

# PUC



ISSN 0103-9741

Monografias em Ciência da Computação  
nº 27/94

## **Recuperação de Requisitos a partir de Especificações Estruturadas - um Estudo de Caso**

Paulo Monteiro Cerqueira  
Julio Cesar Sampaio do Prado Leite

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO  
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900  
RIO DE JANEIRO - BRASIL

PUC RIO - DEPARTAMENTO DE INFORMÁTICA

ISSN 0103-9741

Monografias em Ciência da Computação, Nº 27/94

Editor: Carlos J. P. Lucena

Setembro, 1994

**Recuperação de Requisitos a partir de Especificações  
Estruturadas - um Estudo de Caso \***

Paulo Monteiro Cerqueira

Julio Cesar Sampaio do Prado Leite

\* Trabalho parcialmente patrocinado pelo Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil.

**In charge of publications:**

Rosane Teles Lins Castilho

Assessoria de Biblioteca, Documentação e Informação

PUC Rio — Departamento de Informática

Rua Marquês de São Vicente, 225 — Gávea

22453-900 — Rio de Janeiro, RJ

Brasil

Tel. +55-21-529 9386

Telex +55-21-31048

Fax +55-21-511 5645

E-mail: [rosane@inf.puc-rio.br](mailto:rosane@inf.puc-rio.br)

# **Recuperação de requisitos a partir de especificações estruturadas - Um estudo de caso.**

**Paulo Monteiro Cerqueira**

*Departamento de Informática - Universidade Católica de Petrópolis*

**Julio Cesar Sampaio do Prado Leite**

*Departamento de Informática - Pontifícia Universidade Católica do Rio de Janeiro*

## **Resumo**

Este trabalho apresenta um processo para recuperação de requisitos a partir de especificações estruturadas. Este processo é usado como um mecanismo de engenharia reversa que, aplicado a um número expressivo de especificações, permitiu que uma série de dados fossem compilados. Dentre esses dados, destacamos uma coleta de erros sintáticos, semânticos e vocabulário empregado para descrever a funcionalidade nas especificações. Com base nestes dados, listamos várias observações que podem contribuir para o estudo do problema da construção de domínios.

## **1. Introdução**

Recuperar conhecimento previamente codificado, objetivando um posterior reuso é uma área onde grande parte da comunidade de engenharia de software vem concentrando seus esforços. Uma das modalidades de recuperação, concentra-se no nível dos requisitos, isto é, da representação, em seu mais elevado nível de abstração, das necessidades e desejos dos usuários dos sistemas de software. A recuperação de requisitos torna-se particularmente interessante quando consideramos a grande quantidade de sistemas cujas especificações estão modeladas segundo notações/técnicas conhecidas como Análise Estruturada [3], [2] e variações (p.e. Análise Essencial [7]).

Na Sessão 2 deste trabalho, apresentaremos um processo para recuperação de requisitos juntamente com um modelo de representação de requisitos, composto por sentenças em linguagem natural e um elemento auxiliar de modelagem chamado Léxico Ampliado da Linguagem (LAL). As características da amostra empregada no estudo de caso e exemplos de recuperação são tema da sessão 3. Na Sessão 4, apresentaremos os resultados da aplicação do processo de recuperação, na amostra, através de alguns produtos tais como: uma taxonomia de erros de modelagem e uma coleta de vocabulário. A Sessão 5, conclui este trabalho.

## **2 - Um processo para recuperação de requisitos**

Um processo para recuperação de requisitos necessita de uma infraestrutura de modelagem para armazenar os requisitos recuperados. A aplicação do nosso processo de recuperação pode ser visualizada como uma transformação entre modelos, isto é, a partir de um modelo baseado em notações da análise estruturada, transformamos requisitos, embutidos na representação alvo, para uma outra notação, baseada em linguagem natural restrita. Acreditamos que, o modelo baseado em linguagem natural possa ser empregado como uma ferramenta auxiliar no processo de validação dos requisitos junto à comunidade de usuários do sistema de software.

### **2.1 Uma infraestrutura para modelagem de requisitos**

Durante algum tempo, a área de engenharia de software vêm concentrando seus esforços na representação de requisitos. Várias propostas de formalização das mais formais a menos formais vêm sendo apresentadas. Nestas propostas, destacamos os métodos da análise estruturada e seus correlatos, pela sua presença no ambiente industrial. No entanto, apesar da aparente simplicidade, a tarefa de modelar requisitos

de modo que, possam ser validados junto à comunidade de usuários de um sistema de software está longe de ser uma atividade trivial. A comunidade de usuários de um sistema de software é normalmente heterogênea e, não existem garantias de que todos os seus membros compreenderão as notações empregadas pelos métodos estruturados [1].

As considerações citadas acima, nos levam a concluir que, modelos de requisitos devem ser construídos levando-se em consideração todos os envolvidos no processo de desenvolvimento de um produto de software. Idealmente, diversas representações deveriam ser empregadas, representando diferentes pontos de vista sobre o sistema, em diversos níveis de abstração. Acreditamos que, modelos baseados em linguagem natural com restrições sintáticas possam preencher uma lacuna de comunicação.

Em [4] é apresentado um modelo para representação de requisitos baseado em um conjunto de sentenças de requisitos e um Léxico Ampliado da Linguagem (LAL). Segundo [4], é possível, através da combinação de um conjunto de sentenças em linguagem natural restrita e um Léxico Ampliado da Linguagem (LAL), obter-se informação equivalente à de especificações estruturadas, porém em uma forma orientada para o usuário. Uma razão para se empregar sentenças simples, é que não podemos menosprezar o risco de se complicar o processo de entender o que é realmente desejado em um sistema de software. O léxico é um dicionário estendido onde cada termo é definido quanto à sua denotação e conotação, utilizando-se os princípios do vocabulário mínimo e da circularidade. As sentenças de requisitos são classificadas em entrada, transformação e saída.

## 2.2 Aperfeiçoamento de um processo para recuperação de requisitos

Um processo para recuperação de requisitos a partir de especificações estruturadas foi apresentado em [4]. Este processo produz um Documento de Requisitos do Usuário (URD - User Requirements Document) composto por um conjunto de sentenças de requisitos e um Léxico Ampliado da Linguagem (LAL).

Um estudo de caso conduzido por uma equipe de alunos do programa de mestrado da PUC-Rio produziu resultados iniciais. Cabe ressaltar que, nesta ocasião, o processo ainda não estava totalmente delineado. Estes resultados iniciais, nos levaram a concluir que, o processo poderia ser empregado em um número mais significativo de especificações, tendo por objetivo a comprovação de sua eficácia.

A aplicação do processo, através de um estudo de caso, permitiu que um novo enfoque de recuperação[10], influenciado pelos conceitos da análise essencial [7] fosse empregado. O nosso objetivo ao adotar esse enfoque foi o de otimizar o processo de recuperação de requisitos no estudo de caso.

O processo para recuperação de requisitos emprega, com pequenas alterações, a infraestrutura de modelagem apresentada no item 2.1. A principal alteração refere-se à utilização de apenas duas classes de requisitos (fundamentais, armazenamento) em substituição às classes entrada, transformação e saída. Explicaremos a seguir as duas classes de requisitos :

**Requisitos Fundamentais :** São os requisitos associados às saídas (respostas) do sistema. Determinam a razão da existência do sistema, sendo identificados com as principais expectativas dos usuários.

**Requisitos de Armazenamento :** São identificados com operações de escrita em depósitos de dados. O fato de não considerarmos os acessos de leitura à depósitos, ou seja, o outro extremo da conexão, não invalida a nossa estratégia, pois o consumo da informação armazenada certamente fará parte da produção de uma saída (resposta) do sistema.

A sentença de requisitos associada a um requisito de armazenamento gerará uma entrada no léxico correspondente ao depósito acessado, e o conjunto dos requisitos de armazenamento e suas respectivas referências no léxico proporcionarão uma cobertura adequada da dimensão informação.

Em nossa proposta, a identificação dos requisitos nas especificações é bastante direta. A partir do diagrama de contexto, para cada entidade externa, identificam-se as saídas do sistema para a entidade. Cada

saída produzirá um requisito fundamental. A identificação dos requisitos de armazenamento é igualmente simples. Percorre-se todas as folhas do DFD, identificando os depósitos de dados modelados. Para cada depósito identificam-se os acessos de escrita. Cada acesso de escrita produzirá um requisito de armazenamento. Os procedimentos acima permitem a elaboração de uma lista de recuperação de requisitos, onde temos um mapeamento das sentenças de requisitos que serão recuperadas. Naturalmente, a formação da sentença exigirá do responsável pela recuperação uma investigação mais detalhada da especificação, mas ao aplicarmos o procedimento, saberemos, antecipadamente quantos requisitos deverão ser recuperados. O número de requisitos será igual ao número de fluxos de dados de saída (respostas) identificados no diagrama de contexto, mais o número de acessos de escrita (armazenamento/atualização) identificados em todos os níveis do DFD.

Como afirmamos anteriormente, a formação da sentença de requisitos ainda demandará uma análise mais detalhada da especificação. No caso de requisitos fundamentais, essa análise mais detalhada consiste em uma busca em profundidade na especificação (DFD), buscando-se o(s) processo(s) primitivo(s) responsável(is) pela saída. Ao se localizar tais processos, examinaremos sua descrição detalhada, buscando possíveis condições que determinem a produção da saída. Nesta etapa, também consultaremos o dicionário de dados.

- **Heurísticas para recuperação de requisitos**

#### **Visão geral do processo**

1. Partindo do diagrama de contexto, faça :

1.1 Para cada **Entidade Externa**, faça :

/\* produção de requisito fundamental \*/

1.1.1 Para cada **saída do sistema para a entidade**, faça:

1.1.1.1 Procurar as **atividades** que produzem a saída nos vários níveis (busca em profundidade).

No nível mais detalhado examinar a mini-especificação.

1.1.1.2 Localizar a saída no dicionário de dados.

1.1.1.3 Escrever a sentença com o requisito, seguindo as regras de construção.

1.2 Para cada **depósito de dados** encontrado faça :

/\* produção de requisito de armazenamento \*/

1.2.1 Para cada **acesso de escrita** ao depósito faça :

1.2.1.1 Procurar as atividades que produzem o acesso nos vários níveis do DFD

(busca em profundidade).

No nível mais detalhado examinar a mini-especificação.

1.2.1.2 Localizar o acesso (se possuir descrição) e/ou o depósito no dicionário de dados.

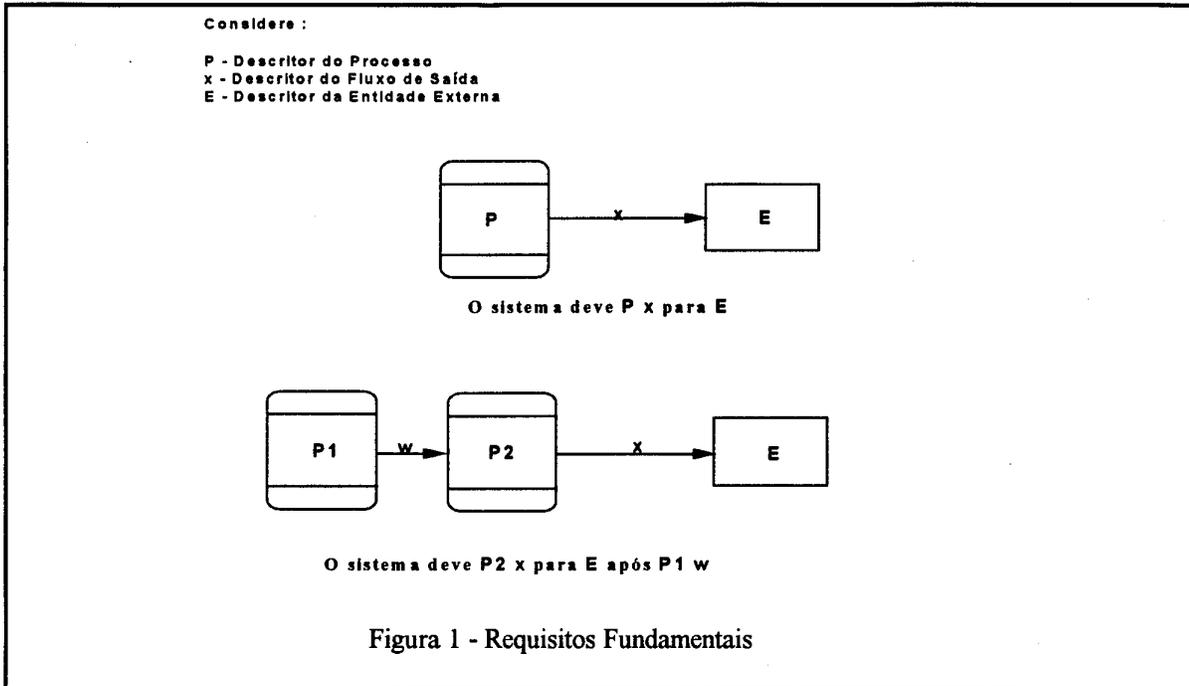
1.2.1.3 Escrever a sentença com o requisito, seguindo as regras de construção.

Obs.: Ao localizarmos os elementos da sentença de requisitos no dicionário de dados, estamos buscando subsídios para o processo de construção de entradas no LAL.

- **Construção de sentenças de requisitos**

**Requisitos Fundamentais**

No caso de requisitos fundamentais, a construção da sentença de requisitos normalmente obedece às seguintes regras de formação (Figura 1):



O nome da ação deve, sempre que possível, ser obtido a partir do descritor do processo<sup>1</sup> que produz a saída. Esta abordagem permite verificar se os descritores dos processos são significativos ou não. Um descritor não significativo produzirá uma sentença de requisitos com pouco valor semântico. Exemplos de descritores pouco significativos são TRATAR, PROCESSAR, MANIPULAR. Nestes casos, sugere-se que na construção da sentença de requisitos o verbo pouco significativo seja substituído por *emitir*. Neste caso, a sentença assumirá a seguinte forma :

**O sistema deve emitir X para E (condição-1, condição-2, condição-n)**

Exemplos de sentenças de requisitos no formato descrito acima seriam :

O sistema deve emitir relatório vendas semanais para gerente quando totalizar arrecadação da semana

O sistema deve emitir conta para cliente

Considerando-se os princípios acima, uma sentença para descrever um requisito fundamental poderia assumir as seguintes formas :

"O sistema deve " + verbo-ação + objeto + "para" Agente Externo + condições-de-emissão

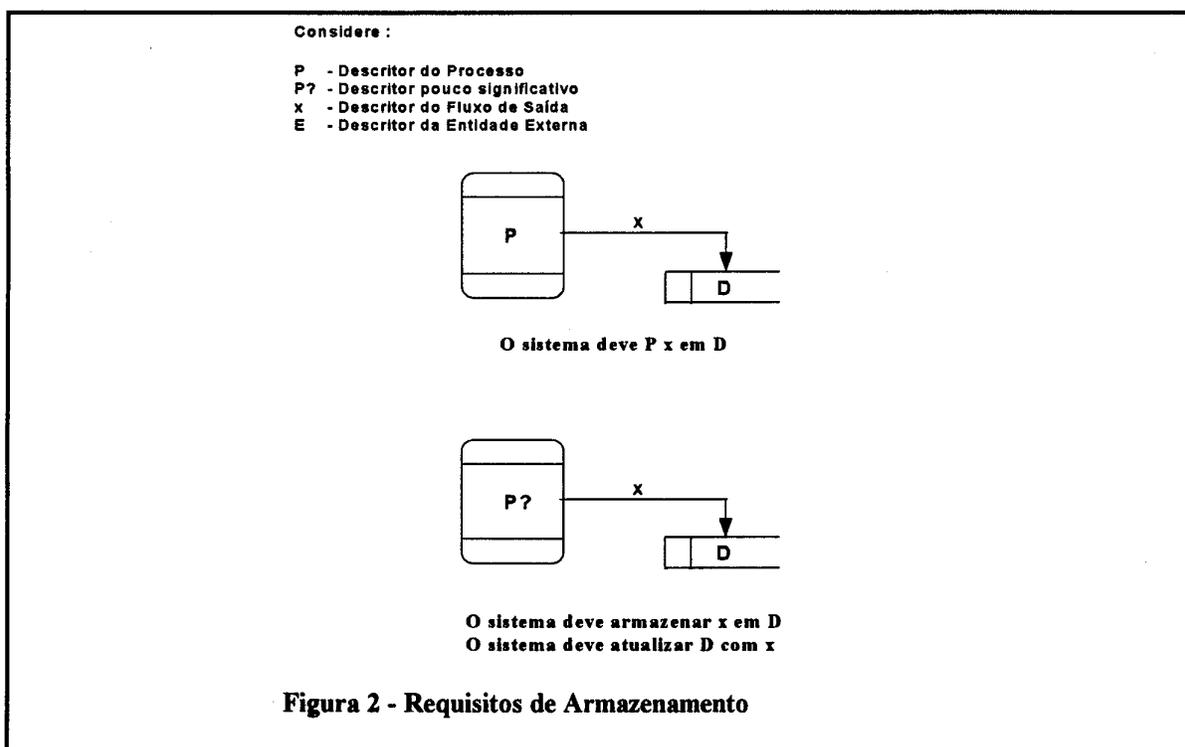
<sup>1</sup>Ao se analisar as especificações, obtivemos uma lista dos verbos indicativos de ação mais empregados.

Onde,

- Verbo-Ação* : Verbo que descreve a ação associada ao requisito fundamental. Sempre que possível, deve ser obtido a partir do nome do processo que produz a saída.
- Objeto* : Fluxo de dados de saída. Equivalente à resposta em análise essencial.
- Agente Externo* : Entidade Externa
- Condições-de-emissão* : Condição(ões) associada(s) à produção de *objeto*. Pode ser a ocorrência de um fluxo de entrada no processo que produz a saída, ou um conjunto de condições. Para se determinar as condições de emissão, deve-se examinar a mini-especificação do processo produtor da saída.

### Requisitos de Armazenamento

No caso de requisitos de armazenamento, adotaremos a seguinte regra de formação para a sentença de requisitos (Figura 2):



Neste caso, observamos que, na grande maioria das especificações empregadas no estudo de caso, os modeladores nem sempre seguem o princípio de não empregar descritores nos acessos à depósitos. Embora esta prática não seja recomendada pelos autores dos métodos / notações estruturadas, podemos aproveitar a descrição (x) para melhorar a legibilidade da sentença de requisitos associada a um requisito de armazenamento. Também observamos que, o descritor do processo responsável pelo acesso de escrita poderia ser empregado para descrever possíveis condição associadas ao requisito de armazenamento.

Exemplos de sentenças de requisitos nos formatos acima seriam :

- O sistema deve armazenar pedido cliente em pedidos
- O sistema deve cadastrar fornecedor em fornecedores
- O sistema deve atualizar pedido com alteração pedido

• **Construção do Léxico Ampliado da Linguagem**

No processo de recuperação aperfeiçoado (vide SADT que mostra o processo - Figura 3), a construção do léxico ampliado da linguagem obedece às regras gerais descritas no processo original. Os elementos fundamentais da sentença de requisitos são destacados (sublinhados) e, geram uma entrada no LAL. Cada entrada por sua vez, pode referenciar novas entradas no LAL em suas noções e impactos. No caso de requisitos fundamentais, a informação (saída) e o consumidor (entidade externa) deverão ser destacados como entradas no LAL. Exemplos de elementos fundamentais nas sentenças de requisitos do processo aperfeiçoado são os depósitos de dados referenciados e a informação armazenada.

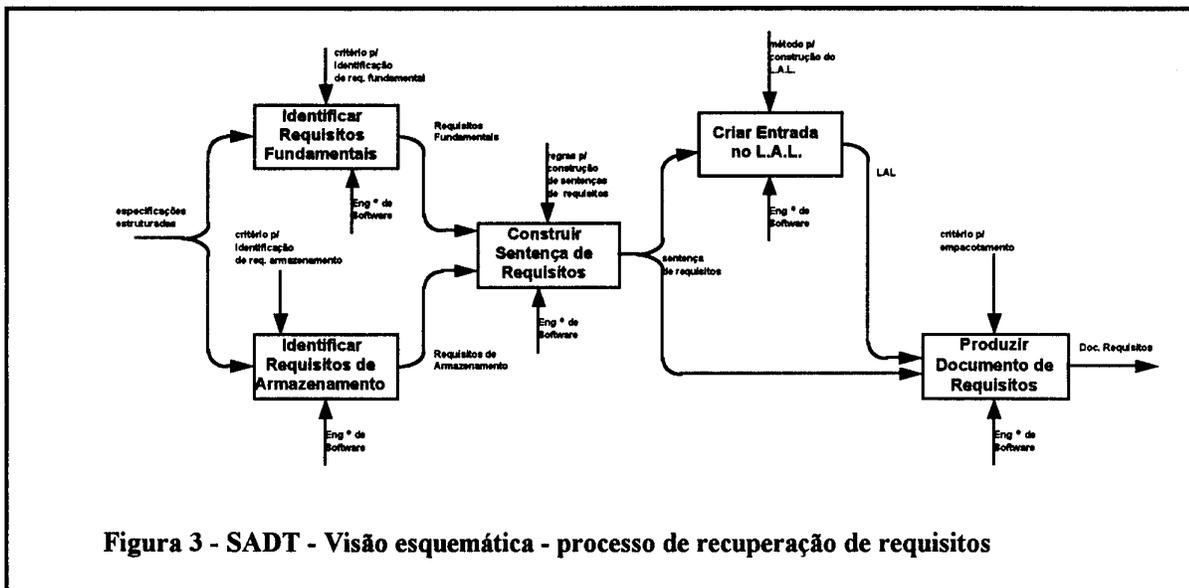


Figura 3 - SADT - Visão esquemática - processo de recuperação de requisitos

Segundo nossas observações, o número médio de requisitos recuperados, considerando todos os domínios foi de 18 requisitos. O número médio de entradas no LAL foi de 28 entradas (tabela 1). Em [5], o autor relata que uma experiência com 10 especificações estruturadas dos domínios VL-Videolocadoras e PC-Postos de Combustíveis, os números médios foram 15 e 22, para requisitos e entradas no léxico, respectivamente.

Como podemos observar, a aplicação do processo aperfeiçoado, aumentou a produtividade média da recuperação de requisitos em 20% . Quanto a produção de entradas no LAL, observamos um aumento médio de 27% no número de entradas recuperadas, considerando todos os domínios envolvidos. Se considerarmos apenas as especificações dos domínios VL-Videolocadoras e PC-Postos de Combustíveis, os valores médios são 19 e 28, para requisitos e entradas no léxico, respectivamente. Neste caso, observamos um aumento de 27%, tanto para requisitos quanto para entradas no LAL.

Embora o tamanho da amostra e as condições do experimento (apenas um ator participou do processo de recuperação aperfeiçoado) não nos permitam generalizar, os resultados indicam que, o processo aperfeiçoado melhorou as possibilidades de recuperação de requisitos.

### 3 - Estudo de Caso

O estudo de caso foi realizado através da análise de 30 especificações funcionais, de alunos de um curso de extensão em Engenharia de Software, da PUC-Rio. Estas especificações apresentavam distintos graus de qualidade, oferecendo uma oportunidade para incorporar mecanismos de comprovação informal de qualidade ao processo de recuperação.

#### 3.1 Domínios das aplicações

As especificações foram classificadas em 10 domínios distintos (RESTAURANTES{RE}, VIDEOLOCADORAS{VL}, POSTOS DE COMBUSTÍVEIS{PC}, FABRICAÇÃO DE FILMES{FI}, ADMINISTRAÇÃO PÚBLICA{AP}, DISTRIBUIÇÃO DE ALIMENTOS{DI}, BIBLIOTECAS{BI}, CONDOMÍNIOS{CD}, CONFECÇÕES{CN} e MANUTENÇÃO DE EQUIPAMENTOS{MA})

#### 3.2 Características das especificações

Algumas características de uma parte das especificações bem como resultados do processo de recuperação são apresentados na tabela 1:

**a) Especificação**

Número de identificação da especificação.

**b) Domínio**

Sigla indicadora do domínio da especificação.

**c) Enunciado**

Refere-se ao número de páginas empregadas para a apresentação de uma descrição informal do sistema modelado.

**d) Ents. Externas**

Refere-se ao número de entidades externas do sistema.

**e) Procs Nível 0**

É o número de atividades (bolhas) encontradas no primeiro nível de particionamento do sistema. Optou-se por empregar o termo *nível 0* [2], para identificar a folha referente ao primeiro nível de decomposição após o diagrama de contexto.

**f) Folhas**

É o número de diagramas (folhas) disponíveis no modelo, sem considerar o diagrama de contexto e o diagrama de nível 0. O número total de folhas do modelo será FOLHAS + 2.

**g) Depósitos**

É o número total de depósitos de dados encontrados em todas as folhas do DFD.

**h) Entradas Dic.**

É o total de entradas no dicionário de dados do modelo. Considera todas as categorias (Fluxos, Depósitos, Entidades Externas, Estruturas de Dados, etc).

**i) Processos Prim.**

Refere-se à quantidade de processos primitivos encontrados no DFD.

**j) Mini-Espec**

Refere-se à quantidade de mini-especificações de processos primitivos, encontradas no modelo

**k) Requisitos**

É o número de sentenças de requisitos recuperadas através da aplicação do processo de recuperação.

**l) LAL**

É o número de entradas no Léxico Ampliado da Linguagem, obtidas através do processo de recuperação.

**Tabela 1 - Características gerais da amostra [especificações 01 - 10]**

Especificação	01	02	03	04	05	06	07	08	09	10	Média
Domínio	VL	VL	VL	RE	RE	VL	PC	RE	RE	RE	
Enunciado	01	01	01	02	01	02	01	01	01	02	02
Ents. Externas	04	05	03	03	04	03	04	04	05	05	04
Procs Nível 0	06	12	08	06	05	06	05	05	05	05	06
Folhas	06	11	02	06	03	06	05	02	05	05	05
Depósitos	04	07	05	10	02	06	03	04	06	07	05
Entradas Dic.	46	94	22	47	35	64	56	26	63	58	61
Processos Prim.	15	32	18	20	15	19	16	10	18	20	18
Mini-Espec.	15	34	12	04	02	19	02	02	02	20	09
Requisitos	24	20	15	23	09	16	15	18	17	22	18
LAL	31	40	19	33	19	26	28	19	40	43	28

### 3.4 Exemplo de recuperação

Para o caso da especificação 6, do domínio videolocadoras, fornecemos um exemplo parcial dos resultados. Em [10], os dados são apresentados na íntegra.

Especificação : 06 - Especificação de uma locadora de vídeo

Domínio : VL - Videolocadora

#### • Requisitos Recuperados (lista parcial)

- 01) O sistema deve emitir o cartão de sócio ao sócio quando do seu cadastro
- 02) O sistema deve emitir prêmio melhor sócio mensalmente
- 03) O sistema deve emitir pedido reserva ok para sócio quando validar pedido

#### • Léxico Ampliado da Linguagem (parcial)

##### 01) ALUGUEL / LOCAÇÃO

**Noção :**

Refere-se à retirada temporária de fitas da locadora por sócios mediante pagamento

**Impacto :**

É uma ação de sócio

É a principal atividade da locadora

##### 04) CARTÃO DE SÓCIO

**Noção :**

cartão emitido pela locadora dando ao sócio o direito de utilização dos serviços da locadora

**Impacto :**

O sócio recebe o cartão de sócio quando de seu cadastro

##### 28) PEDIDO RESERVA OK

**Noção :**

Confirmação da disponibilidade de uma fita reservada, quando de um pedido de reserva

**Impacto :**  
É enviado para o sócio

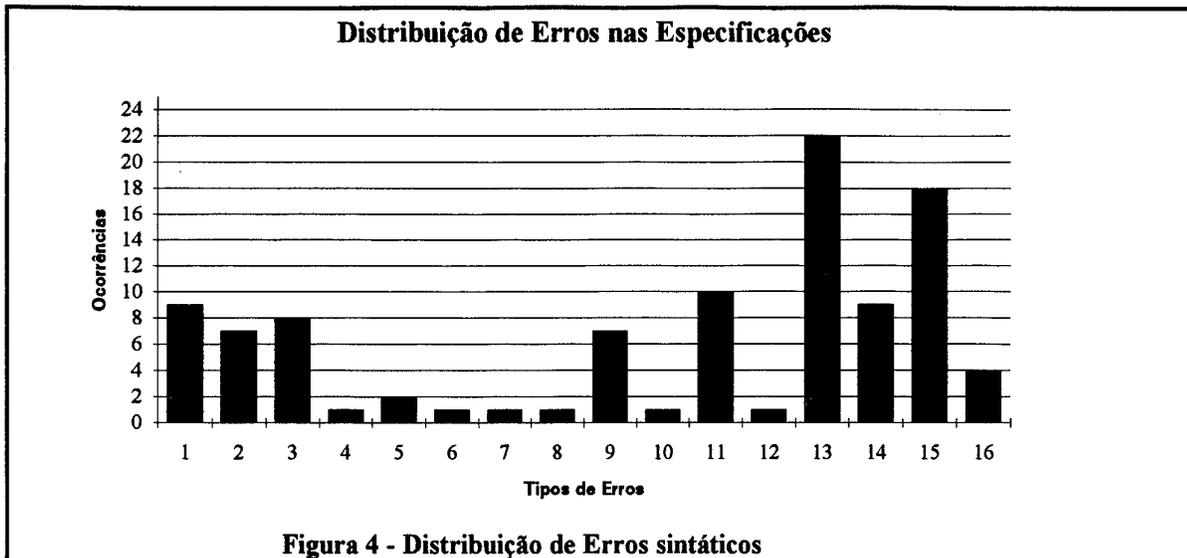
#### 4 - Resultados do processo de recuperação aplicado à amostra

Durante a aplicação do processo de recuperação de requisitos pudemos coletar vários dados sobre especificações estruturadas. Observamos que, mais importante do que a recuperação propriamente dita, eram os resultados obtidos com a aplicação do processo. Dentre os resultados obtidos destacamos :

- **Coleta de Erros Sintáticos**

<b>Tipo</b>	<b>Descrição</b>
	<b>Balanceamento</b>
01	Fluxo de dados desaparece na explosão/composição
02	Fluxo de dados troca o sentido ao se mudar o nível
03	Depósito de dados não aparece nos níveis inferiores
	<b>Depósitos de dados</b>
04	Acesso de consulta/recuperação com seta apontando p/ depósito
05	Fluxo de dados com descritor
06	Comunicação entre depósitos
07	Depósito "buraco negro"
08	Depósito "geração espontânea"
	<b>Fluxos de Dados</b>
09	Fluxo de dados sem seta
10	Fluxo de dados com dupla seta (não representa diálogo)
11	Fluxo de dados troca o descritor ao mudar de nível
12	Fluxo de dados s/ descritor (não é acesso)
	<b>Dicionário / mini-especificações</b>
13	Ausência de mini-especificação para processo primitivo
14	Elemento não dicionarizado
15	Dicionarização recursiva
	<b>Processos</b>
16	Numeração inconsistente

- **Distribuição de Erros Sintáticos**



- **Erros Semânticos**

Além dos erros sintáticos coletados, observamos que as especificações, em sua maioria, não representavam adequadamente as aplicações modeladas. É óbvio que uma afirmação desta natureza só pode ser feita se conhecermos os domínios envolvidos. Como os domínios são relativamente comuns, podemos tecer críticas sobre os aspectos semânticos das modelagens, em especial, quanto à fidedignidade dos elementos modelados em relação ao domínio.

Um esforço no sentido de se classificar os principais erros semânticos encontrados pode ser visto abaixo. Identificamos as quatro grandes categorias de erros semânticos :

**1) Identificação de Contexto**

São os erros onde o modelador não consegue distinguir adequadamente as fronteiras entre o sistema de software e o sistema mais abrangente onde este se insere. Exemplos desse tipo de erro podem ser ausência de entidades externas óbvias (em um domínio) e modelagem de elementos auxiliares (exemplo do garçon).

**2) Modelagem de aspectos de implementação**

É um problema bastante comum nas especificações que empregam as técnicas / notações "clássicas". Um exemplo que foi observado com certa incidência refere-se ao depósito de implementação, ou seja, um depósito de dados que, não precisaria estar necessariamente presente no modelo.

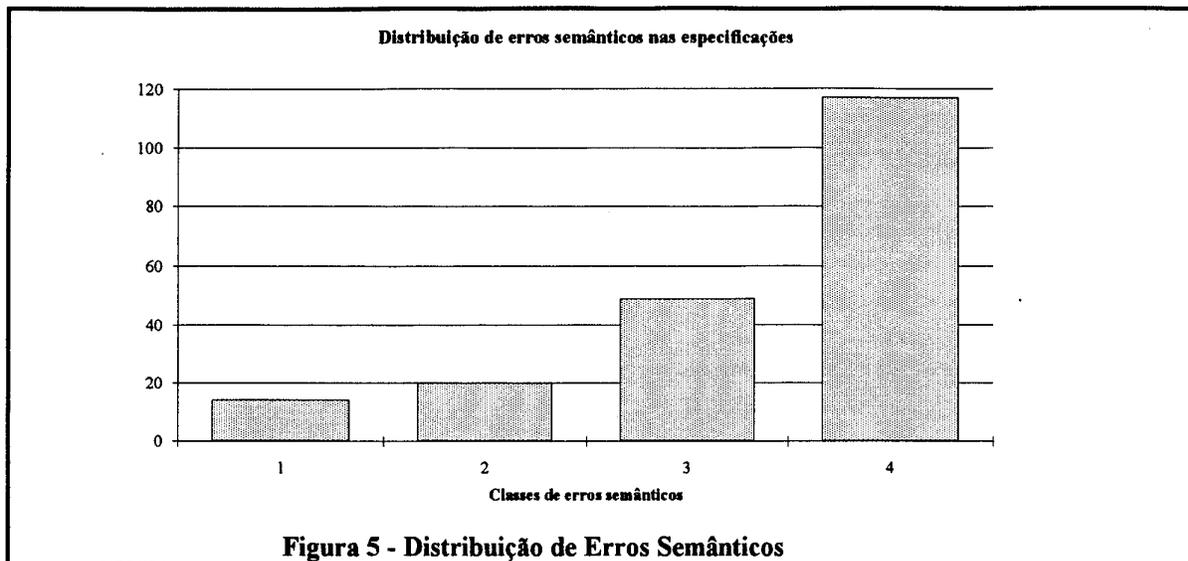
**3) Ausência de funcionalidade (em relação ao domínio)**

Refere-se à aspectos não modelados. Por exemplo, no domínio de videolocadoras, é fundamental a modelagem da devolução de uma fita alugada. Se o modelador não incorporar esta funcionalidade ao modelo, estamos diante de um exemplo de ausência de funcionalidade.

**4) Descritores com baixa carga semântica (Vocabulário Inadequado)**

Um erro relativamente comum, como poderemos observar mais adiante, na coleta de vocabulário. Exemplos deste tipo de erro são o uso excessivo de descritores semanticamente neutros em descrições de processo e descritores denotando comportamento ativo para fluxos de dados e depósitos.

- **Distribuição de Erros Semânticos**



- **Coleta de vocabulário**

Durante a aplicação do processo de recuperação, encontramos a oportunidade de coletar os verbos empregados como descritores de processos. Essa atividade, nos permitiu elaborar uma análise sobre a frequência de utilização dos verbos nos diversos domínios de aplicação da amostra.

Além dessa possível contribuição, podemos empregar o vocabulário coletado para aferir a qualidade das especificações. Como exemplo, podemos analisar a frequência de utilização de verbos vagos (CONTROLAR, TRATAR, PROCESSAR, MANIPULAR, GERAR) em uma especificação. Uma especificação que empregasse de forma excessiva<sup>2</sup> tais verbos, seria considerada de baixa qualidade.

Um outro exemplo seria a comparação da frequência de utilização de verbos em relação a um domínio. Consideremos o domínio postos de combustíveis. Se uma especificação de requisitos neste domínio não contiver em seus descritores de processo o verbo ABASTECER, provavelmente estaremos diante de um erro semântico do tipo ausência de funcionalidade.

**Relação de verbos em todos os domínios**

ABASTECER	02	AUTORIZAR	02
ABATER	01	AVALIAR	02
ACEITAR	01	CADASTRAR	55
ACUMULAR	03	CALCULAR	48
ADMINISTRAR	07	CHECAR	02
ALTERAR	05	CLASSIFICAR	01
ALUGAR	02	COBRAR	05
ANALISAR	01	COMPLETAR	01
APURAR	01	COMPRAR	08
ATENDER	20	COMPUTAR	01
ATUALIZAR	34	CONCLUIR	01

<sup>2</sup>É preciso observar que, em algumas situações, como por exemplo no diagrama que explode o diagrama de contexto, o modelador pode não encontrar verbos capazes de expressar adequadamente a funcionalidade (que se encontra agregada). Nestes caso, o uso de verbos vagos (como TRATAR) não pode ser considerado um erro de modelagem.

CONFECCIONAR	01	LIBERAR	02
CONFERIR	06	LIMPAR	01
CONFIRMAR	01	LISTAR	01
CONSULTAR	01	MEDIR	01
CONSUMIR	01	MODIFICAR	02
CONTABILIZAR	04	MOVIMENTAR	03
CONTROLAR	46	OBTER	14
CRITICAR	01	ORGANIZAR	02
DAR	03	PAGAR	04
DEFINIR	02	PASSAR	01
DETERMINAR	02	PEDIR	02
DEVOLVER	01	PERCORRER	01
EFETIVAR	01	PESQUISAR	10
EFETUAR	22	PREPARAR	01
EMITIR	51	PRIORIZAR	01
ENCAMINHAR	02	PROCEDER	01
ENCERRAR	01	PROCESSAR	08
ENCOMENDAR	01	PROCURAR	01
ENTREGAR	01	QUITAR	02
ENVIAR	05	REALIZAR	02
ESCOLHER	03	RECEBER	33
ESTABELECER	01	RECUPERAR	01
ESTIPULAR	01	REGISTRAR	05
ESTOCAR	01	RELACIONAR	01
EXCLUIR	04	REPOR	01
EXECUTAR	10	RETIRAR	01
EXPULSAR	02	SELECIONAR	17
FAZER	04	SOLICITAR	01
FECHAR	01	SOMAR	07
FORMATAR	03	SUBTRAIR	01
FORNECER	01	TOTALIZAR	07
GERAR	42	TRANSFORMAR	01
GERENCIAR	13	TRATAR	48
IDENTIFICAR	04	VALIDAR	26
IMPRIMIR	03	VENDER	05
INCLUIR	04	VERIFICAR	39
INFORMAR	01		
INSCREVER	01		
LER	01		

97 verbos distintos identificados.

#### Frequência de verbos em todos os domínios

55	CADASTRAR	33	RECEBER
51	EMITIR	26	VALIDAR
48	TRATAR	22	EFETUAR
48	CALCULAR	20	ATENDER
46	CONTROLAR	17	SELECIONAR
42	GERAR	14	OBTER
39	VERIFICAR	13	GERENCIAR
34	ATUALIZAR	10	PESQUISAR

10	EXECUTAR	01	SOLICITAR
08	PROCESSAR	01	RETIRAR
08	COMPRAR	01	REPOR
07	TOTALIZAR	01	RELACIONAR
07	SOMAR	01	RECUPERAR
07	ADMINISTRAR	01	PROCURAR
06	CONFERIR	01	PROCEDER
05	VENDER	01	PRIORIZAR
05	REGISTRAR	01	PREPARAR
05	ENVIAR	01	PERCORRER
05	COBRAR	01	PASSAR
05	ALTERAR	01	MEDIR
04	PAGAR	01	LISTAR
04	INCLUIR	01	LIMPAR
04	IDENTIFICAR	01	LER
04	FAZER	01	INSCREVER
04	EXCLUIR	01	INFORMAR
04	CONTABILIZAR	01	FORNECER
03	MOVIMENTAR	01	FECHAR
03	IMPRIMIR	01	ESTOCAR
03	FORMATAR	01	ESTIPULAR
03	ESCOLHER	01	ESTABELECER
03	DAR	01	ENTREGAR
03	ACUMULAR	01	ENCOMENDAR
02	REALIZAR	01	ENCERRAR
02	QUITAR	01	EFETIVAR
02	PEDIR	01	DEVOLVER
02	ORGANIZAR	01	CRITICAR
02	MODIFICAR	01	CONSUMIR
02	LIBERAR	01	CONSULTAR
02	EXPULSAR	01	CONFIRMAR
02	ENCAMINHAR	01	CONFECCIONAR
02	DETERMINAR	01	CONCLUIR
02	DEFINIR	01	COMPUTAR
02	CHECAR	01	COMPLETAR
02	AVALIAR	01	CLASSIFICAR
02	AUTORIZAR	01	APURAR
02	ALUGAR	01	ANALISAR
02	ABASTECER	01	ACEITAR
01	TRANSFORMAR	01	ABATER
01	SUBTRAIR		

- **Análise da coleta de vocabulário**

A atividade de coleta de vocabulário funcional permitiu que uma série de observações fossem realizadas. Uma das nossas observações refere-se ao cruzamento de vocabulário entre domínios. Através desta atividade, observamos que, alguns verbos não se repetiam entre domínios e, não poderiam ser classificados como verbos vagos. Neste caso, uma conclusão lógica seria a de que tais verbos são característicos de um determinado domínio.

Um exemplo deste raciocínio, pode ser observado através do verbo ABASTECER. Trata-se de um verbo que, só apareceu no domínio PC - Postos de Combustível. Como o domínio de Postos de Combustível é um domínio relativamente conhecido, o exemplo pode não parecer muito significativo mas, lembramos que, em domínios menos conhecidos, o procedimento que nos levou a concluir que ABASTECER pertence ao domínio de Postos de Combustível terá a devida utilidade.

O exemplo acima nos motivou a produzir uma lista de referências cruzadas de verbos entre os domínios da amostra. Com este procedimento, conseguimos produzir relações de verbos característicos de um domínio. A identificação dos verbos é trivial. Para cada domínio, listamos apenas os verbos que não aparecem em outros domínios da amostra. É importante observar que, tal procedimento atua como uma espécie de filtro, eliminando parte dos verbos vagos (TRATAR, PROCESSAR, MANIPULAR), pois existe uma tendência, entre os modeladores, de empregar tais verbos para descrever a funcionalidade agregada nos níveis mais elevados de abstração, consequentemente produzindo repetições destes verbos entre os domínios.

#### **Verbos exclusivos do domínio (Exemplos)**

##### **VL - VIDEOLOCADORAS**

ALUGAR  
DAR  
EFETIVAR  
ENCOMENDAR  
ENTREGAR  
ESTIPULAR  
EXPULSAR  
INSCREVER  
LIMPAR  
RELACIONAR

##### **PC - POSTOS DE COMBUSTÍVEL**

ABASTECER  
CLASSIFICAR  
MEDIR  
ORGANIZAR  
PREPARAR  
REPOR  
VENDER

É importante observar que, alguns domínios não tiveram vocabulário exclusivo associado, denotando uma fraca modelagem ou a presença de poucos exemplares significativos na amostra ou ambos. Alguns verbos se destacaram pela sua elevada incidência em diversos domínios. Identificamos duas classes de verbos que se apresentaram com alta incidência. A primeira classe é composta pelos verbos de significado vago. Neste caso, sua presença em diversos domínios pode ser justificada por dois motivos :

- a) Estão sendo empregados em um nível elevado de abstração.
- b) Estão sendo empregados indiscriminadamente, denotando uma falha de modelagem.

A segunda classe de verbos com elevada incidência, pode ser explicada pelo conceito de hierarquia de domínio. Domínios podem ser vistos como uma estrutura hierárquica, onde os domínios de mais elevado nível de abstração invocam domínios mais primitivos e assim sucessivamente. Logo, os verbos não-vagos, que aparecem em mais de um domínio podem sinalizar a presença de um sub-domínio comum aos domínios onde ocorre a repetição.

Apresentaremos, a seguir, os verbos que se repetiram nos domínios da amostra, indicando em quais domínios apareceram :

ACUMULAR	:	VL, RE
ADMINISTRAR	:	RE, AP, FI
ALTERAR	:	RE, PC, AP, BI
ATENDER	:	VL, RE, PC, CN, MA
ATUALIZAR	:	VL, RE, PC, CD, AP, DI, CN, BI
AUTORIZAR	:	VL, RE
CADASTRAR	:	VL, RE, PC, CD, AP, DI, FI, CN, BI, MA
CALCULAR	:	VL, RE, CD, DI, FI, CN
COBRAR	:	VL, CD
COMPRAR	:	VL, RE, PC
CONFERIR	:	RE, PC, MA
CONTABILIZAR	:	VL, PC, FI
CONTROLAR	:	VL, RE, PC, CD, DI, FI, CN
DEFINIR	:	RE, PC
DETERMINAR	:	PC, CD
EFETUAR	:	VL, RE, PC, CD, CN
EMITIR	:	VL, RE, PC, CD, AP, DI, FI, CN, MA
ENVIAR	:	VL, DI
EXCLUIR	:	PC, DI, BI
ESCOLHER	:	VL, RE
EXECUTAR	:	VL, RE
FAZER	:	VL, RE, PC
GERAR	:	VL, RE, PC, CD, AP, FI, MA
GERENCIAR	:	PC, DI, FI, MA
IMPRIMIR	:	RE, PC, BI
INCLUIR	:	PC, DI, BI
MOVIMENTAR	:	RE, CD, DI
OBTER	:	VL, RE, CD
PAGAR	:	VL, CD
PEDIR	:	VL, RE
PESQUISAR	:	VL, RE, BI
PROCESSAR	:	VL, MA
REALIZAR	:	VL, RE
RECEBER	:	VL, RE, PC, CD, FI, MA
REGISTRAR	:	VL, AP
SELECIONAR	:	VL, RE, CD, AP, DI, BI, MA
SOMAR	:	RE, PC, CD, CN
TOTALIZAR	:	VL, RE, PC, CD, FI
TRATAR	:	VL, RE, PC
VALIDAR	:	VL, RE, CN, BI
VERIFICAR	:	VL, RE, PC, CD, AP, DI, MA

Uma observação interessante refere-se ao verbo CADASTRAR. Foi o único verbo que esteve presente em todos os domínios da amostra. A operação de cadastramento é uma operação típica de um subdomínio, compartilhado pelos diversos domínios da amostra.

Ainda considerando a incidência de determinados verbos, apresentamos, os **10 verbos** que mais foram empregados, em toda a amostra, de um total de 97 verbos distintos identificados, juntamente com suas frequências de aparição :

<b>55</b>	<b>CADASTRAR</b>	<b>42</b>	<b>GERAR</b>
<b>51</b>	<b>EMITIR</b>	<b>39</b>	<b>VERIFICAR</b>
<b>48</b>	<b>TRATAR</b>	<b>34</b>	<b>ATUALIZAR</b>
<b>48</b>	<b>CALCULAR</b>	<b>33</b>	<b>RECEBER</b>
<b>46</b>	<b>CONTROLAR</b>	<b>26</b>	<b>VALIDAR</b>

É interessante observar que, grande parte da lista apresenta verbos excessivamente vagos, isto é de pouca carga semântica. Verbos como TRATAR, CONTROLAR e GERAR, independente de seu complemento pouco informam sobre a natureza da função modelada. Os demais verbos podem ou não ser vagos, dependendo do seu complemento. Nossa análise das especificações da amostra indica que, normalmente os modeladores não empregavam complementos capazes de aumentar a carga semântica da descrição. Este fenômeno tornou-se particularmente visível quando elaborávamos as sentenças de requisitos.

## **5 - Conclusão**

Acreditamos que, o processo para recuperação de requisitos descrito neste trabalho deva ser empregado em maior escala, para que novos dados possam ser incorporados aos já coletados. Dentre os produtos do processo de recuperação, destacamos a coleta de vocabulário, particularmente útil para realizar observações sobre o problema da construção de domínios[9].

O processo para recuperação de requisitos proposto pode ser parcialmente automatizado[10]. A coleta de vocabulário funcional é uma atividade cuja implementação é relativamente simples, em um ambiente CASE como TALISMAN [8]. Outras atividades podem ser implementadas, porém o processo exige intervenção de um ator inteligente nas atividades que envolvam análise semântica. Vemos como trabalhos futuros a condução de outras experiências de recuperação, e a automação parcial do processo, através de uma ferramenta CASE programável, como TALISMAN.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

- [1] Bansler, Jorgen P.; Bodker, Keld; "A Reappraisal of Structured Analysis: Design in an Organizational Context"; ACM Transactions on Information Systems, vol. 11, Nº 2, pp. 165-193, April 1993
- [2] de Marco, Tom; "Structure Systems Analysis and System Specification"; Yourdon Inc.; 1978
- [3] Gane, Crish; Sarson, Trish; "Structured Systems Analysis: Tools and Techniques"; Prentice-Hall Inc.; 1979
- [4] Leite, J.C.S.P.; "A Case Study on Requirements Recovery from Structured Specifications"; Monografias em Ciência da Computação; PUC-Rio; 1992

- [5] Leite, J.C.S.P.; **"Enhancing the Semantics of Requirements Statements"**; Monografias em Ciência da Computação; PUC-Rio; 1992
- [6] Maffeo, Bruno; **"Engenharia de Software e Especificação de Sistemas"**, Campus; 1992
- [7] McMenamin, Stephen; **"Essential Systems Analysis"**; McGraw-Hill, 1986
- [8] Staa, Arndt von; **"Ambiente de Engenharia de Software Assistido por Computador - TALISMAN - Manual de Referência - Versão 4"**; Staa Informática, 1992
- [9] Arango, G.; **"Domain Engineering for Software Reuse"**; Dissertation submitted in partial satisfaction for the degree Doctor of Philosophy in Information and Computer Science; University of California - Irvine; 1988.
- [10] Cerqueira, Paulo M.; **"Um processo para recuperação de requisitos a partir de especificações estruturadas"**; Dissertação submetida ao Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro, como parte dos requisitos para a obtenção do grau de Mestre em Ciências : Informática; 1994.