



UNIVERSIDADE DE SÃO PAULO

Escola de Artes, Ciências e Humanidades

Relatório Técnico PPgSI-002/2013
*Definição e Validação de uma Ontologia para o
Orçamento Público Federal Brasileiro (v.1.0)*

Livio Cravo Martins
Gisele da Silva Craveiro
José de Jesús Pérez Alcázar

Fevereiro - 2013

O conteúdo do presente relatório é de única responsabilidade dos autores.

Série de Relatórios Técnicos

PPgSI-EACH-USP. Rua Arlindo Béttio, 1000 - Ermelino Matarazzo -
03828-000

São Paulo, SP.

TEL: (11) 3091-8197

<http://www.each.usp.br/ppgsi>

Definição e Validação de uma Ontologia para o Orçamento Público Federal Brasileiro (v.1.0)

Livio Cravo Martins¹, Gisele Silva Craveiro¹, José de Jesús Pérez Alcázar¹

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo
São Paulo – SP, Brasil

{livio.martins, giselesc, jperez}@usp.br

Resumo. *Uma boa estruturação do Orçamento Público em forma de ontologias permite que softwares façam uma interpretação contextualizada dos dados e possam realizar tarefas complexas para gerar formas de visualização mais simples para o cidadão comum. Isso facilita o controle social, tornando a administração pública mais efetiva. Nesse contexto, este trabalho propõe uma ontologia para a estrutura da elaboração e execução das receitas e despesas do Orçamento Público Federal Brasileiro, com a qual será possível a análise e comparação dos dados orçamentários de vários anos. Como metodologia de desenvolvimento de ontologias foi utilizado o método iterativo Deronto, que possibilita a definição de uma ontologia por meio de um diagrama entidade-relacionamento (DER) do domínio. A Ontologia foi construída usando o editor de ontologias Protégé. Para validar a ontologia, foram realizadas inferências e consultas com os dados do orçamento federal referentes à execução das receitas e despesas dos anos de 2010 e 2011. Com isso, é demonstrada uma proposta de uso que poderá auxiliar no desenvolvimento de aplicações que permitam ao cidadão exercer seu direito de fiscalização dos gastos públicos. A aplicação de teste da ontologia foi criada usando o framework JENA.*

1. Introdução

O acesso às informações de interesse público é essencial para a transparência e o fortalecimento da democracia. O cidadão é quem se encontra mais próximo das ações e serviços desenvolvidos pelo governo, além de ser o maior beneficiário [Evangelista 2010], tendo então, um importante papel fiscalizador e de controle social.

Um grande aliado no controle social e na fiscalização do governo é o Orçamento Público. Ele determina quais ações (despesas) serão feitas pelo governo com os recursos originados de contribuições (receitas) da sociedade. Assim, segundo [Paludo 2011], é um dos meios de publicidade mais importantes do governo, sendo utilizado tanto para gerenciar e controlar a aplicação dos recursos públicos, quanto para monitorar os gastos realizados pelo governo [Giacomoni 2009].

O Brasil em relação à transparência fiscal tem grande reconhecimento internacional. A obrigação de publicar a execução orçamentária na web de forma detalhada e em tempo real existe desde 2009 [Brasil 2009] e é válida para todos os órgãos dos poderes Executivo, Legislativo e Judiciário em todos os níveis, municipal, estadual e federal. Apenas para dar uma idéia de escala, o país conta com 5.565 municípios e 26 estados, além do nível federal.

Apesar dessa vanguarda na publicação na web em larga escala, e com a fiscalização e controle das ações do governo previstos na Constituição Federal [Brasil 1988] e regulamentados por leis [Brasil 2009] [Brasil 2000] [Brasil 2011], ainda há o que fazer para

facilitar o acesso e a análise de dados. Foi constatado em [da Silva Craveiro et al. 2012], que apesar de haver uma crescente disponibilização de dados nos últimos anos, notou-se uma grande deficiência nos mesmos. Esse mesmo estudo aponta a necessidade de padronização e estruturação para possibilitar a construção de aplicativos que integrem dados de diferentes origens e assim permitir que o cidadão comum possa efetivamente acompanhar a execução do orçamento público.

A preocupação com a padronização e estruturação do orçamento público está também presente no âmbito internacional e atualmente, a [Partnership 2012], que congrega mais de 50 países, possui grupos de trabalho discutindo propostas para essa finalidade [for Fiscal Transparency 2012] [OpenSpending 2012].

Para auxiliar na estruturação e interpretação de dados, surgiram as ontologias. Elas representam uma camada da Web Semântica [Antoniou and Harmelen 2008], e consistem em um modelo de dados que possibilita que as informações sejam entendidas não só pelas pessoas, mas também pelas máquinas, além de permitirem o compartilhamento de seus termos e sua semântica. Essa forma cria condições para que os softwares, através de uma interpretação contextualizada dos dados disponibilizados, possam realizar tarefas complexas para o usuário [Antoniou and Harmelen 2008].

Assim, com uma estruturação do domínio orçamentário em forma de ontologia, os dados do orçamento poderão ser processados e conectados por máquinas, permitindo a interoperabilidade dos sistemas e o fornecimento de formas mais agradáveis de visualização dos dados ao cidadão [Júnior and Rover 2009]. Neste contexto, o presente trabalho propõe uma ontologia para a estruturação das receitas e despesas do Orçamento Público Federal Brasileiro. Apesar de focar na esfera federal, a legislação que regulamenta sua estrutura é nacional [Brasil 1964], logo, o orçamento federal serve de base para os orçamentos de todas as instâncias e esferas governamentais, podendo então sua estrutura ser adaptada em outros entes federativos. Como metodologia de desenvolvimento de ontologias foi utilizado o método iterativo Deronto [Caliari 2007], que possibilita a definição de uma ontologia por meio de um diagrama entidade-relacionamento (DER) do domínio. Para validar a ontologia, foram realizadas inferências e consultas com os dados do orçamento federal referentes à execução das receitas e despesas dos anos de 2010 e 2011 obtidos no Portal Siga Brasil [Federal 2012].

2. Orçamento Público

O Orçamento Público consiste em um instrumento de planejamento e execução para as finanças públicas. Nele são estimadas as receitas e fixadas as despesas orçamentárias que serão realizadas durante o exercício correspondente.

De acordo com o Manual Técnico do Orçamento 2012 (MTO) [SOF 2011b], o planejamento orçamentário brasileiro baseia-se na elaboração e execução de três leis – o Plano Plurianual (PPA), a Lei de Diretrizes Orçamentárias (LDO) e na Lei de Orçamentos Anuais (LOA) – que, em conjunto, colocam em prática o planejamento e a execução das políticas públicas. O PPA consiste no planejamento de um período de quatro anos nos quais nenhuma despesa que ultrapasse um exercício poderá ocorrer sem ter sido nele incluída. A LDO tem como principal finalidade orientar a elaboração dos orçamentos fiscais e da seguridade social e de investimento do Poder Público. A LOA, por sua vez, é o orçamento propriamente dito, contendo as previsões de receitas e fixação das despesas

que serão executadas no exercício correspondente.

Na Figura 1 temos um resumo do processo integrado de planejamento e orçamento. A integração do PPA e a LOA fica bem explicitada pelo papel cumprido pela LDO [Giacomoni 2009], pois além da Lei de Diretrizes Orçamentárias fornecer orientação para a elaboração dos orçamentos anuais, ela também destaca, da programação plurianual, as prioridades e metas a serem cumpridas em cada LOA.

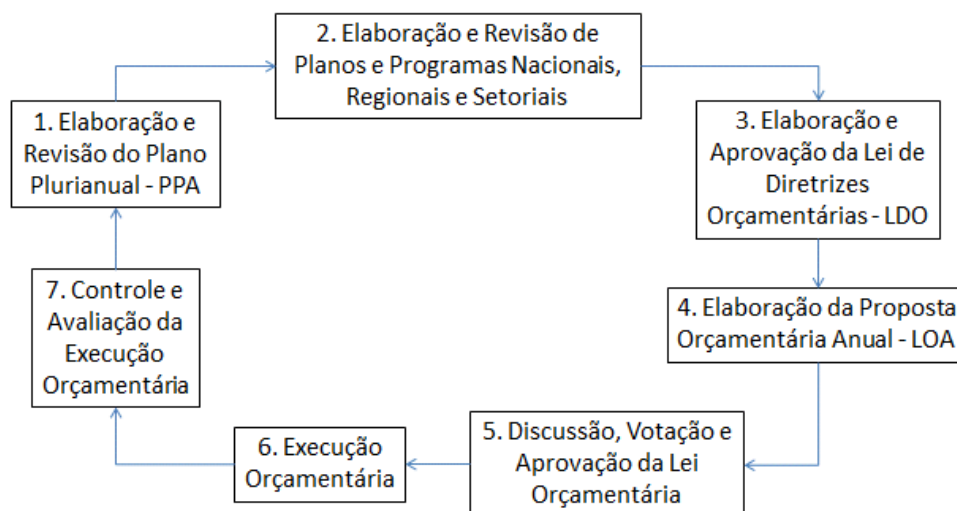


Figura 1. Processo integrado de planejamento e orçamento [Giacomoni 2009].

A fim de conferir racionalidade, eficiência e transparência aos processos de elaboração, execução e controle do orçamento público [SOF 2011b] foram estabelecidos os seguintes princípios orçamentários [Brasil 1964] [Brasil 1988]:

- **Unidade ou Totalidade:** cada ente governamental deve elaborar uma única LOA;
- **Universalidade:** a LOA de cada ente federado deve conter todas as receitas e despesas de todos os Poderes, órgãos, entidades, fundos e fundações instituídas e mantidas pelo poder público;
- **Anualidade ou Periodicidade:** o exercício financeiro coincide com o ano civil (1º de janeiro a 31 de dezembro) e corresponde ao período de tempo ao qual se referem à previsão das receitas e fixação das despesas documentadas na LOA;
- **Exclusividade:** a LOA não conterá nada além da previsão das receitas e fixação das despesas;
- **Orçamento Bruto:** o registro das receitas e despesas deve ser feito com seu valor total e bruto, sem quaisquer deduções;
- **Não Vinculação da Receita de Impostos:** é vedada a vinculação de receitas de impostos à órgão, fundo ou despesa, salvo exceções estabelecidas pela CF.

Para o propósito deste trabalho nos dedicamos à modelagem da execução orçamentária anual (Etapa 6 da Figura 1), pois a partir dela temos a realização do que foi arrecadado (receitas) e os investimentos realizados (despesas). A estrutura das receitas e das despesas serão explicadas nas Seções 2.1 e 2.2, respectivamente.

2.1. Receitas

De acordo com o MTO, as receitas públicas são ingressos de recursos financeiros nos cofres do Estado. Quando representam recursos financeiros para aquele ano, desdobram-se em receitas orçamentárias e integram a LOA.

As receitas orçamentárias possuem três classificações simultâneas. São elas:

- Classificação por Natureza: identifica a receita segundo seu fato gerador. Menor nível de detalhamento das informações orçamentárias sobre as receitas públicas, a natureza da receita é definida por um código numérico de oito dígitos subdivididos em seis níveis: categoria econômica (1º dígito), origem (2º dígito), espécie (3º dígito), rubrica (4º dígito), alínea (5º e 6º dígitos) e subalínea (7º e 8º dígitos). Na Figura 2 é possível ver um exemplo de uma classificação por natureza de uma receita;

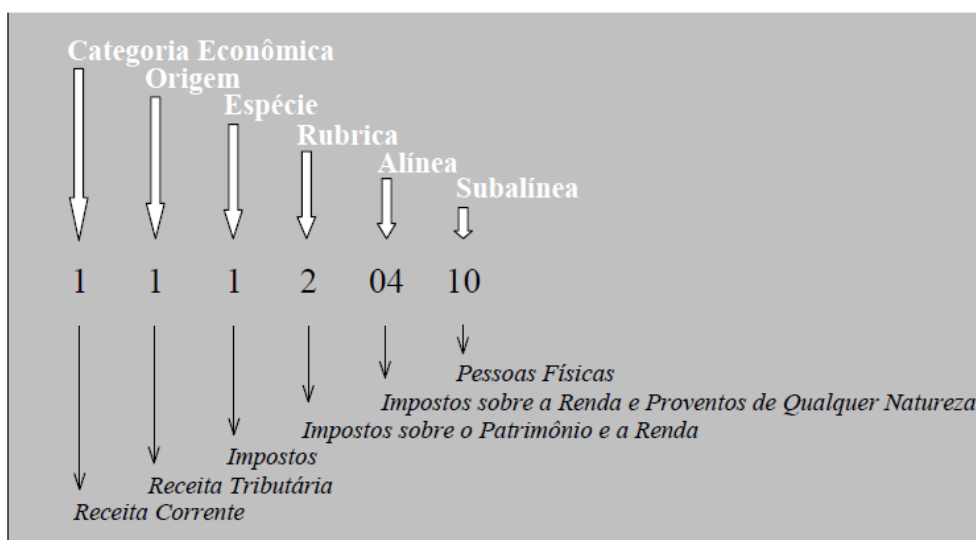


Figura 2. Classificação por Natureza da Receita [SOF 2011b].

- Classificação por Identificador de Resultado Primário: identifica a receita como primária (vinda de impostos, taxas, etc.) ou financeira;
- Classificação por Fonte/Destinação de Recursos: identifica as fontes de financiamento dos gastos públicos. É um mecanismo integrador entre receita e despesa. Na receita, indica qual a destinação do recurso, na despesa, a origem dos recursos utilizados.

O e mentário das receitas [SOF 2011a] ainda define uma quarta classificação:

- Classificação por Grupo da Receita: identifica quais agentes públicos possuem competência legal para arrecadar, fiscalizar e administrar as Receitas Públicas.

Além da estrutura, uma receita apresenta diferentes status ao longo do tempo. Os estágios da receita orçamentária compreendem as fases de previsão, lançamento, arrecadação e recolhimento da receita [SOF 2011b]. A previsão consiste em estimar o valor das receitas que constarão na proposta orçamentária. Além de anteceder a fixação das despesas que constarão na LOA, serve de base também para estimar as necessidades de financiamento do governo. O lançamento é o ato de verificação da procedência do

crédito fiscal e inscrição do débito à pessoa que lhe é devedor. Situa-se no contexto de crédito tributário, aplicando-se a impostos, taxas e contribuições de melhoria. A arrecadação é a entrega dos recursos devidos ao Tesouro Nacional. Pertencem ao exercício financeiro as receitas nele arrecadadas [Brasil 1964]. Por último, o recolhimento consiste na transferência dos recursos arrecadados à conta específica do Tesouro Nacional [Brasil 1964].

2.2. Despesas

A fim de permitir um melhor detalhamento dos gastos públicos [Slomski 2008], a despesa orçamentária possui uma classificação estruturada, subdividida em Programação Qualitativa e Programação Quantitativa. A Tabela 1 apresenta um exemplo de estrutura da despesa.

Segundo [SOF 2011b], a Programação Qualitativa é composta pelos seguintes blocos:

- Esfera: informa sobre a qual orçamento pertence à despesa. O orçamento pode ser: Fiscal, da Seguridade Social ou de Investimento;
- Classificação Institucional: fornece a informação sobre qual órgão e unidade orçamentária realiza a despesa. É definida por um código de cinco dígitos, nos quais os dois primeiros representam o órgão, o restante, a unidade orçamentária;
- Classificação Funcional: dividida em função e subfunção, essa classificação responde a indagação “em que” área de ação aquela despesa será realizada. É definida por um código de cinco dígitos, os dois primeiros representam a função, o restante, a subfunção;
- Classificação Programática: toda ação do governo está estruturada em programas orientados para atingir o objetivo do período correspondente ao PPA. Os programas podem ser temáticos ou de gestão, manutenção e serviços ao Estado. São identificados por um código de quatro dígitos e contém uma ou mais ações. As ações são operações da qual resultam produtos que contribuem para o objetivo de um programa. Cada ação é identificada por um código alfanumérico de oito dígitos. O primeiro dígito identifica o tipo da ação, podendo ser um projeto, uma atividade ou uma operação especial. Do segundo ao quarto é detalhada a ação. Os quatro últimos identificam o subtítulo da ação, estes são utilizados especialmente para descrever a localização da ação.

A Programação Quantitativa define a programação física, ou seja, quanto será desenvolvido e também a programação financeira, que determina o que adquirir e com quais recursos. Essas informações quantitativas estão definidas nos seguintes blocos:

- Meta Física da Ação: define, em nível de subtítulo, a quantidade do produto que será ofertado em uma determinada ação;
- Natureza da Despesa: semelhante à natureza da receita, também é definida por um código numérico de oito dígitos, porém, subdivididos em cinco níveis: categoria econômica (1º dígito), grupo de natureza da despesa (GND) (2º dígito), modalidade da aplicação (3º e 4º dígitos), elemento de despesa (5º e 6º dígitos) e subelemento (7º e 8º dígitos);
- Identificador de Uso (IDUSO): destina-se a indicar se os recursos compõem contrapartida nacional de empréstimos ou de doações ou destinam-se a outras aplicações;

- Identificador de Doação e de Operação de Crédito (IDOC): identifica as doações de entidades internacionais ou operações de crédito contratuais;
- Identificador de Resultado Primário: tem como finalidade auxiliar a apuração do resultado primário previsto na LDO.

Na Tabela 1 temos um código-exemplo da estrutura completa da programação retirada do MTO.

Tabela 1. Código-exemplo da Estrutura Completa da Programação [SOF 2011b].

CÓDIGO COMPLETO*		10.	39.	252.	26.	782.	N059.	7M64.	0001.	9999.	0.	118.	4490.	2
Q U A L I T A T I V A	Esfera: Orçamento Fiscal	10												
	CLASSIFICAÇÃO INSTITUCIONAL	Órgão: Ministério dos Transportes	39											
		Unidade Orçamentária: Departamento Nacional de Infraestrutura de Transportes – DNIT		252										
	CLASSIFICAÇÃO FUNCIONAL	Função: Transporte			26									
		Subfunção: Transporte Rodoviário				782								
	CLASSIFICAÇÃO PROGRAMÁTICA	PROGRAMA: Transporte Rodoviário					N059**							
	AÇÃO: Construção de Trecho Rodoviário							7M64						
	SUBTÍTULO: Rio Grande do Sul								0043					
Q U A N T I T A T I V A	IDOC: Outros recursos									9999				
	IDUSO: Recursos não destinados à contrapartida										0			
	Fonte da Receita: Recursos do Tesouro - Exercício Corrente (1) Recursos Ordinários (00)											100		
	Natureza da Despesa: Categoria Econômica: Despesas de Capital (4); Grupo de Natureza: Investimentos (4); Modalidade de Aplicação: Aplicação Direta (90)												4490	
	Identificador de Resultado Primário: Primária Discricionária													2

*Código visualizado no SIAFI.

** Código provisório até a conclusão da fase qualitativa.

Uma despesa também passa por diferentes estágios: o empenho, liquidação e, por último, o pagamento [SOF 2011b]. O empenho é o primeiro estágio de uma despesa, é o ato de criar obrigação de pagamento pendente para o Estado. A liquidação tem o objetivo de verificar se a despesa foi regularmente empenhada e que a entrega do bem ou serviço foi realizada de forma satisfatória. Após a liquidação é efetuado o pagamento ao responsável pela prestação do serviço ou fornecimento do bem [SOF 2011b].

3. Web Semântica e Ontologias

A Web é baseada no conceito de compartilhamento de informações, mas cresceu de forma quase anárquica [Antoniou and Harmelen 2008]. Como sabemos, diariamente são gerados volumes cada vez maiores de dados. Ao pesquisar sobre um assunto, encontramos milhares de resultados, porém desestruturados. A Web Semântica, uma evolução da web atual, surge neste contexto com a intenção de estruturar os dados da internet de forma que os mesmos possam ser interpretados por máquinas, sendo possível o uso de técnicas inteligentes que tiram proveito da semântica desses dados. Desta forma, conseguimos além de estruturá-los, permitir um mecanismo de busca mais eficiente que não necessite do processamento de linguagem natural em milhares de páginas.

Para o desenvolvimento dessas estruturas, é comum o uso de ontologias [Antoniou and Harmelen 2008]. Uma ontologia é uma especificação explícita formal de um conceito compartilhado [Antoniou and Harmelen 2008] [Studer et al. 2008]. Elas representam um conjunto de conceitos de um determinado domínio, seus relacionamentos e interações. São utilizadas quando se necessita de um vocabulário compartilhado para troca de informações e permite a interoperabilidade semântica entre conceitos, habilitando o computador a processar o dado de acordo com sua semântica, e não apenas de acordo com o conteúdo de seu texto [Allemang and Hendler 2011].

Na Figura 3, temos a representação da "pilha" da Web Semântica. Observa-se que na base da pilha encontram-se o *Unicode*, que permite a representação de alfabetos modernos à escala mundial, e o *Uniform Resource Identifier (URI)*, que permite a identificação de recursos na Web.

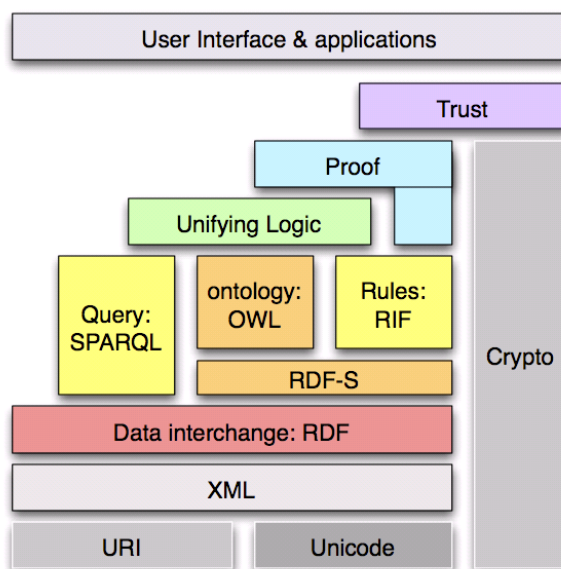


Figura 3. Pilha de Normalizações da Web Semântica [Bratt 2006].

Acima, a *Extensible Markup Language (XML)* permite a interoperabilidade entre diferentes aplicações que partilhem dados com formatos distintos, além do mecanismo de namespaces que trás a desambiguação de termos iguais.

O *Resource Description Framework (RDF)* compreende as especificações do modelo e sua sintaxe, possibilitando a descrição dos recursos por meio de suas propriedades e valores. As informações em RDF são representadas na forma de triplas sujeito-predicado-objeto. O sujeito e o objeto são os recursos que relacionam-se por meio da propriedade (ou predicado).

Acima do RDF, temos o *RDF-Schema (RDFS)*. O RDFS é uma extensão semântica do RDF [Antoniou and Harmelen 2008] utilizado para a descrição do vocabulário RDF, possibilitando a definição de taxonomias de recursos em termos de uma hierarquia de classes, criando uma ontologia mais rica.

Ainda diretamente acima do RDF, temos a *Simple Protocol and RDF Query Language (SPARQL)*. Assim como a *Structured Query Language (SQL)* permite a consulta

e modificação das bases de dados relacionais, a SPARQL permite a realização de consultas em bases de dados no formato RDF [Antoniou and Harmelen 2008]. Esta linguagem tornou-se recomendação da *World Wide Web Consortium* (W3C) e muito se assemelha com o SQL.

Acima do RDFS, temos a linguagem de programação de ontologias indicada pela W3C, a linguagem *Web Ontology Language* (OWL). Essa linguagem estende o RDFS e, conseqüentemente, o RDF. Segundo [Antoniou and Harmelen 2008], a linguagem OWL permite descrever formalmente, de modo mais eficiente, os aspectos semânticos dos termos utilizados e seus respectivos relacionamentos, possibilitando representações mais abrangentes das linguagens RDF e *RDF Schema* e favorecendo uma maior interoperabilidade. Na seção 3.1. ela será melhor detalhada.

A camada ontológica é formada pela OWL e uma linguagem baseada em regras [Antoniou and Harmelen 2008]. As rules permitem uma espécie de “introdução lógica” que será incorporada a *Unifying Logic* possibilitando o uso de lógicas avançadas [Antoniou and Harmelen 2008] em que agentes computacionais possam realizar inferências automáticas a partir das relações existentes entre os recursos informacionais, podendo inclusive inferir novas informações [Antoniou and Harmelen 2008].

A camada *Proof* possibilita a verificação/comprovação da coerência lógica dos recursos e a camada *Trust* garante que as informações estejam representadas de modo correto, possibilitando a confiabilidade das informações.

3.1. OWL

Como dito anteriormente, a linguagem OWL surgiu a partir de outras linguagens para definição de ontologias podendo ser considerada como uma extensão do *Resource Description Framework* (RDF) e do *Resource Description Framework Schema* (RDFS), pois além de possuir todas as primitivas do RDF e RDFS, adiciona algumas expressividades mais ricas. Por estender o RDF, os dados são armazenados em forma de triplas com domínio, propriedade e imagem [Antoniou and Harmelen 2008]. A linguagem OWL possui as seguintes características:

- **Indivíduos:** também chamados de instâncias, são os objetos no domínio de interesse. Eles podem ser referenciados como instâncias de classes;
- **Propriedades:** podem ser dos tipos *object*, *datatype*. Propriedades do tipo *object* relacionam dois indivíduos entre si, propriedades *datatype* relacionam o indivíduo a um valor. Além disso, as propriedades podem ter características funcionais, transitivas, simétricas e inversas. Uma propriedade inversa é sintaticamente oposta a alguma outra propriedade. A propriedade funcional tem um valor único, ou seja, um indivíduo pode ter apenas um relacionamento utilizando aquela propriedade. A propriedade transitiva define o seguinte: sejam “a”, “b” e “c” indivíduos, caso “a” se relacione com um indivíduo “b” e esse indivíduo se relacione com “c” utilizando a mesma propriedade, infere-se que “a” também se relaciona com “c”. Uma propriedade simétrica tem a característica de o relacionamento acontecer pelos dois lados, ou seja, se “a” relaciona-se com “b” com uma propriedade simétrica, “b” também se relaciona com “a” [Horridge 2008];
- **Classes:** também chamadas de conceitos, são os conjuntos que contêm os indivíduos, podendo ser organizadas na forma de uma taxonomia.

De acordo com [Horridge 2008], a linguagem OWL é dividida em três sublinguagens:

- OWL *Lite*: sintaticamente mais simples, deve ser usada quando se necessita de uma hierarquia de classes simples e apenas algumas restrições;
- OWL DL: mais expressiva que OWL *Lite*, é baseada em lógica descritiva sendo possível o raciocínio automático, por inferência, para computar automaticamente a hierarquia de classes e inconsistências no modelo. Esse raciocínio é realizado por *reasoners*, também chamado de raciocinadores;
- OWL *Full*: mais expressiva, destina-se a situações onde a expressividade é mais importante do que a decidibilidade. Com ela não é possível à realização de inferências.

3.2. SWRL

Semantic Web Rule Language (SWRL) é uma linguagem de regras para web semântica. Com ela é possível definir regras que complementam as linguagens OWL DL e OWL *Lite*. Baseia-se na definição de uma condição atômica e uma consequência também atômica. Com o SWRL é possível utilizar condições baseadas em conceitos, propriedades *datatype*, propriedades *object* e também em valores de instâncias. Porém, o processamento de um *reasoner* pode ser lento, além disso poucos deles oferecerem suporte a essa linguagem [Horrocks et al. 2004]. Com o editor de ontologias Protégé, um dos mais utilizados para definição de ontologias, é possível utilizar o *reasoner* chamado Pellet, completo para OWL DL. Desta forma, com ele é possível fazer inferências em ontologias que possuem regras SWRL [Sirin et al. 2007].

3.3. SPARQL

Assim como a *Structured Query Language* (SQL) permite a consulta e modificação das bases de dados relacionais, o *Simple Protocol and RDF Query Language* (SPARQL) permite a realização de consultas em bases de dados no formato RDF [Antoniou and Harmelen 2008]. Esta linguagem tornou-se recomendação da W3C e muito se assemelha com o SQL, porém como utilizamos um *Uniform Resource Identifier* (URI) único para definir cada recurso (seja ele classe, propriedade ou indivíduo), necessitamos definir alguns prefixos. Na Figura 4, um exemplo retirado do site da W3C.

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{ ?x foaf:name ?name .
  ?x foaf:mbox ?mbox }
```

Figura 4. Exemplo de Consulta SPARQL [W3C 2008].

Note que ao definir o prefixo foaf: não foi preciso escrever o URI "http://xmlns.com/foaf/0.1/" inteiro nas próximas utilizações. O resultado está representado na Tabela 2. Neste exemplo foi feito uma consulta para retornar um nome e um e-mail de contato.

Tabela 2. Resultado da Consulta do Exemplo da Figura 4 [W3C 2008].

name	mbox
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

3.4. Apache Jena

Jena é um *framework* Java para o desenvolvimento de aplicações web semânticas. Segundo o site [Jena 2012], este *framework* oferece diversas ferramentas e bibliotecas para o desenvolvimento dessas aplicações, incluindo: API para leitura, processamento e gravação de dados RDF, API para manipulação de ontologias OWL e RDFS, raciocinadores para RDF e OWL, *triple-stores* para armazenar e gerenciar dados em disco, mecanismo de consultas SPARQL, *endpoint* para acesso ao *triple-store*.

Com o Jena é possível persistir os dados de duas formas: com o Jena SDB e com o Jena TDB. O Jena SDB permite persisti-los em um banco de dados relacional e oferece uma forma de transformar as consultas SQL em SPARQL. O Jena TDB persiste os dados em um *triple-store* no qual é acessado por um *endpoint*, chamado Fuseki. O Fuseki cria um servidor em que é possível realizar consultas e atualizações utilizando-se a linguagem SPARQL por meio do protocolo *Hypertext Transfer Protocol* (HTTP) [Jena 2012].

4. Metodologias de Desenvolvimento de Ontologias

Do ponto de vista metodológico, não há um padrão que seja mais aceito para construção de ontologias [da Silva et al. 2008]. Além disso, de acordo com [da Silva et al. 2008] encontra-se certa dificuldade na realização de novos trabalhos para a comparação de metodologias para desenvolvimento de ontologias, pois as metodologias discutidas na literatura são quase sempre as mesmas e poucos trabalhos são publicados sobre o tema. As metodologias e métodos com maior frequência de discussão em documentos relacionados [da Silva et al. 2008] são: Metodologia de Gruninger e Fox [Gruninger and Fox 1995], Metodologia de Uschold e King [Uschold and King 1995], Metodologia Methontology [Fernández et al. 1997], Método 101 [Noy and McGuinness 2000], entre outros.

Uma análise comparativa entre as metodologias e métodos constatou que as mesmas mostram-se pouco eficientes na exposição clara dos procedimentos de construção [da Silva et al. 2008]. Segundo [Caliari 2007], é preciso simplificar a apresentação dos conceitos fornecendo meios que facilitem sua construção para permitir que os benefícios da representação do conhecimento em forma de ontologias sejam usufruídos por um número maior de pessoas.

Para isso, o método Deronto [Caliari 2007] tem como objetivo facilitar a construção de ontologias de domínio e de aplicação para profissionais da área de desenvolvimento de sistemas de informática com pouco ou moderado conhecimento em ontologias. O Deronto é um método iterativo, que contém seis passos os quais podem ser observados na Figura 5. Os números correspondem aos passos e as setas pontilhadas à documentação da ontologia.

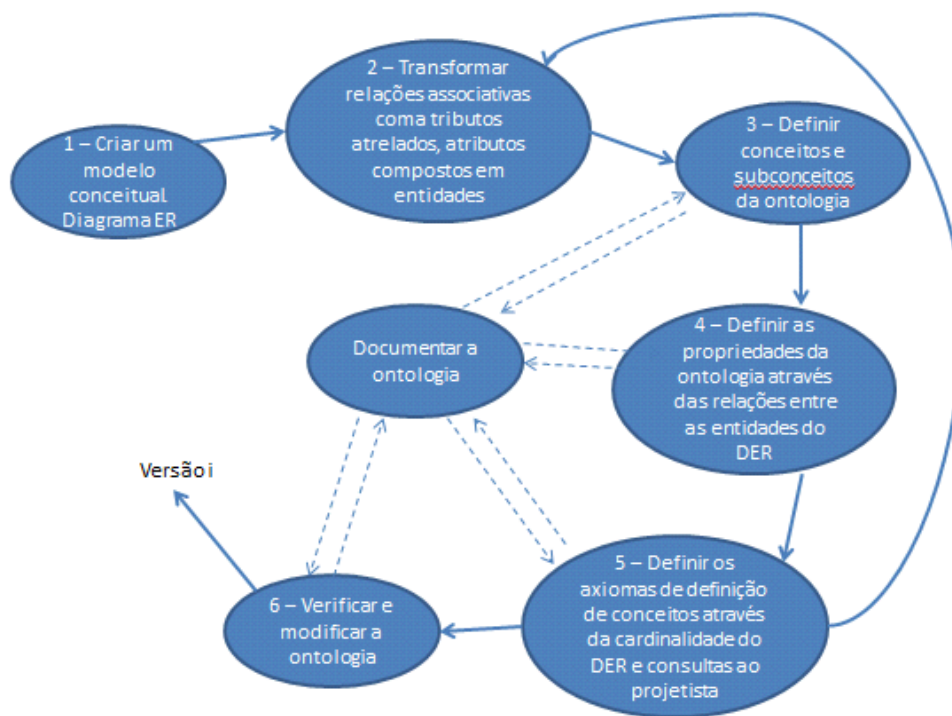


Figura 5. Método para definição de uma Ontologia a partir do DER [Caliari 2007]

De acordo com [Caliari 2007], no primeiro passo é desenvolvido, ou aproveitado, um DER do domínio de interesse. Para automatização dos passos seguintes, no passo 2 o DER é reescrito excluindo-se entidades e relações complexas, tais como, atributos compostos ou multivalorados e relações associativas. A definição da ontologia inicia-se no passo 3, no qual são definidos os conceitos e subconceitos da ontologia, por meio das entidades do DER. No passo 4, são definidas as propriedades a partir dos relacionamentos e atributos do DER. No passo 5 são definidos os axiomas de definição dos conceitos a partir da cardinalidade das relações. No último passo, a ontologia é verificada e modificada. A partir do passo 3 é realizada a documentação da ontologia e após o passo 6 tem-se a versão *i* da ontologia.

5. Trabalhos Relacionados

Trabalhos sobre desenvolvimento de ontologias de orçamento e especificamente para o setor público não são frequentes na literatura. Entretanto, existem alguns trabalhos semelhantes ao nosso. Podemos mencionar o trabalho [Brusa et al. 2006] para uma província argentina, o trabalho [SOF 2012] para o Orçamento Federal Brasileiro e a Ontologia “Payments” [Ontology 2010] desenvolvida dentro do contexto de dados abertos do governo britânico.

Em [Brusa et al. 2006] é descrita uma ontologia de Orçamento e informação financeira para a província de Santa Fe na Argentina, porém o principal objetivo do trabalho é apresentar uma abordagem sobre como técnicas da engenharia ontológica podem ser aplicadas para a solução de problemas de descoberta, agregação e compartilhamento de conteúdo em ambientes de e-government. A ontologia apresentada nesse trabalho se foca

na fase de formulação das despesas e não estão modeladas as receitas, nem conceitos relacionados a outras fases do processo de orçamento como a execução.

Assim como o trabalho acima, em [SOF 2012] limita-se a modelar as despesas e não estão modeladas as receitas. Portanto, não é possível verificar o empenho, liquidação e pagamento das despesas, apenas a previsão delas. Além disso, a informação temporal (ano) das despesas não foi representada, o que impossibilita uma melhor análise delas. Por último, não houve integração com o padrão Dublin Core [Initiative 2012] o que pode dificultar a sua integração com outras ontologias.

A ontologia “Payments” [Ontology 2010] foi desenvolvida como um vocabulário de propósito geral para representar informação de gastos organizacionais em geral e não se limita a aplicações governamentais.

6. Metodologia

Nesta seção é descrita a metodologia utilizada para as fases de conceitualização, definição e avaliação da ontologia, além da fase de desenvolvimento de uma aplicação teste.

Neste trabalho, foi utilizado como base para o desenvolvimento da ontologia o método Deronto. Este método foi escolhido, pois ele define claramente como obter cada conceito e propriedade de uma ontologia [Caliari 2007].

6.1. Conceitualização

O Deronto inicia-se com a aquisição ou desenvolvimento de um diagrama entidade-relacionamento (DER) do domínio. Assim, foi necessário, primeiramente, adquirir o conhecimento sobre o orçamento público para ser possível o desenvolvimento do DER. Para isso, foi utilizado como fonte principal de conhecimento o MTO 2012 [SOF 2011b]. Além dele, foram utilizados também o Ementário de Classificação das Receitas Orçamentárias [SOF 2011a] e outros livros da área de orçamento, contabilidade e finanças públicas [Giacomoni 2009] [Slomski 2008].

A fase de conceitualização da ontologia está detalhada na seção 7.1.

6.2. Definição

Após o desenvolvimento do DER, seguiu-se com a aplicação do método Deronto até obter-se uma primeira versão da ontologia. Essa versão foi então definida no editor de ontologias Protégé 4.1 [Protégé 2000]. Por ser um método iterativo, o processo foi repetido diversas vezes até que fosse obtida uma versão que representasse melhor o domínio. Neste momento, foram realizadas algumas mudanças na taxonomia das classes e nomenclatura das propriedades. Essas mudanças permitiram uma melhor organização dos conceitos e a redução de propriedades datatype, que descrevem relacionamentos entre recurso e tipo de dados, pois a mesma propriedade pôde ser aproveitada em uma grande quantidade de conceitos. Também foi possível reduzir a nomenclatura das propriedades object, que descrevem relacionamentos entre conceitos, deixando o modelo mais enxuto e claro. Para isso, foram utilizadas definições de classes, propriedades e regras para ontologias definidas no Protégé, descritas em [Horridge 2008]. Também houve um aproveitamento de propriedades nativas do RDF, RDFS e OWL, além da integração com o esquema de metadados Dublin Core (DC) [Initiative 2012]. A fase de definição da ontologia será detalhada na seção 7.2.

6.3. Validação

Depois de finalizada, a modelagem foi testada por meio do *Reasoner Pellet* [Sirin et al. 2007], o qual não encontrou nenhuma inconsistência no modelo. Depois foram inseridos alguns dados manualmente e criadas, por meio do Protégé, classes definidas com regras estabelecidas com *Semantic Web Rule Language* (SWRL) para realizar inferências e verificar se o domínio estava modelado corretamente. A fase de validação da ontologia está detalhada na seção 7.3.

6.4. Aplicação

Finalmente, era necessário demonstrar uma possibilidade de uso para a ontologia, ainda que simples, por meio de uma aplicação teste. Então, foram obtidos os dados da execução orçamentária das receitas e despesas de 2010 e 2011 retirados do Portal Siga Brasil [Federal 2012]. Mantido pelo Senado, foi escolhido como fonte por apresentar uma boa diversidade de informações referentes ao orçamento e também pela possibilidade de escolher quais variáveis serão consultadas. Foram obtidas milhares de ocorrências da execução de receitas e despesas nos anos de 2010 e 2011 no formato CSV que, posteriormente, foram convertidos para triplas RDF. Para manipulação dos dados do orçamento foi utilizado o *framework* Jena [Jena 2012], mais especificamente a API Jena TDB [Jena 2012], que juntamente com o *endpoint* Fuseki [Jena 2012], permitiram a utilização de *triple-stores* para persistência dos dados em disco e a criação de um servidor local no qual, por meio de uma interface Web, foi possível inserir os dados RDF serializados no formato *turtle* e também realizar consultas SPARQL nos mesmos. Na seção 8 são demonstrados alguns resultados da aplicação teste da ontologia.

7. Proposta da Ontologia

Nesta seção é apresentada a conceitualização, a definição e a validação da ontologia para o Orçamento Público Federal Brasileiro.

7.1. Conceitualização

Pelo fato do domínio em questão ser bastante complexo, houve a necessidade de desenvolver um DER básico e modificá-lo de acordo com a necessidade percebida. O diagrama foi dividido em três partes: Orçamento, Receitas e Despesas.

Uma entidade “ORCAMENTO” foi criada para relacionar as receitas e despesas de acordo com seu ano de exercício orçamentário. Desta forma, como exemplo de instância poderíamos ter o orçamento do ano de 2012, sendo o “ano” um atributo chave e “descricao” um atributo com informações básicas daquele orçamento. A entidade “ORCAMENTO” relaciona-se com as entidades “RECEITA” e “DESPESA”, ambos relacionamentos do tipo 1:n, ou seja, um orçamento pode ter de 0 a n receitas e/ou despesas. Uma instância de “RECEITA” ou “DESPESA” pertence a apenas um orçamento.

De acordo com [SOF 2011b], as receitas possuem três tipos de classificação: Classificação por Natureza, Classificação por Fonte/Destinação e Resultado Primário. Porém, em [SOF 2011a] também é descrita uma quarta, a Classificação por Grupos. Por entender que a mesma é importante para a observação das receitas, esta foi acrescentada à modelagem. Os termos Classificação por Natureza, Classificação por Fonte/Destinação e Classificação por Grupos foram ignorados na modelagem por serem apenas objetos de

organização dos componentes de classificação de uma receita, desta forma não houve a necessidade de representá-los.

Assim como nas Receitas, as Despesas têm uma modelagem parecida, porém com mais classificadores. A classificação por Fonte/Destinação é de extrema importância, pois representa um elo das receitas com despesas. Desta forma, tanto receitas como despesas possuem o relacionamento com a entidade “ESP_DA_FONTE_DESTINACAO” (Especificação da Fonte/Destinação). Esta entidade, por sua vez, representa um detalhamento de “GRUPO_DA_FONTE_DESTINACAO”.

Após a definição do DER, foi necessário simplificá-lo, pois em [Caliari 2007] recomenda-se que se tenha um diagrama o mais simples possível, ou seja, sem atributos multivalorados, sem atributos de relacionamentos, sem relacionamentos ternários, atributos com nome “tipo” também devem ser modificados, entre outros.

7.2. Definição da Ontologia

Nesta etapa, foram definidos os conceitos e subconceitos da ontologia por meio do DER do domínio. As entidades viram conceitos e as entidades especializadas viram subconceitos da ontologia. Por exemplo: a entidade “ACAO” torna-se o conceito “Acao”, as entidades “ATIVIDADE”, “OPERACAO_ESPECIAL” e “PROJETO” tornam-se os conceitos “Atividade”, “OperacaoEspecial” e “Projeto”, respectivamente.

Após definir os conceitos e subconceitos da ontologia, os relacionamentos do DER tornam-se propriedades do tipo object. As propriedades do tipo datatype são definidas por meio dos atributos do DER. A entidade “ORCAMENTO” tem os atributos “ano” e “descrição”. Estes atributos se transformam nas propriedades datatype “ano” e “descricao”, respectivamente. A propriedade “ano” tem como domínio o conceito “Orçamento” e imagem o tipo de dados “nonNegativeInteger”, pois os anos não podem assumir valores negativos e apenas inteiros. A propriedade “descricao” também tem como domínio o “Orçamento” e como imagem o tipo de dados string.

Os axiomas dos conceitos foram definidos por meio da cardinalidade dos relacionamentos do DER. Neste momento, podemos retirar do DER os axiomas de condição necessária e condição necessária e suficiente. Uma condição necessária determina quais propriedades uma instância daquele conceito precisa obrigatoriamente ter para pertencer àquela classe [Horridge 2008]. Isso não significa que todos os elementos que tem essas propriedades serão considerados daquela classe, pois são condições necessárias e não condições necessárias e suficientes.

Durante uma análise da ontologia obtida, observou-se que algumas modificações poderiam ser feitas para simplificá-la e termos um modelo mais enxuto. Percebeu-se que não havia a necessidade de nomear uma propriedade object com seu domínio, relacionamento e imagem. É possível utilizar somente o nome do relacionamento, permitindo o aproveitamento de propriedades em várias ocasiões sem definir o domínio e imagem da mesma. Por exemplo: há vários casos em que o relacionamento no DER se chama “possui”, é possível definir apenas uma propriedade object “possui” e utilizá-la em todos os casos sem definir o domínio e imagem da mesma, pois o domínio e imagem das propriedades são utilizados apenas para o *reasoner* definir subclasses por inferência [Horridge 2008]. Porém, para um domínio grande como o orçamento optou-se por manter a imagem na nomenclatura de algumas propriedades para efeitos de organização e também para ser

possível identificar o conceito ao qual pertence um indivíduo apenas olhando uma propriedade object que ele tenha. Além disso, aproveitar propriedades teria um benefício maior se houvesse a necessidade de termos a propriedade transitiva, o que não é o caso.

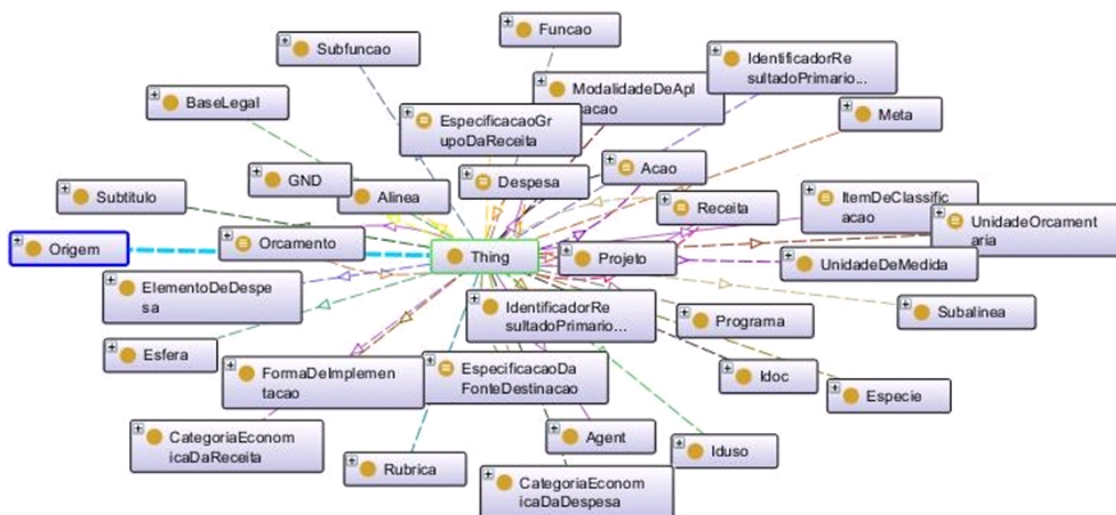


Figura 6. Gráfico com os conceitos e propriedades "Object" da ontologia definida no Protégé.

As propriedades datatype foram reduzidas consideravelmente, pois não era necessário manter, por exemplo, uma propriedade "nome" específica para cada conceito, optou-se então por definir apenas uma propriedade nome sem especificar o domínio dessa propriedade fazendo com que a mesma pudesse ser utilizada em todos os casos. O mesmo aconteceu com a propriedade "código".

Para usufruirmos dos benefícios da web semântica, necessitamos que as modelagens já utilizadas e tradicionais sejam reutilizadas. Por isso, devemos aproveitar ao máximo o que já foi modelado [Antoniou and Harmelen 2008]. Desta forma, a propriedade "nome" foi substituída pela annotation property "label". Essa propriedade é nativa do OWL, pois pertence ao rdfs, herdado pelo OWL, e pode ser usada tanto com instâncias, como classes e propriedades para definir, em linguagem natural, o nome do recurso. Além disso, para explicar cada conceito, foi utilizada a propriedade "comment", também do rdfs. Para determinar o criador da ontologia foi utilizada a propriedade "creator", do Dublin Core. Essa propriedade permite o relacionamento com um elemento do tipo "Agent", no qual é possível colocar informações adicionais a respeito do indivíduo que definiu a ontologia. Além do "label" do rdfs, foi utilizada também a propriedade "title", do Dublin Core, para determinar o nome de cada instância. Isso foi feito por não ser possível colocar uma annotation property, no caso a "label", como obrigatória além de permitir que futuras aplicações possam escolher se utilizarão a annotation property "rdfs:label" ou a datatype property "dc:title" como referência ao nome dos recursos. Para um melhor entendimento, os conceitos da ontologia foram separados em quatro tipos principais: "Orçamento", "Despesa", "Receita" e "ItemDeClassificacao".

As instâncias do "Orçamento" correspondem a LOA de cada ano, com a elaboração e execução orçamentária.

O conceito “ItemDeClassificacao” é uma superclasse de todos os conceitos que classificam as despesas e receitas, sejam eles pertencentes à classificação por natureza das receitas ou das despesas, classificação programática das despesas, entre outros. Assim, fica bem definido o que é receita, o que é despesa e o que é utilizado para organizar as informações sobre as despesas e receitas do orçamento de um determinado ano.

As instâncias de “Despesa” correspondem aos itens de despesa de um determinado “Orçamento”. Cada instância de Despesa tem as propriedades relacionadas a valores de “ItemDeClassificacao” que são utilizados para classificar uma despesa. O mesmo ocorre com a “Receita” e os elementos de “ItemDeClassificacao” que são utilizados para classificar uma receita.

Essa reorganização da taxonomia dos conceitos também foi realizada com as propriedades. Na Figura 6 os retângulos representam cada classe, ou conceito, e as setas representam as relações entre elas. Na Figura 7, temos uma parte da reestruturação dos itens de classificação, os retângulos representam novamente as classes e as setas representam a hierarquia de classes, por exemplo, “Meta” é subclasse de “ItemDeClassificação”.

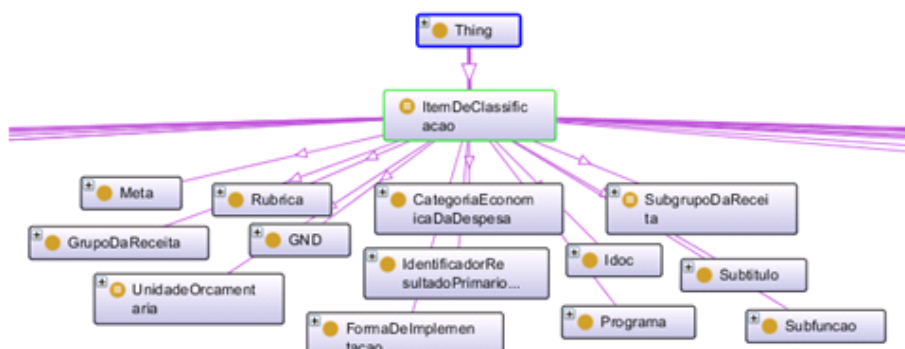


Figura 7. Parte da reestruturação dos itens de classificação.

7.3. Validação da Ontologia

Terminada a definição do modelo, foram inseridos manualmente instâncias de todos os conceitos para serem realizados alguns testes. Primeiramente, o *reasoner* Pellet [Sirin et al. 2007] foi executado e não foi encontrada nenhuma inconsistência no modelo. Após esse primeiro teste, foram criadas algumas subclasses de “Receita” e “Despesa” para separar automaticamente os dados da execução de receitas e despesas por anos e por destinação. A seguir, serão demonstrados dois exemplos de inferência que podem ser realizados na ontologia do orçamento: um com criação de uma classe definida na qual seus membros serão incluídos pelo processo de inferência, e a outra, a criação de uma propriedade inversa que trará mais informações sem a necessidade de inserir ou alterar dados no *triple-store*.

Exemplo 1: Com o intuito de obter todas as receitas do ano de 2011 sem fazer nenhuma consulta, foi criada uma subclasse de “Receita”, a “Receita2011”. Automaticamente, as definições de regras da superclasse “Receita” são herdadas para a subclasse “Receita2011”. Para definir esse tipo de regra foi utilizada a linguagem SWRL. Assim, ao realizar as inferências, todas as instâncias de “Receita” que estão relacionadas com o orçamento de 2011 pertencerão também à classe “Receita2011”. A criação de classes

com regras para inferências poderiam abranger também diversos outros escopos além da divisão de receitas por ano e destinação, como por exemplo, despesas por fontes, despesas por função, despesas por órgão orçamentário, despesas por subtítulo (localização do gasto), entre outras possibilidades. Na Figura 8 há exemplos de regras SWRL criadas.

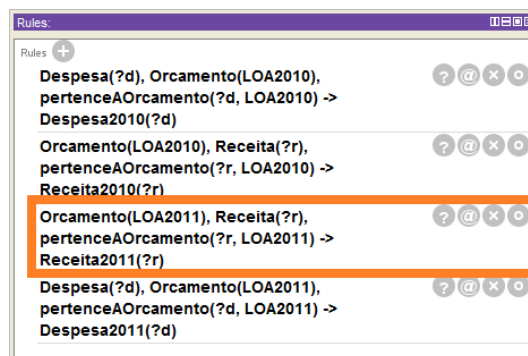


Figura 8. Regra SWRL que define quais receitas pertencerá a “RECEITA2011”

Na Figura 9, temos o resultado da inferência que determina a qual ano pertence as intâncias de determinadas receitas.

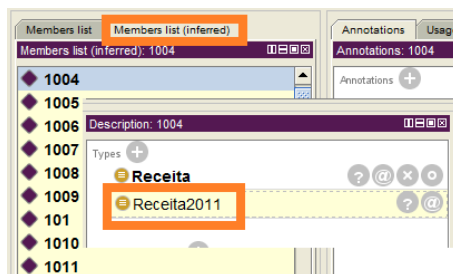


Figura 9. Indivíduos determinados a uma classe por inferência

Exemplo 2: as inferências podem trazer outros benefícios aos dados estruturados na forma de uma ontologia. Temos a propriedade object “pertenceAOrcamento”, cujo domínio é uma instância de “Receita” e a imagem é uma instância de “Orcamento”. Porém, não é possível saber quais receitas pertencem ao orçamento “x” sem realizar uma consulta para obter os dados. É possível somente verificar em cada receita a qual orçamento ela pertence. Com a criação de uma propriedade inversa de “pertenceAOrcamento”, seu domínio será “Orcamento” e sua imagem, “Receita”. Criou-se então a propriedade “orcamentoTem”, inversa à propriedade “pertenceAOrcamento” que pode relacionar o orçamento com receitas e despesas. Desta forma, os relacionamentos com essa propriedade serão feitos automaticamente por meio da inferência do raciocinador evitando consultas ou modificação das triplas. A Figura 10 demonstra os efeitos da criação da propriedade orcamentoTem.

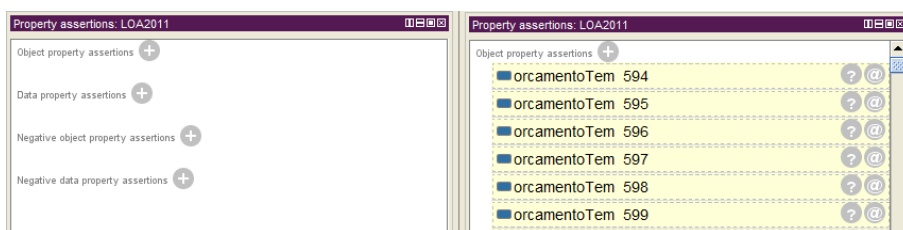


Figura 10. Relações do orçamento de 2011 antes (esquerda) e depois (direita) da propriedade "orcamentoTem"

Para inserir instâncias reais, com receitas e despesas do orçamento federal, foram extraídos os dados fornecidos pelo portal Siga Brasil [Federal 2012]. Nele é possível obter informações da elaboração e execução de receitas e despesas de diversos anos com consultas já formatadas, mas também é possível realizar sua própria consulta, caso seja necessários dados mais detalhados ou específicos.

Os dados extraídos em formato CSV foram convertidos para triplas RDF e serializados no formato *turtle*. A serialização em *Turtle* permitiu uma rápida inserção dos dados no *triple-store* gerado pela API do Jena TDB por meio da interface do servidor Fuseki. Nela é possível escolher um arquivo com dados serializados em *turtle*, entre outros formatos, e inserir as triplas na base de dados rapidamente.

Para testar a ontologia, foram realizadas consultas para obter as receitas ou despesas de cada ano, a soma dos valores, entre outros. Também é possível obter valores de despesas em cada função ou subfunção, em cada órgão ou unidade orçamentária. Esse tipo de consulta permite obter dados de anos diferentes, em áreas diferentes, possibilitando que aplicações desenvolvam meios de comparar essas informações gerando uma ferramenta extremamente eficiente de comparação e avaliação de governos.

Na Figura 11 temos um exemplo de consulta SPARQL para obter o valor líquido arrecadado de cada receita do ano de 2011.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX bra: <http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl#>
PREFIX orc: <http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Orcamento/>

SELECT ?receita ?valor
WHERE { ?receita bra:pertenceAOrcamento orc:LOA2011 ;
        bra:valorLiquidoArrecadado ?valor }
```

SPARQLer Query Results

receita	valor
<http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Receita/1>	"-12763821.01"
<http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Receita/10>	"-124773.76"
<http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Receita/100>	"5.2"
<http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Receita/1000>	"41398652.82"
<http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Receita/1001>	"42467653.69"
<http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Receita/1002>	"43109517.25"

Figura 11. Simples consulta para obter as receitas do ano 2011

A consulta na Figura 12, um pouco mais elaborada, permite extrair a soma dos valores pagos das despesas que atuaram em cada função do governo no ano de 2010.

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX owl: <http://www.w3.org/2002/07/owl#>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5 PREFIX bra: <http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/>
6 prefix desp: <http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Despesa/>
7 prefix orc: <http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Orcamento/>
8 prefix funcao: <http://www.semanticweb.org/ontologies/OrcamentoPublicoBrasileiro.owl/Funcao/>
9
10 SELECT (SUM(?valor) AS ?soma)
11     WHERE { ?despesa bra:pertenceAOrcamento orc:LOA2010 .
12     ?despesa bra:atuaNaFuncao ?funcao .
13     ?despesa bra:valorPago ?valor} GROUP BY ?funcao

```

Figura 12. Consulta SPARQL que retorna a soma das despesas em cada função no ano de 2010

8. Proposta de Uso para a Ontologia

Para demonstrar uma aplicação teste da ontologia foram realizadas comparações entre receitas e despesas da execução orçamentária dos anos de 2010 e 2011. Selecionou-se arbitrariamente a subfunção “Ensino Superior” e decidiu-se por obter as informações sobre quais receitas financiaram os gastos realizados nessas subfunções nesses dois anos. Para realizar essa aplicação foi necessário utilizar os recursos demonstrados na Figura 13. Como a classificação por fonte/destinação (representada na figura 13 pela classe “EspecificacaoFonteDestinacao”) funciona como um elo entre receitas e despesas, foi necessário verificar, primeiramente, quais despesas foram realizadas na subfunção em questão em cada ano. Então foram selecionadas as fontes de financiamento dessas despesas para então ser possível a verificação de quais naturezas de receitas (representadas na figura 13 pelas classes “CategoriaEconomicaDaReceita”, “Origem”, “Especie”, “Rubrica”, “Alinea” e “Subalinea”, que compõem a natureza da receita) financiaram essas despesas nos dois anos.

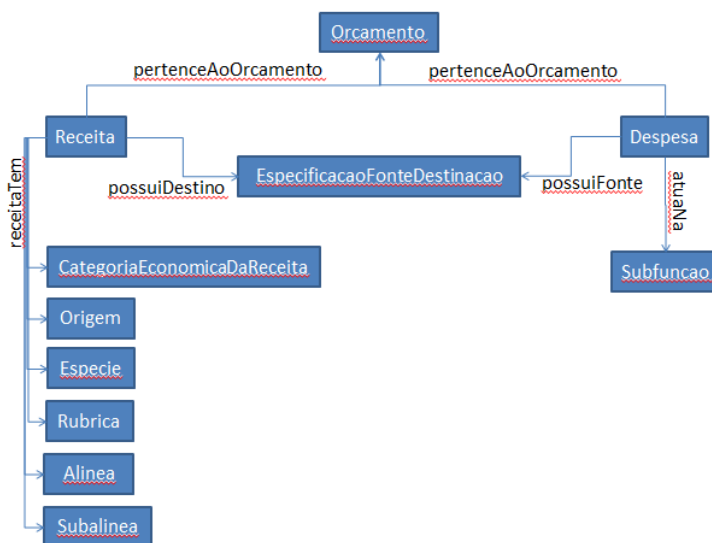


Figura 13. Recursos utilizados para a aplicação.

Verificou-se então que, no ano de 2010 os recursos para as despesas que atuaram na subfunção “Ensino Superior” foram compostos por 12 fontes, sendo a principal fonte “Recursos destinados à manutenção do Ensino Superior” contribuindo com mais de 78

No ano de 2011 os resultados obtidos foram parecidos, os recursos para as despesas que atuaram na subfunção “Ensino Superior” foram compostos por 15 fontes, sendo que a principal fonte também foi “Recursos destinados à manutenção do Ensino Superior” contribuindo com mais de 77

Essa aplicação teste não expressa todo o potencial da ontologia. Regras mais elaboradas estão sendo discutidas e devem ser implementadas num futuro próximo.

9. Conclusões e Diretrizes para Trabalhos Futuros

Como esse trabalho discutiu, existem poucos trabalhos que propõe ontologias de orçamentos públicos e nenhum deles apresentam uma modelagem tão completa quanto à proposta apresentada nesse artigo. A ontologia resultante do presente trabalho oferece a modelagem das despesas e também a das receitas. Além disso, também é possível manipular dados de todas as etapas orçamentárias, não apenas da previsão. Outro aspecto importante, é que com esta ontologia é possível à manipulação de dados do orçamento de diversos anos.

A partir dos resultados apresentados, conclui-se que o presente trabalho atingiu o objetivo de definir uma ontologia para o Orçamento Público Federal, utilizando o método de desenvolvimento Deronto e validando-a por meio da demonstração de inferências e consultas para obtenção de informações sobre receitas e despesas do governo federal no decorrer dos anos.

Os resultados obtidos foram satisfatórios, já que possibilitam a obtenção de valores específicos e a comparação entre os anos. Destaca-se a necessidade da conversão dos dados de outros períodos para RDF, a implementação de aplicações com regras mais elaboradas e a definição de outras ontologias para ser possível a ligação desses dados e a interoperabilidade dos mesmos, como por exemplo, uma ontologia para os dados dos fornecedores do governo, uma ontologia que defina a organização institucional do governo, entre outras. Além disso, é importante que o modelo seja aplicado também com dados do poder executivo nos níveis estadual e municipal.

Observando-se os dados da execução orçamentária retirados do Portal Siga Brasil, foi possível notar algumas informações não obrigatórias pela lei, mas divulgadas nas receitas e despesas. Desta forma, uma possível extensão do modelo levaria a acrescentar alguns conceitos como, por exemplo, um melhor detalhamento das localidades dos gastos.

Seria também de grande valia a inclusão de informações a respeito dos beneficiários, empresas e pessoas físicas que receberam pagamentos do governo, pois essa informação facilitará o confronto com as informações divulgadas pelos fornecedores. Para isso, é possível integrar a ontologia do orçamento com a Payments Ontology [Ontology 2010], devidamente traduzida, que modela um cenário geral de pagamentos.

Os resultados obtidos nesse trabalho também apontam na direção de uma melhor forma de sistematização e modelagem do conhecimento de um domínio. Os passos aqui colocados poderiam ser aplicados também na construção de ontologias de orçamentos públicos de outros países, buscando dessa forma também colaborar nos esforços de

padronização da transparência fiscal em nível global. Além de verificar de estruturas comuns, possíveis mapeamentos poderiam ser levantados.

Espera-se que, num futuro próximo, os dados abertos governamentais sejam totalmente ligados e estruturados com o auxílio da Web Semântica, possibilitando, de fato, o desenvolvimento de aplicações que deem ao cidadão um melhor poder de fiscalização e avaliação dos governos passados, presentes e futuros.

Referências

- Allemang, D. and Hendler, J. (2011). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS*. Elsevier, second edition.
- Antoniou, G. and Harmelen, F. (2008). *A Semantic Web Primer*. The MIT Press, second edition.
- Brasil (1964). Lei nº 4.320, de 17 de março de 1964. http://www.planalto.gov.br/ccivil_03/leis/L4320.htm. [Online; accessed 20-Jan-2012].
- Brasil (1988). Constituição da república federativa do brasil. http://www.planalto.gov.br/ccivil_03/Constituicao/Constitui [Online; accessed 10-Mar-2012].
- Brasil (2000). Lei complementar nº 101, de 4 de maio de 2000. http://www.planalto.gov.br/ccivil_03/leis/LCP/Lcp101.htm. [Online; accessed 20-Jan-2012].
- Brasil (2009). Lei complementar nº 131, de 27 de maio de 2009. http://www.planalto.gov.br/ccivil_03/leis/lcp/lcp131.htm. [Online; accessed 15-Mai-2012].
- Brasil (2011). Lei nº 12.527, de 18 de novembro de 2011. http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm. [Online; accessed 15-Mai-2012].
- Bratt, S. (2006). Semantic web and other w3c technologies to watch. [Online; accessed 09-Sep-2012].
- Brusa, G., Caliusco, L., and Chiotti, O. (2006). A process for building a domain ontology: an experience in developing a government budgetary ontology. http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm. [Online; accessed 10-Mar-2012].
- Caliari, F. M. (2007). Deronto: Método para construção de ontologias a partir de diagramas entidade-relacionamento.
- da Silva, D. L., Souza, R. R., and Almeida, M. B. (2008). Ontologias e vocabulários controlados: comparação de metodologias para construção. http://www9.senado.gov.br/portal/page/portal/orcamento_senado/SigaBrasil. [Online; accessed 20-Ago-2012].
- da Silva Craveiro, G., de Santana, M. T., and ao Porto de Albuquerque, J. (2012). Assessing open government budgetary data in brazil.

- Evangelista, L. (2010). Controle social versus transparência pública: uma questão de cidadania. <http://portal2.tcu.gov.br/portal/pls/portal/docs/2053966.PDF>. [Online; accessed 07-Nov-2012].
- Federal, S. (2012). Portal siga brasil. www9.senado.gov.br/portal/page/portal/.../SigaBrasil. [Online; accessed 20-Ago-2012].
- Fernández, M., Gómez-Pérez, A., and Jurino, N. (1997). Methontology: From ontological art towards ontological engineering. *AAAI-Spring Symposium on Ontological Engineering*.
- for Fiscal Transparency, G. I. (2012). <http://fiscaltransparency.net/>. [Online; accessed 07-Nov-2012].
- Giacomoni, J. (2009). *Orçamento Público*. Atlas, 14ª edition.
- Gruninger, M. and Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. *Workshop on Basic Ontological Issues In Knowledge Sharing*.
- Horridge, M. (2008). *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. [Online; accessed 10-Apr-2012].
- Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosz, B., and Dean, M. (2004). Swrl: A semantic web rule language combining owl and ruleml. W3c member submission, World Wide Web Consortium.
- Initiative, D. C. M. (2012). Dublin core. <http://dublincore.org/>. [Online; accessed 10-Jul-2012].
- Jena, A. (2012). The jena framework. <http://jena.apache.org/>. [Online; accessed 30-Ago-2012].
- Júnior, H. S. R. and Rover, A. J. (2009). Ontologias aplicadas ao direito e ao governo eletrônico: experiência e projetos. In *Alagoas Digital: Inovações Acadêmicas e Projetos de Software Livre*. [Online; accessed 20-Jan-2012].
- Noy, N. F. and McGuinness, D. L. (2000). *Ontology development 101: A guide to creating your first ontology*.
- Ontology, P. (2010). Guide to the payments ontology. <http://data.gov.uk/resources/payments#structure>. [Online; accessed 20-Feb-2012].
- OpenSpending (2012). A data standard for transaction-level spending data. <http://openspending.org/resources/standard/index.html>. [Online; accessed 07-Nov-2012].
- Paludo, A. (2011). *Orçamento Público, Administração Financeira e Orçamentária e Lei de Responsabilidade Fiscal*. Elsevier, Rio de Janeiro.
- Partnership, O. G. (2012). <http://www.opengovpartnership.org/>. [Online; accessed 07-Nov-2012].
- Protégé (2000). Protégé. <http://protege.stanford.edu/>. [Online; accessed 10-Feb-2012].
- Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical owl-dl reasoner. <http://pellet.owldl.com/papers/sirin05pellet.pdf>. [Online; accessed 20-Sep-2012].

Slomski, V. (2008). *Manual de Contabilidade Pública: um enfoque na contabilidade municipal, de acordo com a Lei de Responsabilidade Fiscal*. Atlas, São Paulo, second edition.

SOF (2011a). *Ementário de Classificação das Receitas Orçamentárias da União*. Brasília. [Online; accessed 20-Jan-2012].

SOF (2011b). *Manual Técnico de Orçamento 2012 - MTO*. Brasília. [Online; accessed 20-Jan-2012].

SOF (2012). *Orçamento Federal de 2012 em Formato Aberto – Manual de Referência*. [Online; accessed 20-Oct-2012].

Studer, R., Benjamins, R., and Fensel, D. (2008). Knowledge engineering: Principles and methods. data and engineering.

Uschold, M. and King (1995). Building ontologies: Towards a unified methodology. *16th annual conf. of the british computer society specialist group on expert systems*.

W3C (2008). Sparql query language for rdf.

Apêndice A. Resumo do DER

Tabela 3. Entidades e seus Relacionamentos

ENTIDADES DO DER COM SEUS RESPECTIVOS RELACIONAMENTOS						
ENTIDADE	CONCEITO (MTO 2012)	RELACIONAMENTO	ENTIDADE	CONCEITO (MTO 2012)		
ORÇAMENTO	Lei de Orçamento Anual	PO SBUI	RECEITA	Receita		
		PO SBUI	DESPESA	Despesa		
RECEITA	Receita	PO SBUI	CATEGORIA_ECONOMICA_DA_RECEITA	Categoria Econômica		
		PO SBUI	ORIGEM	Origem		
		PO SBUI	ESPECIE	Espécie		
		PO SBUI	ALINEA	Alínea		
		PO SBUI	SUBALINEA	Subalínea		
		PO SBUI	ID_RESULTADO_PRIMARIO_RECEITA	Identificador do Resultado Primário de Receita		
		PO SBUI	ESP_GRUPO_DA_RECEITA	Especificação do Grupo de Receita *		
		PO SBUI	ESP_DA FONTE_DESTINACAO	Especificação de Fonte / Destinação		
		PO SBUI	CATEGORIA_ECONOMICA_DA_DESPESA	Categoria Econômica		
		PO SBUI	GRUPO	Grupo de Natureza de Despesa		
DESPESA	Despesa	PO SBUI	MODALIDADE DE APLICACAO	Modalidade de Aplicação		
		PO SBUI	ELEMENTO DE DESPESA	Elemento de despesa		
		PO SBUI	ID_RESULTADO_PRIMARIO_DESPESA	Identificador do Resultado Primário de Despesa		
		PERTENCA	ESFERA	Esfere		
		REALIZADOPOR	UNIDADE_ORÇAMENTARIA	Unidade Orçamentária		
		ATUANA	FUNCAO	Função		
		ATUANA	SUBFUNCAO	Subfunção		
		PO SBUI	IDUSO	Identificador de Uso		
		PO SBUI	IDOC	Idoc		
		PO SBUI	SUBTITULO	Subtítulo		
		PO SBUI	ACAO	Ação		
		PERTENCAO	PROGRAMA	Programa		
		PO SBUI	ESP_DA FONTE_DESTINACAO	Especificação de Fonte / Destinação		
		ESP_GRUPO_DA_RECEITA	Especificação do Grupo de Receita *	DETALHA	SUBGRUPO_DA_RECEITA	Subgrupo de Receita *
		SUBGRUPO_DA_RECEITA	Subgrupo de Receita *	DETALHA	GRUPO_DA_RECEITA	Grupo de Receita *
		ESP_DA FONTE_DESTINACAO	Especificação de Fonte / Destinação	DETALHA	GRUPO_DA FONTE_DESTINACAO	Grupo de Fonte / Destinação
UNIDADE_ORÇAMENTARIA	Unidade Orçamentária	COMPOE	ORGAO	Órgão Orçamentário		
ACAO	Ação	PO SBUI	UNIDADE DE MEDIDA	Unidade de Medida de Ação		
		PO SBUI	FORMA_IMPL	Forma de Implementação de Ação		
		PO SBUI	BASE_LEGAL	Base Legal		
		PO SBUI	META	Meta de Ação (produto)		
		IMPACTADAPOR	PROJETO	Projeto		
		COMPOSTAPOR	ETAPA	Etapas de Ação		

* Presente apenas no ementário das receitas de 2012.

Tabela 4. Entidades Generalizadas/Especializadas

GENERALIZAÇÃO/ESPECIALIZAÇÃO			
ENTIDADE GENERALIZADA	CONCEITO (MTO 2012)	ENTIDADE ESPECIALIZADA	CONCEITO (MTO 2012)
AÇÃO	Ação	PROJETO	Projeto
		OPERAÇÃO_ESPECIAL	Operação Especial
		ATIVIDADE	Atividade
PROGRAMA	Programa	TEMÁTICO	Temático
		GMS	Gestão, Manutenção e Serviços ao Estado

Tabela 5. Entidades e seus Atributos

ENTIDADES DO DER COM SEUS RESPECTIVOS ATRIBUTOS			
ENTIDADE	CONCEITO (MTO 2012)	ATRIBUTOS	CONCEITO (MTO 2012)
ENTIDADES PRIMO PAIS			
ORCAMENTO	Lei de Orçamento Anual	ano	Ano de correspondência do orçamento
		descricao	
RECEITA	Receita	ID	
		status	A qual etapa do processo orçamentário pertence
		data	Data de atualização do dado
		valor	Valor atribuído à Receita
DESPESA	Despesa	ID	
		status	A qual etapa do processo orçamentário pertence
		data	Data de atualização do dado
		valor	Valor atribuído à Despesa
		meta_fisica	Quantidade do produto a ser entregue
ENTIDADES UTILIZADAS PARA CLASSIFICAR AS DESPESAS E/OU RECEITAS			
CATEGORIA_ECONOMICA_DA_RECEITA	Categoria Econômica	ID	
ORIGEM	Origem	label	Nome atribuído
ESPECIE	Especie	codigo	Código atribuído
ALINEA	Alinea		
SUBALINEA	Subalinea		
ESP_GRUPO_DA_RECEITA	Especificação do Grupo da Receita		
ESP_DA FONTE_DESTINACAO	Especificação da Fonte /Destinação		
CATEGORIA_ECONOMICA_DA_DESPESA	Categoria Econômica	Nesses casos é necessário um identificador à parte, pois os códigos não são únicos e dependem de outros itens de classificação	
GND	Grupo de Natureza de Despesa		
MODALIDADE_DE_APLICACAO	Modalidade de Aplicação		
SUBGRUPO_DA_RECEITA	Subgrupo da Receita		
ELEMENTO_DE_DESPESA	Elemento de despesa		
ID_RESULTADO_PRIMARIO_DESPESA	Identificador do Resultado Primário da Despesa		
ID_RESULTADO_PRIMARIO_RECEITA	Identificador do Resultado Primário da Receita		
ESFERA	Esfera		
UNIDADE_ORCAMENTARIA	Unidade Orçamentária		
FUNCAO	Função		
SUBFUNCAO	Subfunção		
IDUSO	Identificador de Uso		
IDOC	Idoc		
GRUPO_DA_RECEITA	Grupo da Receita	codigo	Código atribuído
GRUPO_DA FONTE_DESTINACAO	Grupo da Fonte /Destinação	label	Nome atribuído
ORGAO	Órgão Orçamentário		
SUBTITULO	Subtítulo		
UNIDADE_DE_MEDIDA	Unidade de Medida da Ação		
FORMA_IMPL	Forma de Implementação da Ação		
BASE_LEGAL	Base Legal		
META	Meta da Ação		
PROGRAMA	Programa		
TEMATICO	Programa Temático		
GVS	Programa de Gestão, Manutenção e Serviços ao Estado		
ACAO	Ação	codigo	Código atribuído
		label	Nome atribuído
		det_impl	Detalhamento da implementação
		descricao	Descrição da ação
		finalidade	Finalidade da Ação
		objetivo	Objetivo da Ação
PROJETO	Ação do tipo Projeto	codigo	Código atribuído
		label	Nome atribuído
		duracao	Duração do Projeto
		custo_est	Custo total estimado/Gasto
		total_fisico	Total físico a ser realizado
ETAPA	Etapa do Projeto	ID	
		label	Nome atribuído
		detalhamento	Detalhamento da etapa
		numero	Ordem na execução
		cronograma	Cronograma da Etapa
		valor	Valor Estimado/Gasto

Apêndice B. Documentação dos Conceitos da Ontologia

Tabela 6. Documentação dos Conceitos da Ontologia

Documentação dos Conceitos da Ontologia	
Nome:	Orçamento
Descrição:	Programação orçamentária de um determinado ano. Possui instâncias de cada ano para a elaboração e execução orçamentária.
Sinônimo:	LOA
SubClass Of:	Thing
Nome:	Receita
Descrição:	Classe na qual os usuários irão inserir os dados de cada Receita. Disponibilidades de recursos financeiros que ingressam durante o exercício e constituem elemento novo para o patrimônio público. Instrumento por meio do qual se viabiliza a execução das políticas públicas, a receita orçamentária é fonte de recursos utilizada pelo Estado em programas e ações cuja finalidade precípua é atender às necessidades públicas e demandas da sociedade.
Sinônimo:	Receita Orçamentária
SubClass Of:	Thing
Nome:	Despesa
Descrição:	Classe na qual o usuário irá inserir as despesas do governo. Uma instância de Despesa pertence a um orçamento, possui uma Ação, um Idoc, uma Esfera, uma Especificação da Fonte, um Identificador de Resultado Primário e uma Classificação da Natureza da Despesa.
Sinônimo:	Despesa Orçamentária
SubClass Of:	Thing
Nome:	CategoriaEconomicDaReceita
Descrição:	Nível de menor detalhe da classificação por natureza de uma receita. Representa o primeiro dígito da classificação por natureza.
Sinônimo:	Categoria Econômica da Receita
SubClass Of:	ItemDeClassificacao
Nome:	Origem
Descrição:	A origem é o detalhamento das categorias econômicas Receitas Correntes e Receitas de Capital, com vistas a identificar a procedência das receitas no momento em que ingressam nos cofres públicos. Representa o segundo dígito da Classificação por Natureza da Receita.
Sinônimo:	
SubClass Of:	ItemDeClassificacao
Nome:	ItemDeClassificacao
Descrição:	Superclasse utilizada apenas para efeitos taxonômicos. São os itens que são utilizados para detalhar e classificar receitas e despesas do orçamento.
Sinônimo:	Item de Classificação Orçamentária
SubClass Of:	Thing
Nome:	Especie
Descrição:	A espécie, nível de classificação vinculado à origem, permite qualificar com maior detalhe o fato gerador das receitas. Por exemplo, dentro da origem Receita Tributária, identificam-se as espécies Impostos, Taxas e Contribuição de Melhoria. Representa o 3º dígito da Natureza da Receita.
Sinônimo:	Espécie
SubClass Of:	ItemDeClassificacao

Nome:	Subalínea
Descrição:	A subalínea constitui o nível mais analítico da receita, utilizado quando há necessidade de se detalhar a alínea com maior especificidade. Por exemplo, a subalínea Pessoas Físicas corresponde ao detalhamento da alínea Imposto sobre a Renda e Proventos de Qualquer Natureza. Corresponde ao 7º e 8º dígitos da Natureza da Receita.
Sinônimo:	Subalínea
SubClass Of:	ItemDeClassificacao
Nome:	IdentificadorResultadoPrimarioReceita
Descrição:	Conforme esta classificação, as receitas do Governo Federal podem ser divididas em: a) primárias (P), quando seus valores são incluídos na apuração do resultado primário (diferença entre as receitas primárias e as despesas primárias); e b) não primárias ou financeiras (F), quando não são incluídas nesse cálculo.
Sinônimo:	ID Primário da Receita
SubClass Of:	ItemDeClassificacao
Nome:	GrupoDaReceita
Descrição:	Classifica a receita orçamentária em grupos.
Sinônimo:	Grupo da Receita
SubClass Of:	ItemDeClassificacao
Nome:	EspecificacaoGrupoDaReceita
Descrição:	Detalha o subgrupo da receita.
Sinônimo:	Especificação do Grupo da Receita, Classificação da Receita por Grupos
SubClass Of:	ItemDeClassificacao
Nome:	CategoriaEconomicadaDespesa
Descrição:	A despesa, assim como a receita, é classificada em duas categorias econômicas, com os seguintes códigos: 3 – Despesas Correntes: as que não contribuem, diretamente, para a formação ou aquisição de um bem de capital; 4 – Despesas de Capital: as que contribuem, diretamente, para a formação ou aquisição de um bem de capital.
Sinônimo:	Categoria Econômica da Despesa
SubClass Of:	ItemDeClassificacao
Nome:	GND
Descrição:	O GND é um agregador de elemento de despesa com as mesmas características quanto ao objeto de gasto.
Sinônimo:	Grupo de Natureza de Despesa
SubClass Of:	ItemDeClassificacao
Nome:	ModalidadeDeAplicacao
Descrição:	A modalidade de aplicação indica se os recursos serão aplicados mediante transferência financeira, inclusive a decorrente de descentralização orçamentária para outros níveis de Governo, seus órgãos ou entidades, ou diretamente para entidades privadas sem fins lucrativos e outras instituições; ou, então, diretamente pela unidade detentora do crédito orçamentário, ou por outro órgão ou entidade no âmbito do mesmo nível de Governo.
Sinônimo:	Modalidade de Aplicação
SubClass Of:	ItemDeClassificacao
Nome:	ElementoDeDespesa
Descrição:	O elemento de despesa tem por finalidade identificar os objetos de gasto, tais como vencimentos e vantagens fixas, juros, diárias, material de consumo, serviços de terceiros prestados sob qualquer forma, subvenções sociais, obras e instalações, e equipamentos e material permanente, auxílios, amortização e outros que a Administração Pública utiliza para a consecução de seus fins.
Sinônimo:	Elemento de Despesa
SubClass Of:	ItemDeClassificacao

Nome:	IdentificadorResultadoPrimarioDespesa
Descrição:	O identificador de resultado primário, de caráter indicativo, tem como finalidade auxiliar a apuração do resultado primário previsto na LDO, devendo constar no PLOA e na respectiva Lei em todos os GNDs, identificando, de acordo com a metodologia de cálculo das necessidades de financiamento, cujo demonstrativo constará em anexo à LOA. De acordo com o estabelecido no § 5o do art. 7o do PLDO 2012, nenhuma ação poderá conter, simultaneamente, dotações destinadas a despesas financeiras e primárias, ressalvada a reserva de contingência.
Sinônimo:	ID de Resultado Primário
SubClass Of:	ItemDeClassificacao
Nome:	Esfera
Descrição:	Na LOA, a esfera tem por finalidade identificar se a despesa pertence ao Orçamento Fiscal (F), da Seguridade Social (S) ou de Investimento das Empresas Estatais (I), conforme disposto no § 5o do art. 165 da CF.
Sinônimo:	
SubClass Of:	ItemDeClassificacao
Nome:	UnidadeOrçamentaria
Descrição:	Na LOA, a esfera tem por finalidade identificar se a despesa pertence ao Orçamento Fiscal (F), da Seguridade Social (S) ou de Investimento das Empresas Estatais (I), conforme disposto no § 5o do art. 165 da CF.
Sinônimo:	Unidade Orçamentária, UD
SubClass Of:	ItemDeClassificacao
Nome:	Funcao
Descrição:	A função pode ser traduzida como o maior nível de agregação das diversas áreas de atuação do setor público. Reflete a competência institucional do órgão, como, por exemplo, cultura, educação, saúde, defesa, que guarda relação com os respectivos Ministérios.
Sinônimo:	Função
SubClass Of:	ItemDeClassificacao
Nome:	Subfuncao
Descrição:	A subfunção representa um nível de agregação imediatamente inferior à função e deve evidenciar cada área da atuação governamental, por intermédio da identificação da natureza das ações. As subfunções podem ser combinadas com funções diferentes daquelas relacionadas na Portaria MOG no 42, de 1999.
Sinônimo:	Subfunção
SubClass Of:	ItemDeClassificacao
Nome:	Iduso
Descrição:	Esse código vem completar a informação concernente à aplicação dos recursos e destina-se a indicar se os recursos compõem contrapartida nacional de empréstimos ou de doações ou destinam-se a outras aplicações, constando da LOA e de seus créditos adicionais. Conforme § 11 do art. 7o do PLDO 2012.
Sinônimo:	Identificador de Uso
SubClass Of:	ItemDeClassificacao

Nome:	Idoc
Descrição:	O IDOC identifica as doações de entidades internacionais ou operações de crédito contratuais alocadas nas ações orçamentárias, com ou sem contrapartida de recursos da União. Os gastos referentes à contrapartida de empréstimos serão programados com o IDUSO igual a "1", "2", "3" ou "4" e o IDOC com o número da respectiva operação de crédito, enquanto que, para as contrapartidas de doações, serão utilizados o IDUSO "5" e respectivo IDOC.
Sinônimo:	Identificador de Uso
SubClass Of:	ItemDeClassificacao
Nome:	Subtitulo
Descrição:	As atividades, os projetos e as operações especiais serão detalhados em subtítulos, utilizados especialmente para identificar a localização física da ação, não podendo haver, por conseguinte, alteração da finalidade da ação, do produto e das metas estabelecidas. Vale ressaltar que o critério para a priorização da localização física da ação no território é o da localização dos respectivos beneficiados.
Sinônimo:	Subtítulo
SubClass Of:	ItemDeClassificacao
Nome:	Acao
Descrição:	Operação da qual resultam produtos (bens ou serviços) que contribuem para atender ao objetivo de um programa. Incluem-se também no conceito de ação as transferências obrigatórias ou voluntárias a outros entes da Federação e a pessoas físicas e jurídicas, na forma de subsídios, subvenções, auxílios, contribuições, entre outros, e os financiamentos.
Sinônimo:	Ação
SubClass Of:	ItemDeClassificacao
Nome:	Programa
Descrição:	Toda ação do Governo está estruturada em programas orientados para a realização dos objetivos estratégicos de finidos para o período do PPA, ou seja, quatro anos.
Sinônimo:	
SubClass Of:	ItemDeClassificacao
Nome:	EspecificacaoDaFonteDestinacao
Descrição:	A classificação por fonte/destinação de recursos tem como objetivo identificar as fontes de financiamento dos gastos públicos. As fontes/destinações de recursos reúnem naturezas de receita conforme regras previamente estabelecidas. Por meio do orçamento público, essas fontes/destinações são associadas a determinadas despesas de forma a evidenciar os meios para atingir os objetivos públicos.
Sinônimo:	Especificação da Fonte/Destinação, Classificação por Fonte/Destinação
SubClass Of:	ItemDeClassificacao
Nome:	SubgrupoDaReceita
Descrição:	Detalha o Grupo da Receita.
Sinônimo:	Subgrupo da Receita
SubClass Of:	ItemDeClassificacao
Nome:	GrupoDaReceita
Descrição:	Divide as receitas em grupos de acordo com sua fonte.
Sinônimo:	Grupo da Receita
SubClass Of:	ItemDeClassificacao
Nome:	GrupoDaFonteDestinacao
Descrição:	Nível menos detalhado da classificação por fonte/destinação. Define os grupos de fonte/destinação que serão detalhados pela especificação da fonte/destinação. Representa o 1º dígito da classificação por fonte/destinação.
Sinônimo:	Grupo da Fonte/Destinação
SubClass Of:	ItemDeClassificacao

Nome:	Orgao
Descrição:	Um órgão ou uma UO não correspondem necessariamente a uma estrutura administrativa, como ocorre, por exemplo, com alguns fundos especiais e com os órgãos Transferências a Estados, Distrito Federal e Municípios, Encargos Financeiros da União, Operações Oficiais de Crédito, Refinanciamento da Dívida Pública Mobiliária Federal e Reserva de Contingência
Sinônimo:	Órgão
SubClass Of:	ItemDeClassificacao
Nome:	UnidadeDeMedida
Descrição:	Informa qual a unidade de medida que será utilizada para verificar o sucesso, ou não, de uma ação.
Sinônimo:	Unidade de Medida
SubClass Of:	ItemDeClassificacao
Nome:	FormaDeImplementacao
Descrição:	Descrição de todas as etapas do processo até a entrega do produto, inclusive as desenvolvidas por parceiros. Uma ação pode ter mais de uma forma de implementação. Elas podem ser: direta, descentralizada, transferência ou linha de crédito.
Sinônimo:	Forma de Implementação
SubClass Of:	ItemDeClassificacao
Nome:	BaseLegal
Descrição:	Informa qual lei regula aquela despesa.
Sinônimo:	Base legal
SubClass Of:	ItemDeClassificacao
Nome:	Meta
Descrição:	Informa, termos numéricos, qual é o produto final da ação.
Sinônimo:	
SubClass Of:	ItemDeClassificacao
Nome:	Projeto
Descrição:	Instrumento de programação utilizado para alcançar o objetivo de um programa, envolvendo um conjunto de operações, limitadas no tempo, das quais resulta um produto que concorre para a expansão ou o aperfeiçoamento da ação de governo. Exemplo: ação 7M64 Construção de Trecho Rodoviário – Entroncamento BR-472 – Fronteira Brasil/Argentina – na BR-468 – no Estão do Rio Grande do Sul. Corresponde ao 1º dígito 1,3,5 ou 7 do código da ação.
Sinônimo:	
SubClass Of:	Acao
Nome:	Etapa
Descrição:	Atributo específico dos projetos. Os projetos em alguns casos podem ter suas etapas intermediárias detalhadas.
Sinônimo:	
SubClass Of:	ItemDeClassificacao

Nome:	Atividade
Descrição:	Instrumento de programação utilizado para alcançar o objetivo de um programa, envolvendo um conjunto de operações que se realizam de modo contínuo e permanente, das quais resulta um produto ou serviço necessário à manutenção da ação de Governo. Exemplo: ação 4339 Qualificação da Regulação e Fiscalização da Saúde Suplementar. O Código inicia com os números 2, 4, 6 ou 8.
Sinônimo:	
SubClass Of:	Acao
Nome:	OperacaoEspecial
Descrição:	Despesas que não contribuem para a manutenção, expansão ou aperfeiçoamento das ações de governo, das quais não resulta um produto e não geram contraprestação direta sob a forma de bens ou serviços. O código inicia com 0.
Sinônimo:	Operação Especial
SubClass Of:	Acao
Nome:	Agent
Descrição:	Classe importada do Dublin Core. Ne-la é possível instanciar pessoas, organizações, agentes de software, etc.
Sinônimo:	Agente
SubClass Of:	AgentClass
Nome:	ProgramaDeGestaoManutencaoEServicosAoEstado
Descrição:	É instrumento do Plano que classifica um conjunto de ações destinadas ao apoio, à gestão e à manutenção da atuação governamental, bem como as ações não tratadas nos Programas Temáticos por meio de suas iniciativas.
Sinônimo:	Programa de Gestão, Manutenção e Serviços ao Estado; PGMSE
SubClass Of:	Programa
Nome:	ProgramaTematico
Descrição:	Retrata no PPA a agenda de governo organizada pelos Temas das Políticas Públicas e orienta a ação governamental. Sua abrangência deve ser a necessária para representar os desafios e organizar a gestão, o monitoramento, a avaliação, as transversalidades, as multissetorialidade e a territorialidade. O Programa Temático se desdobra em objetivos e iniciativas.
Sinônimo:	Programa Temático
SubClass Of:	Programa

Apêndice C. Documentação das Propriedades da Ontologia

Tabela 7. Documentação das Propriedades da Ontologia

Documentação das Propriedades da Ontologia	
Nome:	acaoTem
Descrição:	Propriedades pertencentes a uma Ação. Apenas para efeitos de organização.
Tipo:	Object
Domínio:	Acao
Imagem:	Nenhuma
Subproperty Of:	topObjectProperty
Característica:	Nenhuma
Nome:	temBaseLegal
Descrição:	Relaciona uma ação com uma ou mais bases legais para informar qual lei sustenta determinada ação
Tipo:	Object
Domínio:	Acao
Imagem:	BaseLegal
Subproperty Of:	acaoTem
Característica:	Nenhuma
Nome:	temFormaDeImplementacao
Descrição:	Relaciona uma ação com uma ou mais formas de implementação
Tipo:	Object
Domínio:	Acao
Imagem:	FormaDeImplementacao
Subproperty Of:	acaoTem
Característica:	Nenhuma
Nome:	temMeta
Descrição:	Relaciona uma ação com uma meta. Exemplo: Ação X temMeta -> Pesquisa Realizada
Tipo:	Object
Domínio:	Acao
Imagem:	Meta
Subproperty Of:	acaoTem
Característica:	Funcional
Nome:	temUnidadeDeMedida
Descrição:	Relaciona uma ação com uma unidade de medida. Exemplo: Ação X temUnidadeDeMedida -> metro ²
Tipo:	Object
Domínio:	Acao
Imagem:	temUnidadeDeMedida
Subproperty Of:	acaoTem
Característica:	Funcional

Nome:	compoeOrgao
Descrição:	Relaciona uma unidade orçamentária com um órgão.
Tipo:	Object
Domínio:	UnidadeOrçamentaria
Imagem:	Orgao
Subproperty Of:	topObjectProperty
Característica:	Funcional, Inverse of eCompostoUO
Nome:	compoeProjeto
Descrição:	Relaciona uma etapa com seu projeto.
Tipo:	Object
Domínio:	Etapa
Imagem:	Projeto
Subproperty Of:	topObjectProperty
Característica:	Funcional, Inverse of eCompostoPorEtapa
Nome:	despesaTem
Descrição:	Propriedades pertencentes a uma Despesa. Apenas para efeitos de organização.
Tipo:	Object
Domínio:	Despesa
Imagem:	Nenhuma
Subproperty Of:	topObjectProperty
Característica:	Nenhuma
Nome:	atuaNaFuncao
Descrição:	Relaciona uma despesa com sua respectiva função.
Tipo:	Object
Domínio:	Despesa
Imagem:	Funcao
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	atuaNaSubfuncao
Descrição:	Relaciona uma despesa com sua respectiva subfunção.
Tipo:	Object
Domínio:	Despesa
Imagem:	Funcao
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	eRealizadoPorUO
Descrição:	Relaciona uma despesa com sua respectiva Unidade Orçamentária.
Tipo:	Object
Domínio:	Despesa
Imagem:	UnidadeOrçamentaria
Subproperty Of:	despesaTem
Característica:	Funcional

Nome:	pertenceAEsfera
Descrição:	Relaciona uma despesa com a esfera a qual ela pertence.
Tipo:	Object
Domínio:	Despesa
Imagem:	Esfera
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	pertenceAPrograma
Descrição:	Relaciona uma despesa ao seu programa
Tipo:	Object
Domínio:	Despesa
Imagem:	Programa
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temAcao
Descrição:	Relaciona uma despesa com sua respectiva ação
Tipo:	Object
Domínio:	Despesa
Imagem:	Base Legal
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temCategoriaEconomicDaDespesa
Descrição:	Relaciona uma despesa com sua categoria econômica
Tipo:	Object
Domínio:	Despesa
Imagem:	CategoriaEconomicDaDespesa
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temElementoDeDespesa
Descrição:	Relaciona uma despesa com seu Elemento de despesa.
Tipo:	Object
Domínio:	Despesa
Imagem:	ElementoDeDespesa
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temFonte
Descrição:	Relaciona uma despesa com sua fonte.
Tipo:	Object
Domínio:	Despesa
Imagem:	EspecificacaoDaFonteDestinacao
Subproperty Of:	despesaTem
Característica:	Funcional

Nome:	temGND
Descrição:	Relaciona uma despesa com seu grupo de natureza de despesa.
Tipo:	Object
Domínio:	Despesa
Imagem:	GND
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temIDOC
Descrição:	Relaciona uma despesa com seu respectivo IDOC.
Tipo:	Object
Domínio:	Despesa
Imagem:	Idoc
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temIDResultadoPrimarioDaDespesa
Descrição:	Relaciona uma despesa com seu identificador de resultado primário.
Tipo:	Object
Domínio:	Despesa
Imagem:	IdentificadorResultadoPrimarioDespesa
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temIDUSO
Descrição:	Relaciona uma despesa com seu identificador de uso.
Tipo:	Object
Domínio:	Despesa
Imagem:	Iduso
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temModalidadeDeAplicacao
Descrição:	Relaciona uma despesa com sua modalidade de aplicação.
Tipo:	Object
Domínio:	Despesa
Imagem:	ModalidadeDeAplicacao
Subproperty Of:	despesaTem
Característica:	Funcional
Nome:	temSubtitulo
Descrição:	Relaciona uma despesa com seu subtítulo.
Tipo:	Object
Domínio:	Despesa
Imagem:	Subtitulo
Subproperty Of:	despesaTem
Característica:	Funcional

Nome:	detalhaGrupoDaFonteDestinacao
Descrição:	Relaciona uma especificação da fonte/destinação com seu respectivo grupo da fonte.
Tipo:	Object
Domínio:	EspecificacaoDaFonteDestinacao
Imagem:	GrupoDaFonteDestinacao
Subproperty Of:	topObjectProperty
Característica:	Funcional, inverseOf e DetalhadoPorEspecificacaoDaFonteDestinacao
Nome:	detalhaGrupoDaReceita
Descrição:	Relaciona um subgrupo da receita com seu respectivo grupo.
Tipo:	Object
Domínio:	SubgrupoDaReceita
Imagem:	GrupoDaReceita
Subproperty Of:	topObjectProperty
Característica:	Funcional, inverseOf e DetalhadoPorSubgrupoDaReceita
Nome:	detalhaSubgrupoDaReceita
Descrição:	Relaciona uma especificacao do grupo da receita com seu respectivo subgrupo.
Tipo:	Object
Domínio:	EspecificacaoGrupoDaReceita
Imagem:	SubgrupoDaReceita
Subproperty Of:	topObjectProperty
Característica:	Funcional, inverseOf e DetalhadoPorEspecificacaoDoGrupoDaReceita
Nome:	eCompostoPorEtapa
Descrição:	Relaciona um projeto com suas etapas.
Tipo:	Object
Domínio:	Projeto
Imagem:	Etapa
Subproperty Of:	topObjectProperty
Característica:	inverseOf compoeProjeto
Nome:	eCompostoPorUO
Descrição:	Relaciona um órgão com suas unidades orçamentárias
Tipo:	Object
Domínio:	Orgao
Imagem:	UnidadeOrçamentaria
Subproperty Of:	topObjectProperty
Característica:	inverseOf compoeOrgao
Nome:	eDetalhadoPorEspecificacaoDaFonteDestinacao
Descrição:	Relaciona um grupo da fonte/destinacao com suas respectivas especificações da fonte/destinação.
Tipo:	Object
Domínio:	GrupoDaFonteDestinacao
Imagem:	EspecificacaoDaFonteDestinacao
Subproperty Of:	topObjectProperty
Característica:	inverseOf detalhaGrupoDaFonteDestinacao

Nome:	eDetalhadoPorEspecificacaoDoGrupoDaReceita
Descrição:	Relaciona um subgrupo da receita com suas respectivas especificações de grupo da receita.
Tipo:	Object
Domínio:	SubgrupoDaReceita
Imagem:	EspecificacaoGrupoDaReceita
Subproperty Of:	topObjectProperty
Característica:	inverseOf detalhaSubgrupoDaReceita
Nome:	eDetalhadoPorSubgrupoDaReceita
Descrição:	Relaciona um grupo da receita com seus respectivos subgrupos.
Tipo:	Object
Domínio:	GrupoDaReceita
Imagem:	SubgrupoDaReceita
Subproperty Of:	topObjectProperty
Característica:	inverseOf detalhaGrupoDaReceita
Nome:	impactaAcao
Descrição:	Relaciona um projeto com uma ação impactada por ele.
Tipo:	Object
Domínio:	Projeto
Imagem:	Acao
Subproperty Of:	topObjectProperty
Característica:	Nenhuma
Nome:	orcamentoTem
Descrição:	Propriedades pertencentes a um orçamento. Apenas para efeitos de organização.
Tipo:	Object
Domínio:	Orcamento
Imagem:	Nenhuma
Subproperty Of:	topObjectProperty
Característica:	InverseOf pertenceAOrcamento
Nome:	temDespesa
Descrição:	Relaciona um orçamento com as despesas pertencentes a ele.
Tipo:	Object
Domínio:	Orcamento
Imagem:	Despesa
Subproperty Of:	orcamentoTem
Característica:	Nenhuma
Nome:	temReceita
Descrição:	Relaciona um orçamento com as receitas pertencentes a ele.
Tipo:	Object
Domínio:	Orcamento
Imagem:	Receita
Subproperty Of:	orcamentoTem
Característica:	Nenhuma

Nome:	pertenceAOrcamento
Descrição:	Relaciona cada despesa com seu respectivo orçamento.
Tipo:	Object
Domínio:	Receita or Despesa
Imagem:	Orcamento
Subproperty Of:	topObjectProperty
Característica:	Funcional, InverseOf orcamentoTem
Nome:	receitaTem
Descrição:	Propriedades pertencentes a uma receita. Apenas para efeitos de organização.
Tipo:	Object
Domínio:	Receita
Imagem:	Nenhuma
Subproperty Of:	topObjectProperty
Característica:	Nenhuma
Nome:	temAlinea
Descrição:	Relaciona uma receita com sua alínea.
Tipo:	Object
Domínio:	Receita
Imagem:	Alinea
Subproperty Of:	receitaTem
Característica:	Funcional
Nome:	temCategoriaEconomicDaReceita
Descrição:	Relaciona uma receita com sua categoria econômica.
Tipo:	Object
Domínio:	Receita
Imagem:	CategoriaEconomicDaReceita
Subproperty Of:	receitaTem
Característica:	Funcional
Nome:	temClassificacaoDoGrupoDaReceita
Descrição:	Relaciona uma receita com sua especificação do grupo da receita.
Tipo:	Object
Domínio:	Receita
Imagem:	EspecificacaoGrupoDaReceita
Subproperty Of:	receitaTem
Característica:	Funcional
Nome:	temDestino
Descrição:	Relaciona uma receita com sua especificação da fonte/destinação.
Tipo:	Object
Domínio:	Receita
Imagem:	EspecificacaoDaFonteDestinacao
Subproperty Of:	receitaTem
Característica:	Funcional

Nome:	temEspecie
Descrição:	Relaciona uma receita com sua espécie.
Tipo:	Object
Domínio:	Receita
Imagem:	Especie
Subproperty Of:	receitaTem
Característica:	Funcional
Nome:	temIDResultadoPrimarioDaReceita
Descrição:	Relaciona uma receita com seu identificador de resultado primário da receita.
Tipo:	Object
Domínio:	Receita
Imagem:	IdentificadorResultadoPrimarioReceita
Subproperty Of:	receitaTem
Característica:	Funcional
Nome:	temOrigem
Descrição:	Relaciona uma receita com sua origem.
Tipo:	Object
Domínio:	Receita
Imagem:	Origem
Subproperty Of:	receitaTem
Característica:	Funcional
Nome:	temRubrica
Descrição:	Relaciona uma receita com sua rubrica.
Tipo:	Object
Domínio:	Receita
Imagem:	Rubrica
Subproperty Of:	receitaTem
Característica:	Funcional
Nome:	temSubalinea
Descrição:	Relaciona uma receita com sua subalinea.
Tipo:	Object
Domínio:	Receita
Imagem:	Subalinea
Subproperty Of:	receitaTem
Característica:	Nenhuma
Nome:	rdfs:label
Descrição:	Propriedade de anotação do rdfs que descreve o recurso em linguagem comum.
Tipo:	AnnotationProperty
Domínio:	Nenhuma
Imagem:	Nenhuma
Subproperty Of:	Nenhuma
Característica:	Nenhuma

Nome:	dc:title
Descrição:	Propriedade do Dublin Core que descreve o recurso em linguagem comum.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	Literal
Subproperty Of:	topDataProperty
Característica:	Nenhuma
Nome:	codigo
Descrição:	Nesta propriedade deve ser informado o código padronizado do recurso.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	string
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	cronograma
Descrição:	Descrição do cronograma de uma etapa de um determinado projeto
Tipo:	Datatype
Domínio:	Etapa
Imagem:	String
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	dc:date
Descrição:	Propriedade do Dublin Core que descreve datas.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	Literal
Subproperty Of:	topDataProperty
Característica:	Nenhuma
Nome:	ano
Descrição:	Descreve a qual ano pertence um recurso, mais precisamente um orçamento.
Tipo:	Datatype
Domínio:	Orcamento
Imagem:	Literal
Subproperty Of:	dc:date
Característica:	Funcional
Nome:	dataDeAtualizacao
Descrição:	Informa a data de atualização da informação representada no recurso.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	Literal
Subproperty Of:	dc:date
Característica:	Funcional

Nome:	dataDeInicio
Descrição:	Informa a data de início de uma etapa ou projeto.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	Literal
Subproperty Of:	dc:date
Característica:	Funcional
Nome:	dataDeTermino
Descrição:	Informa a data de início de uma etapa ou projeto.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	Literal
Subproperty Of:	dc:date
Característica:	Funcional
Nome:	unidadeResponsavel
Descrição:	Unidade administrativa, entidade, inclusive empresa estatal ou parceiro (Estado, Distrito Federal, Município, ou setor privado), responsável pela execução da ação.
Tipo:	Datatype
Domínio:	Acao
Imagem:	string
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	finalidade
Descrição:	Objetivo a ser alcançado pela ação.
Tipo:	Datatype
Domínio:	Acao
Imagem:	string
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	totalFisico
Descrição:	Quanto se pretende desenvolver? Quanto foi desenvolvido? Este atributo é da despesa pois cada localização pode ter uma meta física diferente. Também pode ser usado em um projeto.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	integer
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	custoTotalEstimado
Descrição:	Custo estimado do Projeto.
Tipo:	Datatype
Domínio:	Projeto
Imagem:	double
Subproperty Of:	topDataProperty
Característica:	Funcional

Nome:	descricao
Descrição:	Expressa, de forma sucinta, o que é efetivamente feito no âmbito da ação, seu escopo e suas delimitações.
Tipo:	Datatype
Domínio:	Acao
Imagem:	string
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	detalhamento
Descrição:	De forma sucinta, o que é efetivamente feito no âmbito da etapa ou da implementação da ação
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	string
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	especificacaoDoProduto
Descrição:	Características do produto acabado, visando sua melhor identificação. Para o exemplo citado, a especificação é Km de Trecho Pavimentado.
Tipo:	Datatype
Domínio:	Acao
Imagem:	string
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	justificativaDoImpacto
Descrição:	Breve explicação do motivo do impacto que o projeto gera em uma ação.
Tipo:	Datatype
Domínio:	Projeto
Imagem:	string
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	mes
Descrição:	Informa o mês da execução da receita ou da despesa. Pode ser usado em qualquer outro recurso se necessário.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	integer
Subproperty Of:	dc:date
Característica:	Funcional
Nome:	numero
Descrição:	Informa o número de uma etapa.
Tipo:	Datatype
Domínio:	Etapa
Imagem:	integer
Subproperty Of:	topDataProperty
Característica:	Funcional

Nome:	status
Descrição:	Informa em qual etapa do processo orçamentário encontra-se a receita ou despesa.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	string
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	valor
Descrição:	Informa o valor de uma despesa ou receita.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	double
Subproperty Of:	topDataProperty
Característica:	Funcional
Nome:	valorLiquidoArrecadado
Descrição:	Informa o valor líquido arrecadado de uma receita.
Tipo:	Datatype
Domínio:	Receita
Imagem:	double
Subproperty Of:	valor
Característica:	Funcional
Nome:	valorBrutoArrecadado
Descrição:	Informa o valor bruto arrecadado de uma receita.
Tipo:	Datatype
Domínio:	Receita
Imagem:	double
Subproperty Of:	valor
Característica:	Funcional
Nome:	valorPago
Descrição:	Informa o valor pago de uma despesa.
Tipo:	Datatype
Domínio:	Despesa
Imagem:	double
Subproperty Of:	valor
Característica:	Funcional
Nome:	valorEmpenhado
Descrição:	Informa o valor do empenho de uma despesa.
Tipo:	Datatype
Domínio:	Despesa
Imagem:	double
Subproperty Of:	valor
Característica:	Funcional

Nome:	valorPrevisto
Descrição:	Informa o valor previsto de uma Receita ou Despesa.
Tipo:	Datatype
Domínio:	Nenhuma
Imagem:	double
Subproperty Of:	valor
Característica:	Funcional