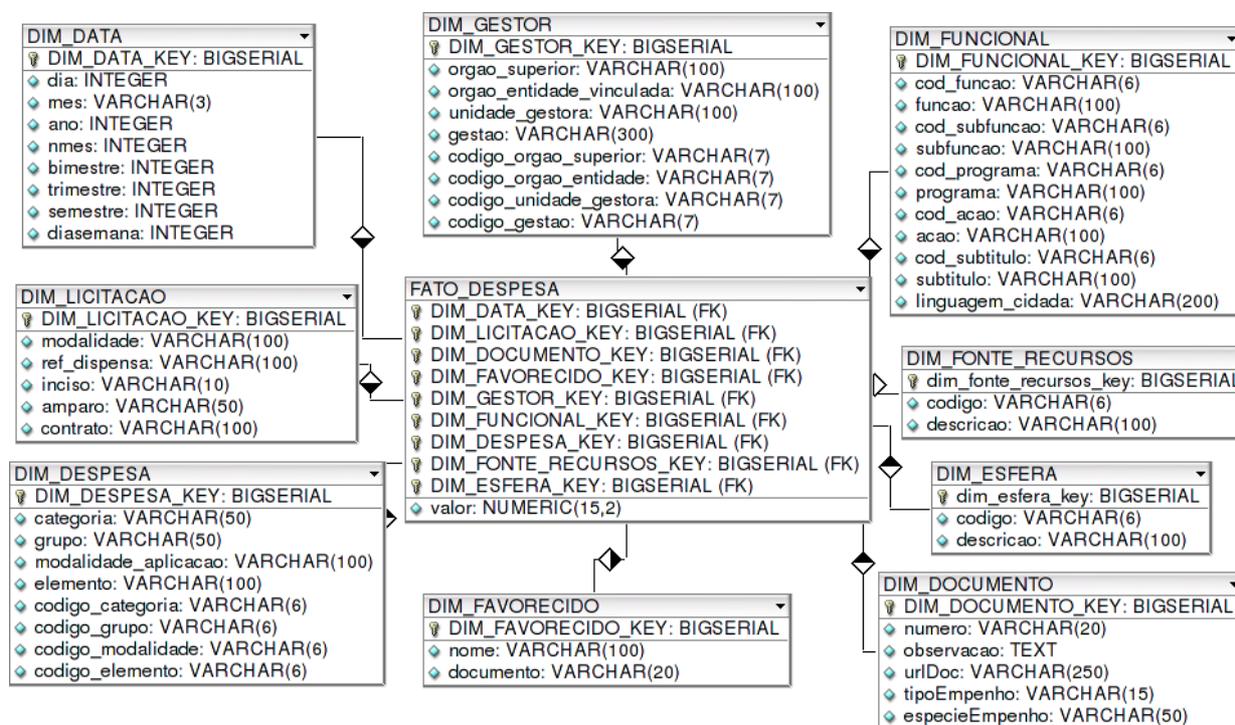


Figura 4: Modelo do Data Warehouse do Portal da Transparência Participativa

Conforme a Figura 4 demonstra, o *Data Warehouse* desenvolvido segue o modelo estrela proposto por Kimball, onde a tabela de fato é preenchida com o valor em moeda corrente de cada documento e as tabelas de dimensões são preenchidas com os demais campos.

Novembro de 2011, FACCAT, Taquara - RS

É importante ressaltar que a dimensão **DIM_DOCUMENTO** possui um tratamento especial na interface *web*. Esta dimensão mantém, na prática, uma relação 1 para 1 com a tabela de fatos, pois contém dados que são únicas por documento como por exemplo, o número de identificação, endereço *web* e observações específicas da operação. Por ter estas características, ela não é exibida na lista de dimensões disponíveis para cruzamentos para o usuário, mas pode ser consultada na tela de análise detalhada.

3.2 Interface Web

Com o banco de dados do *Data Warehouse* populado, pode-se criar uma interface com o usuário de forma a implementar as funcionalidades de uma ferramenta OLAP (*Online Analytical Processing*), que segundo Anzanello (2011) corresponde às aplicações que fornecem solução para a síntese, análise e consolidação de dados. Segundo a autora, uma ferramenta OLAP deve ser capaz de recuperar informações de um *Data Warehouse* na forma de **cubos** contendo fatos e suas respectivas dimensões e permitindo ao usuário a criação de consultas *ad-hoc*, alterando o ângulo de visão dos dados do cubo conforme desejar.

O software Transparência Participativa, desenvolvido no presente trabalho procurou implementar estes conceitos em uma interface *web*, a fim de tornar possível a qualquer cidadão, através da internet, realizar consultas sobre os dados obtidos do Portal da Transparência visando obter um conhecimento mais apurado da forma como o governo está aplicando os recursos públicos.

A interface OLAP constitui-se de uma aplicação Java rodando em servidor J2EE que fornece acesso para consulta ao *Data Warehouse*. Esta interface permite ao usuário selecionar quais as dimensões que deseja analisar e exibe os resultados de forma tabular, totalizando os valores das despesas dentro das dimensões selecionadas e permitindo o acréscimo de novas dimensões, o que leva a criação de um ambiente de consultas dinâmicas que facilitam a descoberta de conhecimento de forma interativa pelo usuário.

A Figura 5 mostra a tela principal de consulta do software desenvolvido. Para esta versão do software optou-se por limitar a extração de dados à etapa de empenho, pois os documentos desta etapa possuem mais campos, o que permite o detalhamento mais apurado das despesas.

A lista a esquerda na Figura 5 exibe todas as dimensões disponíveis no *Data Warehouse*, permitindo que o usuário analise as despesas governamentais sob diferentes ângulos. A tabela gerada identifica através de cores, o maior e o menor valor em cada linha ou coluna.

Colunas: Trimestre
 Linhas: Modalidade
 Filtro:
 MDX Nova Tabela

Identificar maior e menor valor: Em cada coluna

Modalidade	2010/T2	2010/T3	2010/T4	2011/T1
CONCORRENCIA	1,054,995,141.44	4,806,617,877.07	5,925,099,112.82	2,621,130,896.36
CONCURSO	858,632.78	4,561,578.40	3,725,590.06	545,564.95
CONSULTA	19,863,587.89	41,886,933.87	54,400,459.67	14,834,251.26
CONVITE	22,867,136.00	59,758,554.58	162,029,825.84	14,244,357.58
DISPENSA DE LICITACAO	728,821,971.99	1,674,163,040.71	3,076,301,546.21	1,553,639,463.69
INEXIGIVEL	896,571,914.51	2,067,934,250.60	1,949,234,780.44	4,205,588,010.79
NAO SE APLICA	214,052,415,352.56	78,322,081,931.63	41,616,612,918.23	1,162,292,554,231.24
PREGAO	1,471,547,266.64	3,705,195,870.45	6,239,631,307.96	3,247,587,984.61
SUPRIMENTO DE FUNDO	16,666,438.24	48,327,609.21	35,060,223.81	30,107,570.68
TOMADA DE PRECO	64,961,112.75	161,693,705.24	382,908,810.61	66,269,435.54

Tempo: 4ms

Figura 5: Exemplo de consulta utilizando a interface OLAP: Total de despesas por trimestre e modalidade de licitação

O software oferece a funcionalidade de marcar uma célula da tabela para análise detalhada em dois níveis: **Lista de documentos**, que consiste na listagem completa de todos os fatos que foram totalizados naquela célula, e **lista de favorecidos**, onde são listadas todas as empresas, instituições ou pessoas físicas que receberam os recursos federais.

3.3 Ambiente Colaborativo

O portal Transparência Participativa visa criar um ambiente colaborativo de descoberta de conhecimento. Por este motivo, todas as análises, tabelas e gráficos podem ser salvos pelo usuário para consulta futura e compartilhamento com os demais. O software fornece o recurso de fórum de discussão, onde podem ser trocadas informações e discutidas as análises feitas entre os usuários. Este recurso visa incentivar a análise colaborativa dos gastos públicos, possivelmente trazendo à tona questões que de outra forma ficariam ocultas devido ao grande volume de dados que são gerados diariamente pelo Governo Federal, no que se refere a suas despesas.

4. Extração de dados do Portal da Transparência

Segundo KIMBALL (1998), uma das etapas da criação de um *Data Warehouse* consiste em extrair os dados de uma ou mais fontes, transformá-los para que se tornem consistentes, e então carregá-los no banco de dados. Este processo é chamado de ETL (*Extract, Transform, Load*).

Devido aos dados originais do portal da transparência estarem disponibilizados somente através da página web, torna-se necessário o uso de uma ferramenta de *screen scraping*. Segundo SCHRENK (2007), este tipo ferramenta faz o *download* de uma determinada página e então efetua o *parser* do conteúdo HTML a fim de extrair os dados desejados. A ferramenta desenvolvida para

Novembro de 2011, FACCAT, Taquara - RS

esta etapa do software acumula a função de *web scraper* e de extração, transformação e carga dos dados no *Data Warehouse*.

Cabe ressaltar que o conjunto de soluções Pentaho (PENTAHO, 2011) oferece uma ferramenta para a criação de processos de ETL chamada *Pentaho Spoon*. Esta é capaz de ler e gravar em diversas fontes de dados, além de oferecer diversas funcionalidades de transformação dos dados. Porém na versão avaliada, não havia uma transformação específica para utilizá-la como um *scraper* de páginas *web*. Por este motivo optou-se então pelo desenvolvimento de um aplicativo específico para esta tarefa.

4.1 TransparenciaETL

O processo de importação dos dados do Portal da Transparência para o *Data Warehouse* foi implementado em um aplicativo escrito em linguagem Java chamado de **TransparenciaETL**. Como já mencionado, para a fase de extração dos dados o aplicativo implementa a técnica de *web scraper*, ao acessar diretamente a página de despesas diárias do Portal da Transparência, disponível no endereço <http://www.portaltransparencia.gov.br/despesasdiarias>, através de uma requisição *HTTP* e processar o conteúdo *HTML* da página de cada documento. Com isto são extraídos os dados que serão armazenados no *Data Warehouse* pelas fases de transformação e carga implementadas pelo mesmo aplicativo.

No desenvolvimento deste aplicativo procurou-se garantir o reaproveitamento do código desenvolvido. A fim de garantir este reaproveitamento, as partes comuns, interfaces, classes abstratas e também as funcionalidades básicas como a conexão ao servidor foram implementadas em uma biblioteca separada, chamada de *EtlAPI*. Espera-se que futuros projetos de *Business Intelligence* possam se beneficiar do uso desta biblioteca, agilizando o seu desenvolvimento.

4.2 Processo de extração dos dados do Portal da Transparência

O aplicativo desenvolvido simula a navegação nas páginas do Portal da Transparência, enviando os requisições *HTTP-GET*, de forma a obter os dados dos documentos exibidos. O processo de extração é composto das seguintes etapas:

- a) O aplicativo deve ser chamado via linha de comando ou agendamento do sistema operacional, passando-se por parâmetro a data de início e fim do período a ser extraído;
- b) Para cada dia do período solicitado, o aplicativo executa a requisição *HTTP-GET* que aconteceria caso o usuário acessasse manualmente a página utilizando um navegador *web* e clicasse no botão consultar, informando o dia solicitado. Uma requisição *HTTP-GET* é feita acessando-se um endereço, chamado de URL, onde os parâmetros da consulta são

Novembro de 2011, FACCAT, Taquara - RS

acrescentados ao final do endereço acessado, utilizando-se o caracter “?”. A Figura 6 mostra uma consulta sendo feita através do navegador web.

Consulta Rápida [Consulta Avançada](#) | [Consulta por Documento](#)

Período: 13/10/2011 a 13/10/2011 Formato: dd/mm/aaaa

Fase da Despesa: Empenho Liquidação Pagamento

Órgão Superior: Todos (período de 1 dia ou favorecido específico)

Favorecido: Fornecer CNPJ, CPF, UG-Gestão ou outros (sem pontuações)

Figura 6: Opções de consulta no Portal da Transparência que são simuladas pelo aplicativo.

A consulta mostrada na Figura 6, corresponde a uma requisição *HTTP-GET* feita para o seguinte endereço:

[http://www.portaltransparencia.gov.br/despesasdiarias/resultado?
consulta=rapida&periodoInicio=13%2F10%2F2011&periodoFim=13%2F10%2F2011&fase
=EMP&codigoOS=TOD&codigoFavorecido=](http://www.portaltransparencia.gov.br/despesasdiarias/resultado?consulta=rapida&periodoInicio=13%2F10%2F2011&periodoFim=13%2F10%2F2011&fase=EMP&codigoOS=TOD&codigoFavorecido=)

Com esta consulta, o aplicativo obtém a lista de documentos registrados no portal, no dia;

- c) Como o Portal da Transparência implementa a paginação dos resultados, a lista de documentos está agrupada em páginas de 15 documentos. O aplicativo navega entre elas para obter de forma sequencial todos os documentos listados, e em seguida passa para a próxima página. Para acessar o documento, o aplicativo processa o *HTML* da lista de documentos e extrai o endereço de cada um destes. É possível identificar estes endereços por que o código *HTML* que os armazena segue um padrão facilmente identificável:

```
<a href="empenho?documento=170502000012011NE811997">2011NE811997</a> ;
```

- d) Para cada documento encontrado, o aplicativo efetua uma requisição *HTTP-GET* para o endereço que corresponde à página onde são exibidos os seus dados;
- e) Por fim, o aplicativo processa o *HTML* da página do documento, cria um registro na tabela de fato com o valor deste e preenche as dimensões conforme os demais campos.

Este processo é implementado em duas classes principais. A primeira é chamada de **TransparenciaSource** e é responsável por realizar as requisições *HTTP-GET*, obter o *HTML* de cada documento e então processá-lo utilizando os dados extraídos para gerar os objetos que mapeiam o banco de dados do *Data Warehouse*. A segunda é chamada de **TransparenciaLoader** e é responsável por receber os objetos gerados pela classe **TransparenciaSource** e gravá-los no banco de dados.

Na classe **TransparenciaSource** a conexão com servidor é feita utilizando-se a classe **URLConnection**, do pacote **java.net**. O quadro 1 mostra o código fonte que implementa o processo de obtenção do conteúdo de uma página.

Novembro de 2011, FACCAT, Taquara - RS

Caso ocorra alguma falha na conexão de rede com o servidor o Portal da Transparência, o aplicativo aguarda por 2 minutos antes de tentar novamente. Este processo é repetido indefinidamente até que o aplicativo consiga estabelecer novamente a conexão com o servidor e extrair o conteúdo HTML da página. Tal procedimento é necessário para que o aplicativo consiga lidar com eventuais indisponibilidades que podem ser causadas tanto por problemas na conexão com a internet quanto por problemas no próprio servidor que hospeda o portal da transparência.

```
URL url = new URL(endereco);
int tentativas = 0;
while (!ok) {
    try {
        URLConnection conn = url.openConnection();
        conn.setRequestProperty("User-Agent", "Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6 GTB6");
        conn.setRequestProperty("Accept-Language", "pt-br,pt;q=0.8,en-us;q=0.5,en;q=0.3");
        conn.setReadTimeout(URL_CONNECTION_TIMEOUT);
        conn.connect();
        isr = new InputStreamReader(conn.getInputStream(), "UTF-8");
        html = new StringBuilder();
        if (isr != null) {
            in = new BufferedReader(isr);
            String linha;
            while ((linha = in.readLine()) != null) {
                html.append(linha);
            }
        }
        ok = true;
    } catch (SocketTimeoutException timeout) {
        tentativas = tratarTimeout(tentativas, timeout, endereco);
    } catch (UnknownHostException unk) {
        tentativas = tratarTimeout(tentativas, unk, endereco);
    }
}
```

Quadro 1: Trecho de código que obtém o conteúdo HTML de uma página

A parte principal do código demonstrado no Quadro 1 é executada pela classe `URLConnection` que executa uma conexão *HTTP* cujos valores de cabeçalhos são definidos pelos métodos `setRequestProperty()`, e cuja resposta é lida pelo objeto da classe `InputStreamReader`.

4.3 Processamento do conteúdo HTML do Portal da Transparência

O HTML recebido é então processado através de uma classe *parser* que foi implementada para cada página que o aplicativo deve processar. Se estiver sendo lida a tabela de documentos de um determinado dia, o aplicativo utiliza a classe `ParserDespesas`, que basicamente procura pelas *tags* que contenham o endereço de um documento, seguindo o padrão mencionado anteriormente e retorna uma lista de endereços de documentos a serem processados. Ao obter o *HTML* de cada

Novembro de 2011, FACCAT, Taquara - RS

documento da lista retornada, o aplicativo utiliza a classe **ParserDocumento** que extrai os dados do *HTML* e os prepara para serem salvos no *Data Warehouse*.

Para encapsular o processo de extração, do processo de gravação dos dados, foram implementadas classes que mapeiam a estrutura do banco de dados em objetos. A Figura 7 mostra o diagrama destas classes.

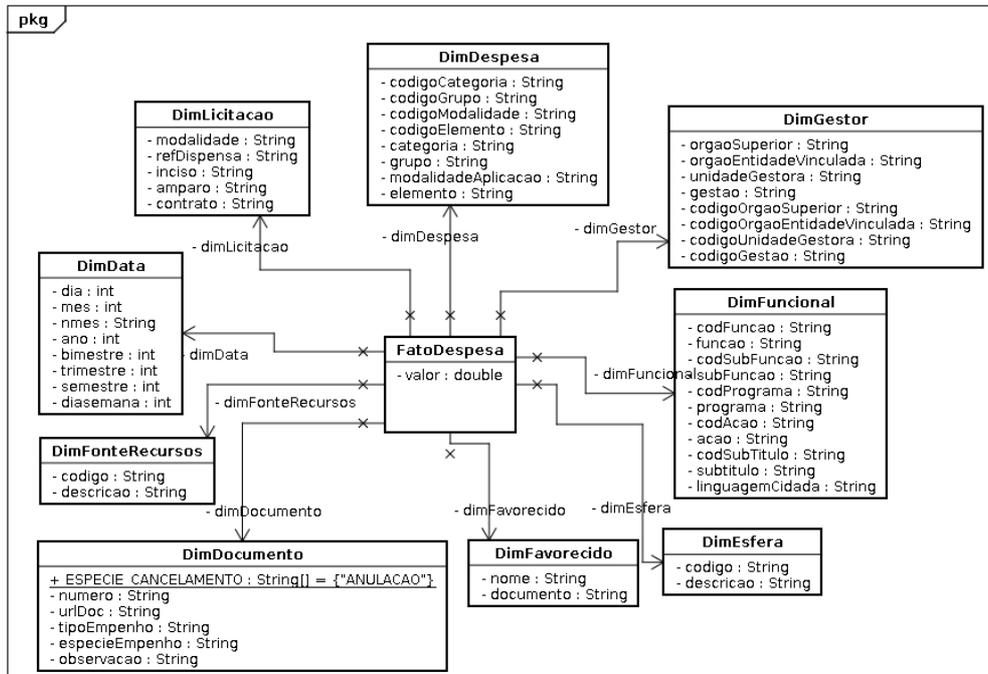


Figura 7: Mapeamento Objeto Relacional do aplicativo de extração

As classes mostradas na Figura 7 correspondem exatamente às mesmas tabelas mostradas no diagrama ER da Figura 4.

O principal método da classe `ParserDocumento` chama-se `extrairFato(String htmlDoc, String urlDoc)`, e é responsável por gerar um objeto da classe `FatoDespesa`, que contém objetos das classes de dimensão preenchidos com os valores extraídos a partir do *HTML*. A implementação deste método percorre todas as *tags* do *HTML* do documento e identifica aquelas que contenham os dados que correspondam aos atributos destas classes. Assim como o *HTML* da listagem, as *tags* dos documentos seguem um padrão que é facilmente identificável, conforme mostrado no quadro 2:

```

<tr class="impar">
  <td class="rotulo">Órgão Superior:</td>
  <td colspan="3">52000 - MINISTERIO DA DEFESA</td>
</tr>

```

Quadro 2: Trecho de HTML da página de um documento

Novembro de 2011, FACCAT, Taquara - RS

O trecho de HTML mostrado no quadro 2 corresponde ao nome do órgão superior responsável pela emissão do documento, que por sua vez está mapeado no atributo `orgaoSuperior` da classe `DimGestor`.

Para o campo 'Valor' que armazena o total da despesa a qual o documento se refere, é necessário um tratamento especial visto que alguns documentos referem-se a despesas realizadas no exterior. Portanto seus valores estão em dólar (US\$) e não em reais (R\$). Como os valores no *Data Warehouse* devem ser uniformizados, é necessário que a classe capture apenas o valor em Reais, e ignore o valor em Dólares. Isto é possível pois nos casos citados, a página do portal da transparência adota o rótulo 'Valor convertido' ao invés de 'Valor', para representar o valor do documento já convertido para moeda nacional.

Após a transformação dos dados em um objeto mapeado, o aplicativo utiliza a classe `TransparenciaLoader` para processar os objetos gerados pela classe `TransparenciaSource` e então armazená-los no *Data Warehouse*.

4.4 Persistência dos dados extraídos

Os dados armazenados nos objetos de mapeamento são persistidos através de comandos SQL enviados ao banco de dados via *JDBC*. Porém, antes de armazenar qualquer informação, a classe `TransparenciaLoader` verifica se tratam-se de novos dados ou de uma atualização de dados já existentes.

Por exemplo, o primeiro fato processado em um dia certamente fará com que a classe identifique os dados do objeto `DimData` como sendo um novo registro, o que provocará uma operação de *INSERT* no banco de dados. A partir do segundo fato identificado no mesmo dia, a classe identificará que os dados contidos no objeto `DimData` já estão no banco e portanto nenhuma operação adicional é necessária nesta tabela, situação em que o objeto assume o valor da chave primária armazenado.

A verificação se um registro de dimensão é novo é feita comparando-se a igualdade de todos os campos armazenados na tabela com os dados que foram extraídos da página. A classe também verifica se o fato lido é um fato novo ou uma releitura de um fato já processado considerando, neste caso, o número do documento armazenado na classe `DimDocumento`.

O Quadro 3 mostra o código fonte do método da classe `TransparenciaLoader` responsável por armazenar os dados da classe `FatoDespesa` no *Data Warehouse*.

```
protected IFact importFact(Connection conn, IFact fact) throws Exception {
    fact = processDimensions(conn, fact);
    if (isNewFact(conn, fact)) {
        doUpdate(conn, fact.getInsertSQL());
        Long id = getPk(conn, fact.getSelectPkSQL());
        if (id == null || id == 0) {
            throw new Exception("EtlAPI - INSERT FAILED: [" + fact.getInsertSQL() + "]);
        }
    } else {
        int test = doUpdate(conn, fact.getUpdateSQL());
        if (test != 1) {
            throw new Exception("EtlAPI-UPDATE FAILED: ["+fact.getUpdateSQL()+"] result = " + test);
        }
        Long id = getPk(conn, fact.getSelectPkSQL());
        if (id == null || id == 0) {
            throw new Exception("EtlAPI-UPDATE FAILED: ["+fact.getUpdateSQL()+"]");
        }
    }
    return fact;
}
```

Quadro 3: Trecho de código fonte que implementa a gravação de fatos no *Data Warehouse*

No Quadro 3, o método **importFact** recebe um objeto da interface **IFact**. Esta interface define os métodos **getInsertSQL()** e **getUpdateSQL()**, que são implementados pela classe **FatoDespesa** conforme definida na Figura 7. Ambos os métodos retornam os comandos SQL prontos para serem executados, já com os valores atuais dos atributos do objeto.

O Quadro 4 mostra o método responsável por importar os dados de dimensões.

```
protected IDimension importDimension(Connection conn, IDimension dim) throws Exception {
    if (dim.getId() <= 0) {
        Long id = getPk(conn, dim.getSelectPkSQL());
        if (id == null || id == 0) {
            doUpdate(conn, dim.getInsertSQL());
            id = getPk(conn, dim.getSelectPkSQL());
            if (id == null || id == 0) {
                throw new Exception("EtlAPI - INSERT FAILED: " + dim.getInsertSQL());
            } else {
                dim.setId(id);
            }
        } else {
            dim.setId(id);
        }
    }
    return dim;
}
```

Quadro 4: Trecho de código demonstrando o processo de gravação de dimensões no *Data Warehouse*

No Quadro 4, método recebe um objeto da interface **IDimension**, que é implementada por todas as classes que mapeiam as tabelas de dimensão, e determina que estas classes implementem o método **getInsertSQL()** que retorna o comando SQL responsável por realizar a inserção de dados nas respectivas tabelas. Este método somente é chamado caso o objeto contenha um novo registro.

O Portal da Transparência armazena uma média aproximada de 15.000 documentos por dia. O aplicativo leva cerca de 2 horas para processar um dia completo, porém este valor pode variar em função da velocidade do link e ocupação da banda além da carga e disponibilidade do servidor do Portal da Transparência.

5. Ferramentas e técnicas utilizadas no desenvolvimento da interface *web*

O pacote de soluções *open source* para *business intelligence* Pentaho Solutions (2011) oferece uma interface gráfica que implementa as funcionalidades OLAP. No desenvolvimento deste software optou-se pelo desenvolvimento de uma interface própria, pois a ferramenta do Pentaho não oferece por exemplo, uma funcionalidade para se criar grupos de discussão a respeito dos dados analisados, que é um dos objetivos deste trabalho.

Outro aspecto a se considerar é que a própria interface de consulta do Pentaho mostrou-se pouco amigável, pois como sua proposta é a de ser uma interface genérica, ela inicia exibindo uma tabela com todas as dimensões, o que pode se tornar bastante confuso para o usuário final.

5.1 Linguagem de desenvolvimento

Como já mencionado, o software foi desenvolvido em Java e roda em um servidor de aplicações J2EE que implementa as funcionalidades básicas de servidor web. O software desenvolvido é um aplicativo web simples, que não faz uso de EJB (ORACLE, 2011).

Para a interface com o usuário foi utilizada a biblioteca ICE Faces, que implementa o *framework Java Server Faces (JSF)*. Este framework utiliza o conceito de componentes para facilitar o desenvolvimento de aplicações com interfaces ricas.

5.2 API Mondrian

Para comunicação com o WebServices foi utilizada a *API Mondrian*, que integra o pacote Pentaho. Esta API é responsável pela implementação de *queries MDX* que significa *Multi Dimensional eXtensions* (PASUMANSKY, 1998). Trata-se é uma linguagem de consulta a bancos de dados multi-dimensionais, desenvolvida pela Microsoft para seu banco de dados SQL Server. A API Mondrian implementa esta linguagem em Java, expandindo sua utilização para outros bancos de dados. A estrutura da linguagem é similar ao SQL padrão, porém implementa a ideia de consulta em cubos. Sua sintaxe básica é esta:

```
SELECT axis1 ON COLUMNS, axis2 ON ROWS FROM cube
```

A consulta é executada sempre sobre um cubo e não sobre uma tabela como na linguagem SQL padrão.

O cubo deve ser definido em um documento XML que mapeia a tabela de fato e as tabelas de dimensões que o compõem. A API Mondrian chama este documento de *Schema* e pode ser escrito utilizando o aplicativo *Schema Workbench* distribuído em conjunto com o Pentaho. Sua interface pode ser vista na Figura 8.

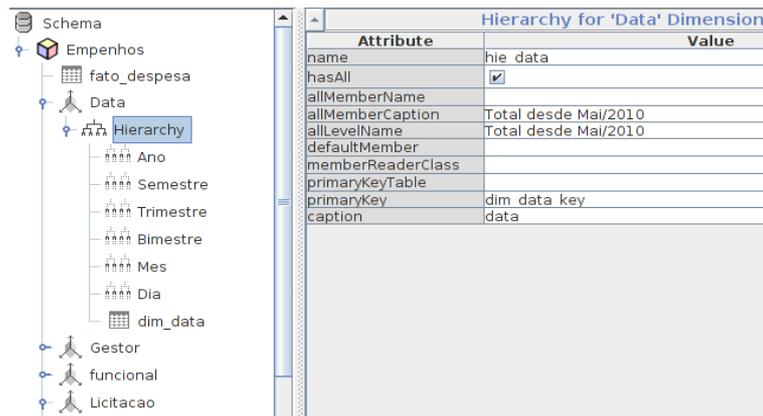


Figura 8: Definição do *schema* utilizado no Portal da Transparencia Participativa

A estrutura do *schema* define para cada dimensão uma hierarquia que agrupa os campos de uma determinada tabela na ordem em que devem ser lidos. Por exemplo, na Figura 8 a hierarquia da dimensão “Data” utiliza os campos ano, semestre, trimestre, bimestre, mês e dia, da tabela DIM_DATA, nesta ordem. Esta ordem deve ser considerada no momento em que se monta a *query* MDX.

Uma *query* MDX retorna um objeto da classe **mondrian.olap.Result**, que mapeia os dados da forma em que devem ser visualizados pelo usuário. A API chama as dimensões do cubo visualizado de *AXES*, e estabelece um código numérico para cada dimensão. Assim, as colunas correspondem ao *axis 0* e as linhas ao *axis 1*. A API suporta até 128 *axes* (MICROSOFT, 2011b), porém como a maioria das visualizações feitas são de forma tabular isto significa que, na maioria dos casos somente as dimensões 0 e 1 são utilizadas. Deve-se tomar o cuidado de não confundir as dimensões de visualização, que são os *axes*, com as dimensões físicas estabelecidas pelo modelo de dados, no *Data Warehouse*.

Uma consulta MDX deve definir quais as dimensões do banco de dados irão compor as dimensões de visualização, nos respectivos *axes*, conforme mostra o Quadro 5.

```
SELECT
  NON EMPTY {
    [Data.hie_data].[All Data.hie_datas].[2010].[1].[2],
    [Data.hie_data].[All Data.hie_datas].[2010].[2].[3],
    [Data.hie_data].[All Data.hie_datas].[2010].[2].[4]
  } ON COLUMNS
, NON EMPTY {
  [Licitacao.hie_licitacao].[All Licitacao.hie_licitacoes].[CONCORRENCIA],
  [Licitacao.hie_licitacao].[All Licitacao.hie_licitacoes].[CONCURSO],
  [Licitacao.hie_licitacao].[All Licitacao.hie_licitacoes].[CONSULTA],
  [Licitacao.hie_licitacao].[All Licitacao.hie_licitacoes].[CONVITE]
} ON ROWS
FROM [Empenhos]
```

Quadro 5: Exemplo de consulta MDX usando as palavras-chave ON ROWS e ON COLLUMNS para representar os *axes* e utilizando as definições de *schema* demonstradas na figura 8.

Novembro de 2011, FACCAT, Taquara - RS

Através do método **Result.getAxes()** é retornado um vetor com as dimensões do cubo a serem visualizadas. Para a *query* especificada no Quadro 5 seria retornado um vetor de 2 posições, já que foram utilizados os *axes* COLUMNS e ROWS. Neste exemplo foram usadas as palavras chave ON COLUMNS e ON ROWS, mas os *axes* também podem ser especificados por números:

```
SELECT axis1 ON 0, axis2 ON 1, axis3 ON 2, ..., axisn ON n FROM cube
```

Cada objeto Axis no objeto Result contém uma lista de objetos da classe **Position**, do pacote mondrian.olap, correspondentes a cada posição a ser visualizada no *axis*. Por exemplo, cada linha na tabela é representada por um objeto Position no *axis 1*, enquanto que cada coluna da tabela é representada por um objeto Position no *axis 0*. Cada objeto Position pode ter 1 ou mais objetos da classe **Member**, também do pacote mondrian.olap, que as dimensões selecionadas em cada posição.

Para exemplificar, vamos considerar o acréscimo da dimensão órgão superior às linhas da tabela mostrada na Figura 5, fazendo com que as despesas de cada modalidade de licitação sejam desmembradas por ministério. Nesta situação, cada objeto Position do *axis 1*, conterà dois objetos Member, o primeiro referenciando a modalidade de licitação e o segundo o ministério. A Figura 9 mostra a visualização parcial da tabela gerada por esta consulta.

Modalidade	Órgão Superior	2010/T2	2010/T3	2010/T4	2011/T1
CONCORRENCIA	MINISTERIO DA CIENCIA E TECNOLOGIA	15,424,981.53	44,239,180.97	54,773,860.20	24,808,509.80
	MINISTERIO DA DEFESA	167,603,334.62	270,935,232.88	597,207,319.80	225,144,284.85
	MINISTERIO DA EDUCACAO	138,743,415.53	429,847,966.89	659,867,609.57	236,361,172.37
CONCURSO	MINISTERIO DA CIENCIA E TECNOLOGIA			15,000.00	127,894.21
	MINISTERIO DA DEFESA	3,700.00	92,036.04	337,155.26	119,079.32
	MINISTERIO DA EDUCACAO	7,466.09	33,000.00	315,000.00	
CONSULTA	MINISTERIO DA DEFESA			31.44	
CONVITE	MINISTERIO DA CIENCIA E TECNOLOGIA	138,741.31	345,869.78	2,029,042.82	83,854.51
	MINISTERIO DA DEFESA	16,348,183.51	44,759,426.33	131,992,087.05	7,004,470.50
	MINISTERIO DA EDUCACAO	3,246,169.91	7,270,887.97	9,477,835.09	3,508,272.19

Figura 9: Exemplo de consulta com duas dimensões nas linhas da tabela

Considerando-se os dados exibidos na Figura 9, para obter o objeto Member referente ao órgão superior chamado “Ministério da Educação” na modalidade chamada “Concurso” utiliza-se o código mostrado no Quadro 6:

```
public Member getMembroExemplo(mondrian.olap.Result result) {  
    mondrian.olap.Axis[] axis = result.getAxes();  
    List<mondrian.olap.Position> positions = axis[1].getPositions();  
    mondrian.olap.Position sextaLinha = positions.get(5);  
    mondrian.olap.Position segundoMembro = sextaLinha.get(1);  
    return segundoMembro;  
}
```

Quadro 6: Exemplo de código fonte que ilustra o processo de leitura das linhas e colunas de um objeto mondrian.olap.Result

O código do Quadro 6 inicia lendo a lista de objetos da classe Position do *axis* 1, que corresponde as linhas da tabela. Em seguida obtém o objeto que está na posição 5 da lista. Como a lista inicia em 0 (zero), o 5º elemento corresponde a 6ª linha. Como nesta consulta, foram usadas 2 dimensões para as linhas, este objeto contém dois objetos Member. O primeiro, na posição 0 (zero), aponta para o valor “Concurso” da dimensão modalidade, e o segundo, na posição 1, aponta para o valor “Ministério da Educação” na dimensão órgão superior.

Por fim, para a leitura das células com os valores a serem exibidos, deve ser utilizado o método **Result.getCell(int[])**. Este método recebe as coordenadas da célula. Como neste *software* toda a visualização é feita em uma tabela, o *array* a ser passado deverá ter 2 posições. O Quadro 7 exemplifica o código fonte necessário para a leitura das células da tabela de um objeto Result.

```

for (int row = 0; row < rowAxis.getPositions().size(); row++) {
    for (int col = 0; col < colAxis.getPositions().size(); col++) {
        System.out.print(result.getCell(new int[2]{col, row}).getFormattedValue() + " | ");
    }
    System.out.print("\n");
}
    
```

Quadro 7: Exemplo de código necessário para ler as células de valores de um objeto mondrian.olap.Result

O código no Quadro 7 percorre todas as linhas da tabela e para cada linha percorre a lista de colunas da tabela, utilizando as coordenadas de linha e coluna para obter o conteúdo de cada célula.

5.3 HTML e CSS para visualização dos dados

A visualização dos dados é produzida gerando-se o código *HTML* que representa uma tabela, composto pelas *tags* `<table>`, `<tr>`, `<th>`, `<td>` (W3C, 2011). Para encapsular a transformação de um objeto Result em código HTML, foram criadas classes que representam cada uma destas *tags*. A Figura 10 demonstra o diagrama destas classes.

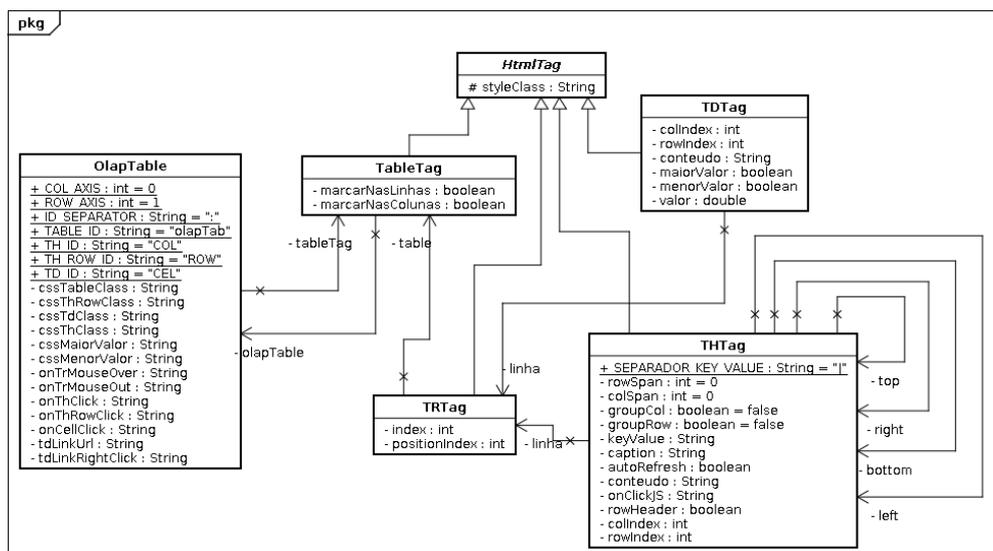


Figura 10: Diagrama de classes utilizadas para gerar o código HTML.

Novembro de 2011, FACCAT, Taquara - RS

As classes mostradas na Figura 10 têm como função principal gerar o código HTML dos dados visualizados, e como funções secundárias implementar o agrupamento de cabeçalhos de colunas e linhas, definir a formatação utilizando código CSS (W3C, 2011) e identificar visualmente o maior e menor valor em cada linha ou coluna.

A classe **OlapTable** é a responsável por receber um objeto `mondrian.olap.Result` e gerar o objeto **TableTag** com seus respectivos objetos **TRTag**, **TDTag** e **THTag** que correspondem respectivamente às linhas, células e cabeçalhos da tabela. A fim de implementar o agrupamento de cabeçalhos, que em HTML é definido pelos atributos *colspan* e *rowspan*, cada objeto da classe **THTag** possui atributos que apontam para as células adjacentes. Este mapeamento permite que, após carregados todos os dados, a classe possa aplicar os atributos *colspan* e *rowspan* nos cabeçalhos que possuem o mesmo título, como por exemplo na coluna “Modalidade” da tabela demonstrada na Figura 9. O código que implementa o agrupamento de cabeçalhos são demonstrados nos Quadros 8 e 9 a seguir.

```
public static void calculateColSpan(THTag tagInicial, THTag tagAtual) {
    if (tagInicial != null && tagAtual != null) {
        if (tagAtual.equals(tagInicial)) {
            tagInicial.colSpan = 1;
            tagInicial.groupCol = false;
            if (tagInicial.keyValue != null) {
                if (tagInicial.getRight() != null
                    && tagInicial.keyValue.equals(tagInicial.getRight().keyValue)) {
                    calculateColSpan(tagInicial, tagInicial.getRight());
                }
            }
        } else {
            tagInicial.colSpan += 1;
            tagAtual.groupCol = true;
            if (tagAtual.getRight() != null
                && tagInicial.keyValue.equals(tagAtual.getRight().keyValue)) {
                calculateColSpan(tagInicial, tagAtual.getRight());
            }
        }
    }
}
```

Quadro 8: Código fonte para calcular o agrupamento de cabeçalhos nas colunas da tabela

O código mostrado no Quadro 8 percorre recursivamente os cabeçalhos de coluna, da esquerda para a direita e calcula o *colspan* células que possuam o título de cabeçalhos iguais.

```
public static void calculateRowSpan(THTag tagInicial, THTag tagAtual) {
    if (tagInicial != null && tagAtual != null) {
        if (tagAtual.equals(tagInicial)) {
            tagInicial.rowSpan = 1;
            tagInicial.groupRow = false;
            if (tagInicial.keyValue != null) {
                if (tagInicial.getBottom() != null
                    && tagInicial.keyValue.equals(tagInicial.getBottom().keyValue)) {
                    calculateRowSpan(tagInicial, tagInicial.getBottom());
                }
            }
        } else {
            tagInicial.rowSpan += 1;
            tagAtual.groupRow = true;
            if (tagAtual.getBottom() != null
                && tagInicial.keyValue.equals(tagAtual.getBottom().keyValue)) {
                calculateRowSpan(tagInicial, tagAtual.getBottom());
            }
        }
    }
}
```

Quadro 9: Código fonte para calcular o agrupamento de cabeçalhos nas linhas da tabela

O código mostrado no Quadro 9 percorre as linhas, de cima para baixo a procura de cabeçalhos de linha iguais para calcular o atributo rowspan.

6. Utilização do Portal da Transparência Participativa

Para iniciar uma análise, o usuário deve acessar a aba “Tabela de Cruzamentos”, na interface web, que é demonstrada na Figura 11 a seguir.

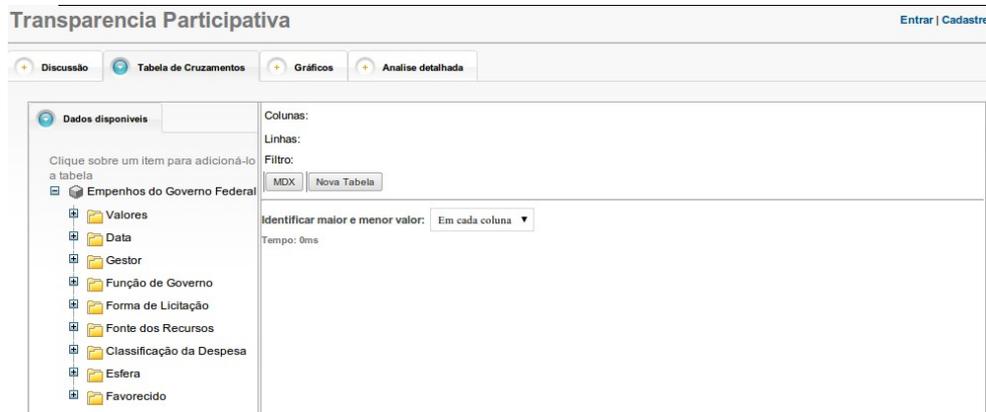


Figura 11: Tela inicial exibida ao usuário que deseja iniciar uma análise

Como mostrado na figura 11, no lado esquerdo da tela é exibida uma lista das dimensões disponíveis e os dados de cada dimensão. O usuário deve escolher as dimensões que irão compor as linhas e colunas da tabela de análise.

Para selecionar uma dimensão, o usuário deve abrir a pasta que refere-se a mesma e então selecionar o nível dos dados que deseja avaliar. Por exemplo, para criar uma uma tabela com colunas orientas ao trimestre, o usuário deverá abrir a dimensão data e clicar no item “Trimestre”. Como resultado desta ação o sistema irá abrir a tela de configuração da dimensão mostrada na Figura 12.

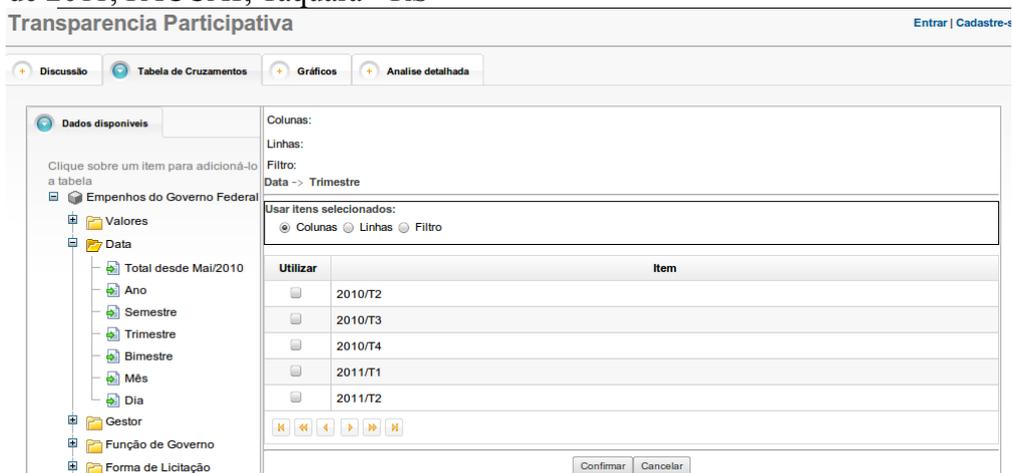


Figura 12: Selecionar uma dimensão

Na tela mostrada na Figura 12, o usuário deverá marcar os trimestres que deseja incluir na análise e escolher se deseja usar esta dimensão como colunas ou como linhas da tabela a ser exibida e em seguida clicar no botão “Confirmar”.

O próximo passo é selecionar a dimensão que será usada nas linhas da tabela, definindo dessa forma quais dados serão cruzados nesta análise. O procedimento é o mesmo descrito no parágrafo anterior, porém desta vez, marcando a opção “Linhas”. Para exemplificar, selecionaremos as Funções de Governo “Ciência e Tecnologia”, “Cultura” e “Educação”. Após clicar em confirmar, como já existe pelo menos 1 dimensão selecionada para linha e 1 para coluna, o sistema monta a tabela com estas dimensões. O resultado é mostrado na Figura 13.

identificar maior e menor valor: Em cada coluna ▼

Função	2010/T2	2010/T3	2010/T4	2011/T1
CIENCIA E TECNOLOGIA	892,677,331.35	1,500,369,512.40	2,280,457,806.95	1,573,289,945.04
CULTURA	143,773,219.76	234,540,063.23	371,471,917.07	40,920,014.14
EDUCACAO	2,032,208,343.85	5,878,261,025.61	6,829,231,648.91	14,554,060,755.97

Figura 13: Exemplo de cruzamento de dados

Como pode ser visto na Figura 13, foi montada uma tabela com as duas dimensões selecionadas. O usuário pode acrescentar novas dimensões, tanto às linhas quanto às colunas, repetindo o procedimento descrito. Ao fazer isso, as dimensões são acrescentadas como subgrupos das dimensões já existentes. Por exemplo, ao acrescentar a dimensão 'Modalidade de Licitação' às linhas da tabela, seria acrescentada a coluna 'Modalidade' logo após a coluna 'Função'. Com isto a tabela iria exibir o total de despesas por modalidade dentro de cada função de governo.

Ainda na figura 13, percebe-se que o sistema identificou visualmente qual o maior e o menor valor em cada coluna, que no contexto desta consulta significa indicar a função de governo que recebeu mais ou menos recursos, em cada trimestre. Caso o usuário altere a opção “Identificar maior e menor valor” para “Em cada linha”, o sistema irá exibir o trimestre em que cada função de

Novembro de 2011, FACCAT, Taquara - RS

governo recebeu mais ou menos recursos. Este recurso permite a detecção de desvios de forma visual e interativa.

Outra forma de analisar visualmente os dados é através da geração de gráficos da tabela gerada. O sistema oferece a possibilidade de geração de gráficos usando como séries os dados das linhas ou das colunas da tabela. Por exemplo, na figura 13, se o usuário clicar na função 'Cultura', o sistema irá exibir um submenu com a opção 'Gerar Gráfico'. Ao selecionar esta opção será gerado um gráfico de linhas exibindo as despesas trimestrais na função 'Cultura', considerando os 4 trimestres que aparecem nas colunas desta tabela.

Ainda usando como exemplo os dados da figura 13, chama a atenção o fato de que houve um aumento substancial nas despesas referentes a função Educação no 1º Trimestre de 2011. Para analisar em detalhes este item o usuário pode clicar sobre este valor e selecionar a opção “Analisar”, que o direciona para a aba “Análise detalhada”, onde é possível consultar quem foram os favorecidos e também todos os documentos que registram as despesas referentes a este item, conforme mostra a Figura 14.

Nr. Documento Empenho	Tipo de Empenho	Espécie de Empenho	Doc. Ident. Favorecido	Nome do Favorecido	Observação do Documento	Valor do Documento
2011NE748006	ESTIMATIVO	ORIGINAL	06307102000130	PREFEITURA DE SAO LUIS	COMPLEMENTACAO DA UNIAO AO FUNDEB - FUNDEB - 2011 - FUNDEB - COMPLEMENTACAO DO FUNDEB MA	1.287.139.674,20
2011NE748005	ESTIMATIVO	ORIGINAL	05055009000113	BELEM PREFEITURA	COMPLEMENTACAO DA UNIAO AO FUNDEB - FUNDEB - 2011 - FUNDEB - COMPLEMENTACAO DO FUNDEB MA	1.241.053.251,53
2011NE748003	ESTIMATIVO	ORIGINAL	13927801000149	SALVADOR PREFEITURA	COMPLEMENTACAO DA UNIAO AO FUNDEB - FUNDEB - 2011 - FUNDEB - COMPLEMENTACAO DO FUNDEB MA	1.122.526.865,26

Figura 14: Exemplo de consulta de documentos referentes a Educação no 1º trimestre de 2011.

A funcionalidade mostrada na Figura 14 é justamente o que acrescenta maior inteligência à análise dos dados de despesas diárias do Portal da Transparência, pois ao invés de analisar todo o montante de documentos, o usuário poderá agrupar suas análises, restringindo-se somente aos documentos que aparentarem ter alguma relevância.

O usuário têm a possibilidade de salvar suas análises para consultas futuras e ainda compartilhá-las com outros usuários, iniciando uma discussão a respeito dos itens analisados. A Figura 15 mostra a tela de discussões do Portal da Transparência.

Ordenar: Mais Recentes

Procurar: OK

Tópicos em discussão

Nr.	Titulo	Autor	Publicado
8	Aumento nas despesas do ministerio da educação	Tiago Wiedemann	29/Jun/2011 22:27
17	teste	Tiago Wiedemann	29/Jun/2011 23:33
23	Prefeitura de salvador recebe 10000000 do ministerio da cultura	Tiago Wiedemann	29/Set/2011 13:56

[H](#) [<](#) [>](#) [H](#)

Aumento nas despesas do ministerio da educação

Analisando os totais por trimestre, percebi um aumento de 7 bilhões para 22 bilhões nos gastos do ministerio da educação, no primeiro trimestre de 2011.

Update: conforme identificamos nos comentários, isto deve-se a um aumento no Fundo Nacional de Educação.

Comentarios Anexos

Comentário de: Joao da Silva Em: 30/Jun/2011 - 10:38

Houve um aumento significativo no valor destinado ao fundo nacional de educação. De uma olhada na tabela que inseri nos anexos.

Responder

Comentário de: Tiago Wiedemann Em: 30/Jun/2011 - 10:43

Realmente, já editei o tópico para inserir esta informação.

Joao da Silva escreveu:

Houve um aumento significativo no valor destinado ao fundo nacional de educação. De uma olhada na tabela que inseri nos anexos.

Responder

Figura 15: Aba de Discussões

Na tela mostrada na Figura 15, o usuário poderá consultar as discussões em andamento e também dar sua contribuição às mesmas, após efetuar seu login através do link “Entrar”. Caso o usuário não possua senha, poderá cadastrar-se rapidamente utilizando o link “Cadastre-se”. Uma vez autenticado no sistema, o usuário terá acesso ao botão “Novo tópico” e também aos links “Responder” abaixo de cada tópico criado.

7. Conclusões e Trabalho Futuro

O software desenvolvido atendeu ao objetivo de expandir as funcionalidades do portal da transparência. Como demonstrado no Capítulo 6 o usuário passa a ter a possibilidade de analisar as despesas governamentais iniciando em um nível macro, realizando cruzamentos entre as diversas dimensões das despesas e então identificando visualmente possíveis anomalias.

O sistema oferece então a possibilidade de consultar apenas os documentos que detalham as operações que resultaram na variação de despesas que o usuário deseja analisar. Além da possibilidade de analisar na forma de gráficos as linhas e colunas das tabelas montadas para análise.

A função de discussões permite ao usuário compartilhar suas análises com outras pessoas, criando um ambiente de incentivo onde grupos de pessoas interessadas nos gastos públicos podem se reunir para analisar em conjunto a destinação que o governo vêm fazendo dos recursos públicos.

A partir dos dados organizados do *Data Warehouse* podem ser desenvolvidas novas funcionalidades uteis ao cidadão que deseja fiscalizar de perto as ações do Governo. Uma das possibilidades é a criação de ferramentas de mineração para descoberta de conhecimento em bases de dados, que segundo Fayyad et al (1996) é um processo para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados. Tais ferramentas poderiam ser utilizadas para identificar de forma automática padrões e anomalias nos gastos públicos.

Novembro de 2011, FACCAT, Taquara - RS

Outra funcionalidade útil seria a geração de links permanentes para as tabelas e gráficos gerados. Esta funcionalidade iria expandir ainda mais o conceito de colaboração, visto que os links poderiam ser enviados por e-mail ou ainda serem utilizados como referência em outros trabalhos.

Referências

ARMSTRONG, Eric; et al. **The J2EE Tutorial**. Santa Clara CA: Sun Microsystems, Inc., 2006.

ANZANELLO, Cynthia Aurora . **OLAP Conceitos e Utilização**, Porto Alegre, RS: Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS). Disponível em: <<http://www.fag.edu.br/professores/limanzke/Administra%E7%E3o%20de%20Sistemas%20de%20Informa%E7%E3o/OLAP.pdf>>. Acesso em: 10 out. 2011

BRASIL. Lei Complementar nº 101, de 4 de maio de 2000. Estabelece normas de finanças públicas voltadas para a responsabilidade na gestão fiscal e dá outras providências. Disponível em <http://www.planalto.gov.br/ccivil_03/Leis/LCP/Lcp101.htm>. Acesso em: 02 out. 2011.

BRASIL. Lei Complementar nº 131, de 27 de maio de 2009. Acrescenta dispositivos à Lei Complementar no 101, de 4 de maio de 2000, que estabelece normas de finanças públicas voltadas para a responsabilidade na gestão fiscal e dá outras providências, a fim de determinar a disponibilização, em tempo real, de informações pormenorizadas sobre a execução orçamentária e financeira da União, dos Estados, do Distrito Federal e dos Municípios. Disponível em <http://www.planalto.gov.br/ccivil_03/Leis/LCP/Lcp131.htm>. Acesso em: 02 out. 2011.

COLEMAN, David; KHANNA, Raman. **Groupware Technology and Applications**, Upper Saddle River, NJ: Prentice Hall PTR, 1995.

FAYYAD, Usama; SHAPIRO, Gregory Piatsky; SMYTH, Padhraic. **The KDD process for extracting useful knowledge from volumes of data**. New York, NY: ACM, 1996, Communication of the ACM, Volume 39, Issue 11, (November, 1996) pages 27-34. Disponível em: <<http://portal.acm.org/citation.cfm?id=240455.240464>>. Acesso em 08 Set. 2010.

KIMBALL, Ralph. **Data Warehouse Toolkit**, São Paulo, SP: Makron Books do Brasil Editora Ltda, 1998

MICROSOFT. **MDX Language Reference**. Disponível em <<http://msdn.microsoft.com/en-us/library/ms145595.aspx>>. Acesso em 14 out 2011a

MICROSOFT. **The Basic MDX Query**. Disponível em <<http://msdn.microsoft.com/en-us/library/ms144785.aspx>>. Acesso 14 out. 2011b

MONDRIAN PROJECT. Disponível em: <<http://mondrian.pentaho.com/>>. Acesso em: 02 out. 2011

ORACLE Technology Network – Java – Java EE, **Java EE at a Glance**. Disponível em <<http://www.oracle.com/technetwork/java/javaee/overview/index.html>>. Acesso em 02 out. 2011.

PASUMANSKY, Mosha. **MDX For Everyone**, 1998 disponível em: <<http://www.mosha.com/msolap/articles/MDXForEveryone.htm>>. Acesso em 14 out. 2011

PENTAHO. **Open Source Business Intelligence**, disponível em: <<http://www.pentaho.com/>>. Acesso em 02 out. 2011

PORTAL DA TRANSPARÊNCIA. Disponível em: <<http://www.transparencia.gov.br>>. Acesso em: 02 out. 2011.

PRESIDENCIA DA REPUBLICA. Disponível em: <<http://www.planalto.gov.br>>. Acesso em: 02 out. 2011.

PRESSMAN, Roger S. **Engenharia de Software**. 6ª Edição. São Paulo, SP: McGraw-Hill. 2009.

ROLDÁN, María Carina. **Pentaho 3.2 Data Integration**. Birmingham, B27 6PA, UK: Packt Publishing Ltd., 2010.

Novembro de 2011, FACCAT, Taquara - RS

SCHRENK, Michael, **Webbots, Spiders, and Screen Scrappers**. São Francisco, CA: No Starch Press, 2007.

TRANSPARÊNCIA BRASIL. **Página da ONG Transparência Brasil**, disponível em: <<http://www.transparencia.org.br/>>. Acesso em 08 Set. 2010.

W3C – World Wide Web Consortium – Standards. **HTML & CSS**, disponível em: <<http://www.w3.org/standards/webdesign/htmlcss>>. Acesso em 17 out. 2011

Dados do autor:

Nome completo: Tiago Wiedemann

Nº Matrícula: 100115

Telefone: (51) 91332500

e-mail: tiagowrs@gmail.com