

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**ENGENHARIA REVERSA DE UMA APLICAÇÃO DE**  
**GESTÃO DE PROCESSOS JUDICIAIS**

**CHRISTIAN MARCEL KLUG**

**BLUMENAU**  
**2008**

**2008/1-03**



**CHRISTIAN MARCEL KLUG**

**ENGENHARIA REVERSA DE UMA APLICAÇÃO DE  
GESTÃO DE PROCESSOS JUDICIAIS**

Trabalho de Conclusão de Curso submetido à  
Universidade Regional de Blumenau para a  
obtenção dos créditos na disciplina Trabalho  
de Conclusão de Curso II do curso de Sistemas  
de Informação — Bacharelado.

Prof .Everaldo Artur Grahl, Ms - Orientador

**BLUMENAU  
2008**

**2008/1-03**

# **ENGENHARIA REVERSA DE UM SISTEMA DE GESTÃO DE PROCESSOS JUDICIAIS**

Por

**CHRISTIAN MARCEL KLUG**

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Everaldo Artur Grahl, Ms – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Marcel Hugo, Ms – FURB

Membro: \_\_\_\_\_  
Prof. Alexander Roberto Valdameri, Ms – FURB

Blumenau, 30 de junho de 2008

Dedico este trabalho a meus pais por terem me proporcionado a oportunidade de ingressar na universidade.

## **AGRADECIMENTOS**

A Deus, pelo seu imenso amor e graça.

À minha família e minha namorada que sempre estiveram ao meu lado.

Ao meu orientador, Everaldo Artur Grahl por ter acreditado na conclusão deste trabalho.

“Não tá morto quem pelega.”

Autor desconhecido

## RESUMO

O presente trabalho tem como objetivo realizar a engenharia reversa de um sistema de Gestão de Processos Judiciais através de documentação *Unified Modeling Language* (UML). O sistema Gestão de Processos Judiciais tem objetivo de gerenciar os processos de uma empresa através de controles de andamentos processuais, pedidos, depósitos, penhoras, tributos, lançamentos financeiros, faturas etc. Neste trabalho são documentados os requisitos, Objetivos, Diagramas de Processo, Diagramas de Casos de Uso, Diagrama de Atividades e Diagrama Entidade Relacionamento, a documentação é feita utilizando os pacotes da UML. Para a documentação do Diagrama Entidade Relacionamento foi construído uma rotina de leituras de tabelas e campos da base para gerar automaticamente este diagrama.

Palavras-chave: Engenharia Reversa. Gestão de Processos Judiciais. UML.

## **ABSTRACT**

The present work aims accomplish the reverse engineering of a system of process management through /Unified Modeling Language/UML documentation. The system of process management aims to manage the process f a company through control of process proceeding, petition, deposit, distress, impost, financial registration, bills etc. In that work are documented the requirement, goal, process diagram, use case diagram, activity diagram and entity relationship diagram, the documentation is done using the packages of UML. For documentation of entity relationship diagram was constructed a routine of reading of tables and fields of the base generating automatically this diagram.

Key-words: Reverse engineering. Judiciary process management. UML.

## LISTA DE ILUSTRAÇÕES

FIGURA 1 – CADASTRO DE PROCESSOS.....	17
FIGURA 2 – FERRAMENTA BUILDER.....	21
QUADRO 1 – PADRÕES DE ENGENHARIA REVERSA.....	23
FIGURA 3 – EXEMPLO DO MODELO DE OBJETIVOS.....	24
FIGURA 4 – DIAGRAMA DE PROCESSOS EXEMPLO.....	25
FIGURA 5 – EXEMPLO DE DIAGRAMA ENTIDADE RELACIONAMENTO.....	26
FIGURA 6 – DIAGRAMA DE ATIVIDADES EXEMPLO.....	27
FIGURA 7 – DIAGRAMA CASOS DE USO EXEMPLO.....	28
FIGURA 8 – EXEMPLO DE DIAGRAMAS SEPARADOS EM PACOTES.....	29
FIGURA 9 – TELA DO MÓDULO FINANCEIRO.....	30
FIGURA 10 – TELA DO MÓDULO FINANCEIRO.....	32
QUADRO 2 – REQUISITOS FUNCIONAIS.....	34
QUADRO 3 – REQUISITOS NÃO FUNCIONAIS.....	35
FIGURA 11 – DIAGRAMAS DE CASOS DE USO DO SISTEMA.....	36
QUADRO 4 – CASO DE USO REGISTRA ROTINA.....	36
QUADRO 5 – CASO DE USO REGISTRA TABELAS.....	37
QUADRO 6 – CASO DE USO PESQUISA TABELAS.....	37
QUADRO 7 – CASO DE USO CADASTRO DE ROTINAS.....	37
QUADRO 8 – ENTIDADE ROTINAS.....	38
QUADRO 9 – ENTIDADE ROTINAS E TABELAS.....	38
QUADRO 10 – ENTIDADE TABELAS.....	38
QUADRO 11 – TRECHO DE CÓDIGO.....	40
FIGURA 12 – CADASTRO DE ROTINAS.....	42
FIGURA 13 – CADASTRO DE TABELAS.....	43
FIGURA 14 – CADASTRO DE ROTINAS EXEMPLO.....	44
QUADRO 12 – ESTADO INICIAL DA TABELA.....	44
QUADRO 13 – ESTADO DA TABELA NO SEGUNDO NÍVEL.....	45
FIGURA 15 – CADASTRO DE TABELAS EXEMPLO.....	46
QUADRO 14 – ARQUIVO GERADO.....	47
FIGURA 16 – OPÇÃO PARA IMPORTAÇÃO DE ARQUIVO DELPHI.....	48
FIGURA 16 – OPÇÃO PARA IMPORTAÇÃO DE ARQUIVO DELPHI.....	48

<b>FIGURA 17 – SELECIONANDO ARQUIVO GERADO.....</b>	<b>48</b>
<b>FIGURA 17 – SELECIONANDO ARQUIVO GERADO.....</b>	<b>48</b>
<b>FIGURA 18 – DIAGRAMA ENTIDADE RELACIONAMENTO EXEMPLO.....</b>	<b>49</b>
<b>FIGURA 18 – DIAGRAMA ENTIDADE RELACIONAMENTO EXEMPLO.....</b>	<b>49</b>
<b>FIGURA 19 – FIGURA PARA DEMONSTRAÇÃO.....</b>	<b>50</b>
<b>FIGURA 19 – FIGURA PARA DEMONSTRAÇÃO.....</b>	<b>50</b>
<b>FIGURA 20 – VISÃO GERAL DO EA COM A DOCUMENTAÇÃO CONCLUÍDA..</b>	<b>52</b>
<b>FIGURA 20 – VISÃO GERAL DO EA COM A DOCUMENTAÇÃO CONCLUÍDA..</b>	<b>52</b>
<b>FIGURA 21 – MODELO DE OBJETIVOS DO MÓDULO JURÍDICO.....</b>	<b>53</b>
<b>FIGURA 21 – MODELO DE OBJETIVOS DO MÓDULO JURÍDICO.....</b>	<b>53</b>
<b>FIGURA 22 – DIAGRAMA DE PROCESSOS DO MÓDULO JURÍDICO.....</b>	<b>54</b>
<b>FIGURA 23 – DIAGRAMA DE ATIVIDADES DO FECHAMENTO DE PROVISÃO</b>	<b>56</b>
<b>FIGURA 24 – DIAGRAMA DE CASOS DE USO DO FECHAMENTO DA PROVISÃO</b>	<b>57</b>
<b>.....</b>	<b>57</b>
<b>QUADRO 15 – CENÁRIOS DO CASO DE USO EXECUTA FECHAMENTO.....</b>	<b>57</b>
<b>QUADRO 16 – CENÁRIOS DO CASO DE USO CANCELA FECHAMENTO.....</b>	<b>57</b>
<b>FIGURA 25 – REQUISITOS FUNCIONAIS DO FECHAMENTO DE PROVISÃO</b>	<b>58</b>
<b>.....</b>	<b>58</b>
<b>FIGURA 26 – DIAGRAMA ENTIDADE RELACIONAMENTO DO FECHAMENTO</b>	<b>58</b>
<b>DE PROVISÃO.....</b>	<b>58</b>
<b>FIGURA 27 – DIAGRAMA DE CASOS DE USO DE OBRIGAÇÕES.....</b>	<b>59</b>
<b>FIGURA 28 – REQUISITOS FUNCIONAIS DE OBRIGAÇÕES.....</b>	<b>59</b>
<b>FIGURA 29 – DIAGRAMA DE ATIVIDADES DE PEDIDOS.....</b>	<b>60</b>
<b>FIGURA 30 – DIAGRAMA DE CASOS DE USO DE PEDIDOS.....</b>	<b>61</b>
<b>FIGURA 31 – REQUISITOS FUNCIONAIS DA ROTINA DE PEDIDOS.....</b>	<b>61</b>
<b>FIGURA 32 – DIAGRAMA ENTIDADE RELACIONAMENTO DE PEDIDOS.....</b>	<b>62</b>
<b>FIGURA 33 – DIAGRAMA DE ATIVIDADES DE REMESSA DE PASTAS.....</b>	<b>64</b>
<b>FIGURA 34 – DIAGRAMA DE CASOS DE USO DE REMESSA.....</b>	<b>65</b>
<b>FIGURA 35 – DIAGRAMA ENTIDADE RELACIONAMENTO DE REMESSAS.....</b>	<b>66</b>
<b>FIGURA 36 – DIAGRAMA DE ATIVIDADES DE PROVIDÊNCIAS.....</b>	<b>67</b>
<b>FIGURA 37 – DIAGRAMA DE CASOS DE USO DE PROVIDÊNCIAS.....</b>	<b>68</b>
<b>QUADRO 17 – CASO DE USO CANCELA PROVIDÊNCIA.....</b>	<b>68</b>
<b>QUADRO 18 – CASO DE USO CUMPRIMENTO DE PROVIDÊNCIA.....</b>	<b>68</b>
<b>QUADRO 19 – CASO DE USO PRORROGA PROVIDÊNCIA.....</b>	<b>69</b>

<b>QUADRO 20 – CASO DE USO REGISTRA E ENCAMINHA PROVIDÊNCIA.....</b>	<b>69</b>
<b>QUADRO 21 – CENÁRIOS DO CASO DE USO TRANSFERIR PROVIDÊNCIA.....</b>	<b>69</b>
<b>FIGURA 38 – MODELO DE OBJETIVOS DO FINANCEIRO.....</b>	<b>70</b>
<b>FIGURA 39 – DIAGRAMA DE PROCESSOS DO FINANCEIRO.....</b>	<b>71</b>
<b>FIGURA 40 – DIAGRAMA DE CASOS DE USO DOS CONTRATOS.....</b>	<b>72</b>
<b>FIGURA 41 – REQUISITOS FUNCIONAIS DOS CONTRATOS.....</b>	<b>72</b>
<b>FIGURA 42 – DIAGRAMA DE ATIVIDADES DO FATURAMENTO.....</b>	<b>73</b>
<b>FIGURA 43 – DIAGRAMA DE CASOS DE USO DO FATURAMENTO.....</b>	<b>74</b>
<b>FIGURA 44 – DIAGRAMA ENTIDADE RELACIONAMENTO DO FATURAMENTO .....</b>	<b>75</b>
<b>FIGURA 45 – DOCUMENTAÇÃO HTML GERADA PELO EA.....</b>	<b>76</b>
<b>QUADRO 22 – QUANTIDADE DE COMPONENTES UTILIZADOS.....</b>	<b>77</b>



## LISTA DE SIGLAS

DER – *Diagrama Entidade Relacionamento*

FK – *Foreign Key* (chave estrangeira)

HTML – *HyperText Markup Language*

PK – *Primary Key* (chave primária)

SQL - *Structured Query Language*

UML - *Unified Modeling Language*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>17</b>
1.1 OBJETIVOS DO TRABALHO.....	18
1.2 ESTRUTURA DO TRABALHO.....	19
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>20</b>
2.1 EMPRESA BENNER SISTEMAS.....	20
<b>FIGURA 2 – FERRAMENTA BUILDER.....</b>	<b>21</b>
2.2 ENGENHARIA REVERSA.....	22
<b>QUADRO 1 – PADRÕES DE ENGENHARIA REVERSA.....</b>	<b>23</b>
<b>FONTE: ADAPTADO DE PERES (2003).....</b>	<b>23</b>
2.3 MODELAGEM.....	23
2.3.1 MODELO DE OBJETIVOS.....	24
<b>FIGURA 3 – EXEMPLO DO MODELO DE OBJETIVOS.....</b>	<b>24</b>
2.3.2 DIAGRAMA DE PROCESSOS.....	24
<b>FIGURA 4 – DIAGRAMA DE PROCESSOS EXEMPLO.....</b>	<b>25</b>
2.3.3 REQUISITOS FUNCIONAIS.....	25
2.3.4 DIAGRAMA ENTIDADE RELACIONAMENTO.....	26
2.3.5 UML.....	26
2.3.5.1 DIAGRAMA DE ATIVIDADES.....	27
2.3.5.2 DIAGRAMA DE CASOS DE USO.....	28
2.3.5.3 DIAGRAMA DE PACOTES.....	28
2.4 ESTRUTURA BÁSICA DO SISTEMA GESTÃO DE PROCESSOS.....	29
2.4.1 MÓDULO JURÍDICO.....	29
<b>FIGURA 9 – TELA DO MÓDULO FINANCEIRO.....</b>	<b>30</b>
2.4.2 MÓDULO FINANCEIRO.....	32
2.5 TRABALHOS CORRELATOS.....	33
<b>3 DESENVOLVIMENTO.....</b>	<b>34</b>
3.1 REQUISITOS DO SISTEMA.....	34
3.1.1 REQUISITOS FUNCIONAIS.....	34
<b>QUADRO 2 – REQUISITOS FUNCIONAIS.....</b>	<b>34</b>
3.1.2 REQUISITOS NÃO FUNCIONAIS.....	35
<b>QUADRO 3 – REQUISITOS NÃO FUNCIONAIS.....</b>	<b>35</b>

<a href="#">3.2 ESPECIFICAÇÃO.....</a>	<a href="#">35</a>
<a href="#">3.2.1 CASOS DE USO.....</a>	<a href="#">35</a>
<b>FIGURA 11 – DIAGRAMAS DE CASOS DE USO DO SISTEMA.....</b>	<b>36</b>
<a href="#">3.2.1.1 CENÁRIOS DOS CASOS DE USO.....</a>	<a href="#">36</a>
<b>QUADRO 4 – CASO DE USO REGISTRA ROTINA.....</b>	<b>36</b>
<b>QUADRO 5 – CASO DE USO REGISTRA TABELAS.....</b>	<b>37</b>
<b>QUADRO 6 – CASO DE USO PESQUISA TABELAS.....</b>	<b>37</b>
<b>QUADRO 7 – CASO DE USO CADASTRO DE ROTINAS.....</b>	<b>37</b>
<a href="#">3.2.2 DICIONÁRIO DE DADOS.....</a>	<a href="#">37</a>
<b>QUADRO 8 – ENTIDADE ROTINAS.....</b>	<b>38</b>
<b>QUADRO 9 – ENTIDADE ROTINAS E TABELAS.....</b>	<b>38</b>
<b>QUADRO 10 – ENTIDADE TABELAS.....</b>	<b>38</b>
<a href="#">3.3 IMPLEMENTAÇÃO.....</a>	<a href="#">38</a>
<b>QUADRO 11 – TRECHO DE CÓDIGO.....</b>	<b>40</b>
<a href="#">3.4 OPERACIONALIDADE DO SISTEMA.....</a>	<a href="#">41</a>
<a href="#">3.4.1 CADASTRO DE ROTINAS .....</a>	<a href="#">41</a>
<b>FIGURA 12 – CADASTRO DE ROTINAS.....</b>	<b>42</b>
<a href="#">3.4.2 CADASTRO DE TABELAS.....</a>	<a href="#">42</a>
<b>FIGURA 13 – CADASTRO DE TABELAS.....</b>	<b>43</b>
<a href="#">3.4.3 EXEMPLO DE FUNCIONAMENTO DA ROTINA.....</a>	<a href="#">43</a>
<b>FIGURA 14 – CADASTRO DE ROTINAS EXEMPLO.....</b>	<b>44</b>
<b>QUADRO 12 – ESTADO INICIAL DA TABELA.....</b>	<b>44</b>
<b>QUADRO 13 – ESTADO DA TABELA NO SEGUNDO NÍVEL.....</b>	<b>45</b>
<b>FIGURA 15 – CADASTRO DE TABELAS EXEMPLO.....</b>	<b>46</b>
<b>QUADRO 14 – ARQUIVO GERADO.....</b>	<b>47</b>
<b>FIGURA 16 – OPÇÃO PARA IMPORTAÇÃO DE ARQUIVO DELPHI.....</b>	<b>48</b>
<b>FIGURA 17 – SELECIONANDO ARQUIVO GERADO.....</b>	<b>48</b>
<b>FIGURA 18 – DIAGRAMA ENTIDADE RELACIONAMENTO EXEMPLO.....</b>	<b>49</b>
<b>FIGURA 19 – FIGURA PARA DEMONSTRAÇÃO.....</b>	<b>50</b>
<a href="#">3.5 DESENVOLVIMENTO DOS DIAGRAMAS .....</a>	<a href="#">51</a>
<b>FIGURA 20 – VISÃO GERAL DO EA COM A DOCUMENTAÇÃO CONCLUÍDA. .</b>	<b>52</b>
<a href="#">3.5.1 DIAGRAMAS DO MÓDULO JURÍDICO.....</a>	<a href="#">52</a>
<a href="#">3.5.1.1 DIAGRAMA DE OBJETIVOS.....</a>	<a href="#">52</a>
<b>FIGURA 21 – MODELO DE OBJETIVOS DO MÓDULO JURÍDICO.....</b>	<b>53</b>

<a href="#">3.5.1.2 DIAGRAMA DE PROCESSOS.....</a>	<a href="#">53</a>
<b>FIGURA 22 – DIAGRAMA DE PROCESSOS DO MÓDULO JURÍDICO.....</b>	<b>54</b>
<a href="#">3.5.2 ROTINAS DO MÓDULO JURÍDICO.....</a>	<a href="#">55</a>
<a href="#">3.5.2.1 FECHAMENTO DE PROVISÃO.....</a>	<a href="#">55</a>
<b>FIGURA 24 – DIAGRAMA DE CASOS DE USO DO FECHAMENTO DA PROVISÃO.....</b>	<b>57</b>
<b>QUADRO 15 – CENÁRIOS DO CASO DE USO EXECUTA FECHAMENTO.....</b>	<b>57</b>
<b>QUADRO 16 – CENÁRIOS DO CASO DE USO CANCELA FECHAMENTO.....</b>	<b>57</b>
<b>FIGURA 26 – DIAGRAMA ENTIDADE RELACIONAMENTO DO FECHAMENTO DE PROVISÃO.....</b>	<b>58</b>
<a href="#">3.5.2.2 OBRIGAÇÕES.....</a>	<a href="#">59</a>
<b>FIGURA 27 – DIAGRAMA DE CASOS DE USO DE OBRIGAÇÕES.....</b>	<b>59</b>
<b>FIGURA 28 – REQUISITOS FUNCIONAIS DE OBRIGAÇÕES.....</b>	<b>59</b>
<a href="#">3.5.2.3 PEDIDOS.....</a>	<a href="#">60</a>
<b>FIGURA 29 – DIAGRAMA DE ATIVIDADES DE PEDIDOS.....</b>	<b>60</b>
<b>FIGURA 30 – DIAGRAMA DE CASOS DE USO DE PEDIDOS.....</b>	<b>61</b>
<b>FIGURA 31 – REQUISITOS FUNCIONAIS DA ROTINA DE PEDIDOS.....</b>	<b>61</b>
<b>FIGURA 32 – DIAGRAMA ENTIDADE RELACIONAMENTO DE PEDIDOS.....</b>	<b>62</b>
<a href="#">3.5.2.4 REMESSA E RECEPÇÃO DE PASTAS.....</a>	<a href="#">63</a>
<b>FIGURA 33 – DIAGRAMA DE ATIVIDADES DE REMESSA DE PASTAS.....</b>	<b>64</b>
<b>FIGURA 34 – DIAGRAMA DE CASOS DE USO DE REMESSA.....</b>	<b>65</b>
<b>FIGURA 35 – DIAGRAMA ENTIDADE RELACIONAMENTO DE REMESSAS.....</b>	<b>66</b>
<a href="#">3.5.2.5 EVENTOS E PROVIDÊNCIAS.....</a>	<a href="#">66</a>
<b>FIGURA 36 – DIAGRAMA DE ATIVIDADES DE PROVIDÊNCIAS.....</b>	<b>67</b>
<b>FIGURA 37 – DIAGRAMA DE CASOS DE USO DE PROVIDÊNCIAS.....</b>	<b>68</b>
<b>QUADRO 17 – CASO DE USO CANCELA PROVIDÊNCIA.....</b>	<b>68</b>
<b>QUADRO 18 – CASO DE USO CUMPRIMENTO DE PROVIDÊNCIA.....</b>	<b>68</b>
<b>QUADRO 19 – CASO DE USO PRORROGA PROVIDÊNCIA.....</b>	<b>69</b>
<b>QUADRO 20 – CASO DE USO REGISTRA E ENCAMINHA PROVIDÊNCIA.....</b>	<b>69</b>
<b>QUADRO 21 – CENÁRIOS DO CASO DE USO TRANSFERIR PROVIDÊNCIA.....</b>	<b>69</b>
<a href="#">3.5.3 MÓDULO FINANCEIRO.....</a>	<a href="#">69</a>
<a href="#">3.5.3.1 MODELO DE OBJETIVOS.....</a>	<a href="#">70</a>
<b>FIGURA 38 – MODELO DE OBJETIVOS DO FINANCEIRO.....</b>	<b>70</b>
<a href="#">3.5.3.2 DIAGRAMA DE PROCESSOS.....</a>	<a href="#">70</a>

<b>FIGURA 39 – DIAGRAMA DE PROCESSOS DO FINANCEIRO.....</b>	<b>71</b>
<b><u>3.5.4 ROTINAS DO MÓDULO FINANCEIRO.....</u></b>	<b><u>71</u></b>
<b><u>3.5.4.1 CONTRATOS COM ESCRITÓRIOS EXTERNOS.....</u></b>	<b><u>71</u></b>
<b>FIGURA 40 – DIAGRAMA DE CASOS DE USO DOS CONTRATOS.....</b>	<b>72</b>
<b>FIGURA 41 – REQUISITOS FUNCIONAIS DOS CONTRATOS.....</b>	<b>72</b>
<b><u>3.5.4.2 FATURAMENTOS.....</u></b>	<b><u>72</u></b>
<b>FIGURA 42 – DIAGRAMA DE ATIVIDADES DO FATURAMENTO.....</b>	<b>73</b>
<b>FIGURA 43 – DIAGRAMA DE CASOS DE USO DO FATURAMENTO.....</b>	<b>74</b>
<b>FIGURA 44 – DIAGRAMA ENTIDADE RELACIONAMENTO DO FATURAMENTO .....</b>	<b>75</b>
<b><u>3.6 DOCUMENTAÇÃO HTML.....</u></b>	<b><u>75</u></b>
<b>FIGURA 45 – DOCUMENTAÇÃO HTML GERADA PELO EA.....</b>	<b>76</b>
<b><u>3.7 RESULTADOS E DISCUSSÃO.....</u></b>	<b><u>77</u></b>
<b>QUADRO 22 – QUANTIDADE DE COMPONENTES UTILIZADOS.....</b>	<b>77</b>
<b><u>4 CONCLUSÕES.....</u></b>	<b><u>79</u></b>
<b><u>4.1 EXTENSÕES.....</u></b>	<b><u>80</u></b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>81</b>

## 1 INTRODUÇÃO

Um software muitas vezes não passa pelo ciclo ideal de desenvolvimento, sendo implementado sem que exista um processo adequado de análise. Isto ocorre por diversos fatores como falta de recursos humanos e pouco tempo para o desenvolvimento. Esta prática na maioria dos casos ocasiona uma série de problemas e muitas vezes é necessário refazer boa parte, ou até mesmo remodelar o sistema. Além disso, em outros casos realizar uma engenharia reversa para obter alguma documentação do sistema nem sempre é fácil, principalmente se o sistema foi concebido a partir de um gerador de código.

Foi caso do sistema de Gestão de Processos Judiciais da empresa Benner desenvolvido com a ferramenta Builder usando banco de dados SQL Server. Este sistema não possuía documentação. O sistema de Gestão de Processos tem como objetivo o controle de processos, incluindo cadastros de desdobramentos, eventos, providências, pedidos, depósitos, penhoras e outros. Este sistema é destinado a departamentos jurídicos de empresas, entre os clientes estão Telefônica, Mercedes, Bayer, Santander, Kaiser e HSBC. A Figura 1 mostra uma tela de cadastro do sistema.

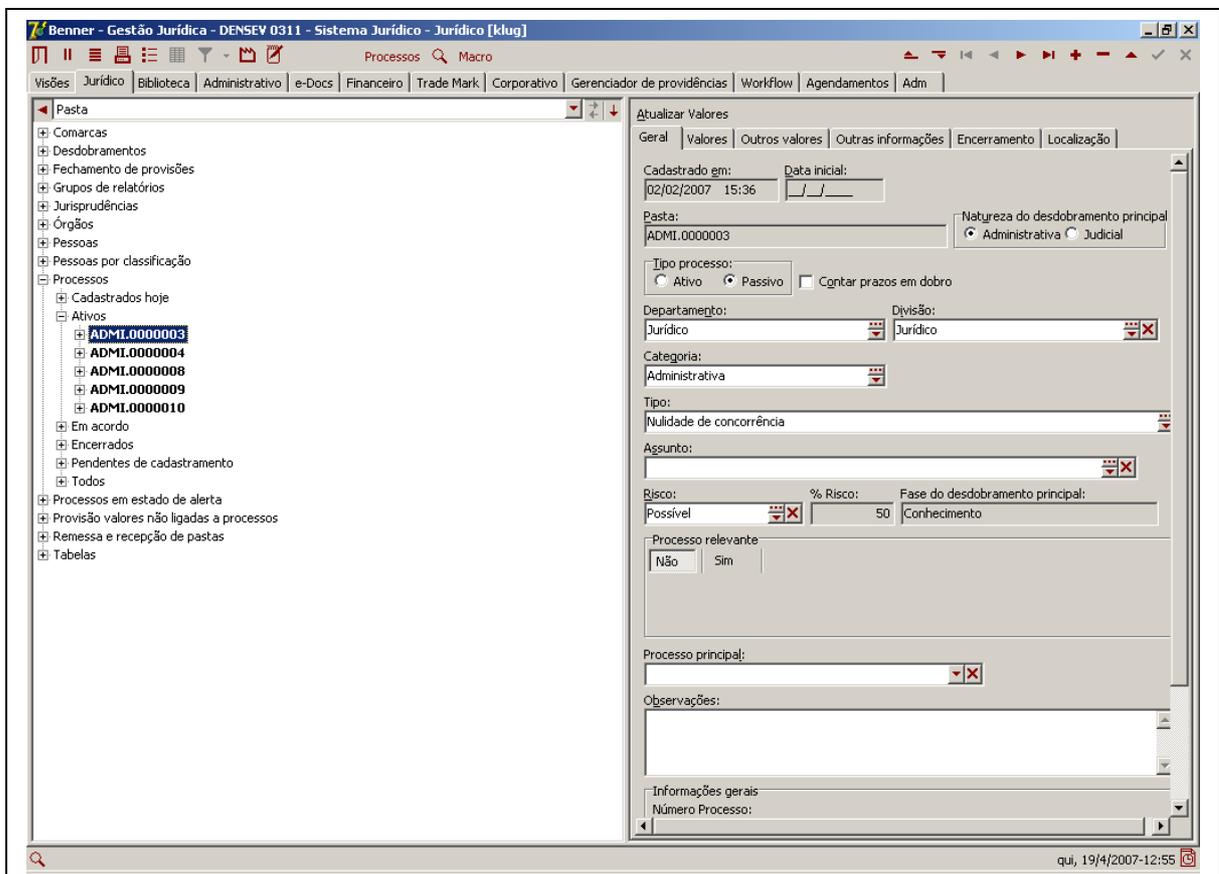


Figura 1 – Cadastro de processos

Após a saída dos programadores e analista que começaram o desenvolvimento do sistema começou a ficar muito complicado a manutenção de rotinas antigas.

Tentando resolver este problema surgiu a idéia de se fazer a engenharia reversa do sistema, modelando o sistema e utilizando alguns diagramas da *Unified Modeling Language* (UML).

A engenharia reversa fornece informações da especificação e do projeto de um sistema de software (PFEEGER, 2004). Ela busca recuperar informações de engenharia com base nos métodos de especificação e de projeto de software. Depois essas informações são armazenadas de forma que sejam possíveis manipulá-las (PFEEGER, 2004).

Para tornar o processo mais eficaz foram utilizados alguns padrões de engenharia reversa. Estes padrões são divididos em três passos identificar, organizar e recuperar. O objetivo de utilizar estes padrões é melhorar a qualidade da engenharia reversa e aumentar a agilidade do processo (PERES, 2003).

Para realizar a engenharia reversa foram desenvolvidos o Modelo de Objetivos, Diagrama de Processos, Diagrama de Atividades, Requisitos Funcionais, Diagrama de Casos de Uso, Diagrama Entidade Relacionamento e Diagrama de Pacotes.

O software usado para construção dos sistemas Benner não disponibiliza o código e por isso não existe como fazer diretamente a engenharia reversa. Por este motivo foi desenvolvida uma rotina em Delphi que irá gerar os arquivos para importação no Enterprise Architect (EA). Estes arquivos têm a função de automatizar a geração do Diagrama Entidade Relacionamento.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é realizar a engenharia reversa parcialmente automatizada do sistema de Gestão de Processos da empresa Benner Sistemas.

Os objetivos específicos do trabalho são:

- a) criar um modelo de documentação na ferramenta CASE EA utilizando o Diagrama de Pacotes para organizar a documentação do sistema. Este Diagrama será criado manualmente;
- b) desenvolver uma rotina em Delphi a partir das informações armazenadas na ferramenta Builder para possibilitar a construção de um Diagrama Entidade

Relacionamento;

- c) documentar o módulo jurídico e financeiro do sistema Gestão de Processos utilizando a ferramenta EA e criar o Modelo de Objetivos, Diagrama de Processos, Requisitos funcionais, Diagrama de Atividades, Diagrama de Casos de Uso e Diagrama de Pacotes.

## 1.2 ESTRUTURA DO TRABALHO

O primeiro capítulo apresenta a introdução do trabalho. O segundo capítulo apresenta a fundamentação teórica além de informações do sistema Gestão de Processos. O terceiro capítulo aborda o desenvolvimento da rotina de geração dos diagramas, além do desenvolvimento dos diagramas propostos. Finalizando o trabalho, o quarto capítulo descreve as considerações finais.

## 2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho apresenta cinco tópicos. O primeiro comenta sobre a empresa, e as ferramentas utilizadas para o desenvolvimento do sistema Gestão de Processos. O segundo comenta a engenharia reversa, a terceira comenta a modelagem e UML. O quarto tópico apresenta o sistema e suas principais características e o quinto apresenta os trabalhos correlatos.

### 2.1 EMPRESA BENNER SISTEMAS

A empresa Benner Sistemas possui oito anos de experiência no mercado de soluções corporativas e foi criada em Blumenau. A empresa começou o desenvolvimento do sistema jurídico em meados de 2004. Os sistemas são desenvolvidos utilizando a ferramenta Builder desenvolvida pela própria empresa e emuladas através do Runner também desenvolvido pela empresa.

O Benner Builder é uma ferramenta de produtividade que permite a criação e manutenção de sistemas de maneira extremamente simples e rápida. Todos os sistemas gerados são baseados na arquitetura Cliente/Servidor e são compatíveis com Bancos de Dados relacionais, que atendam a comandos *Structured Query Language* (SQL).

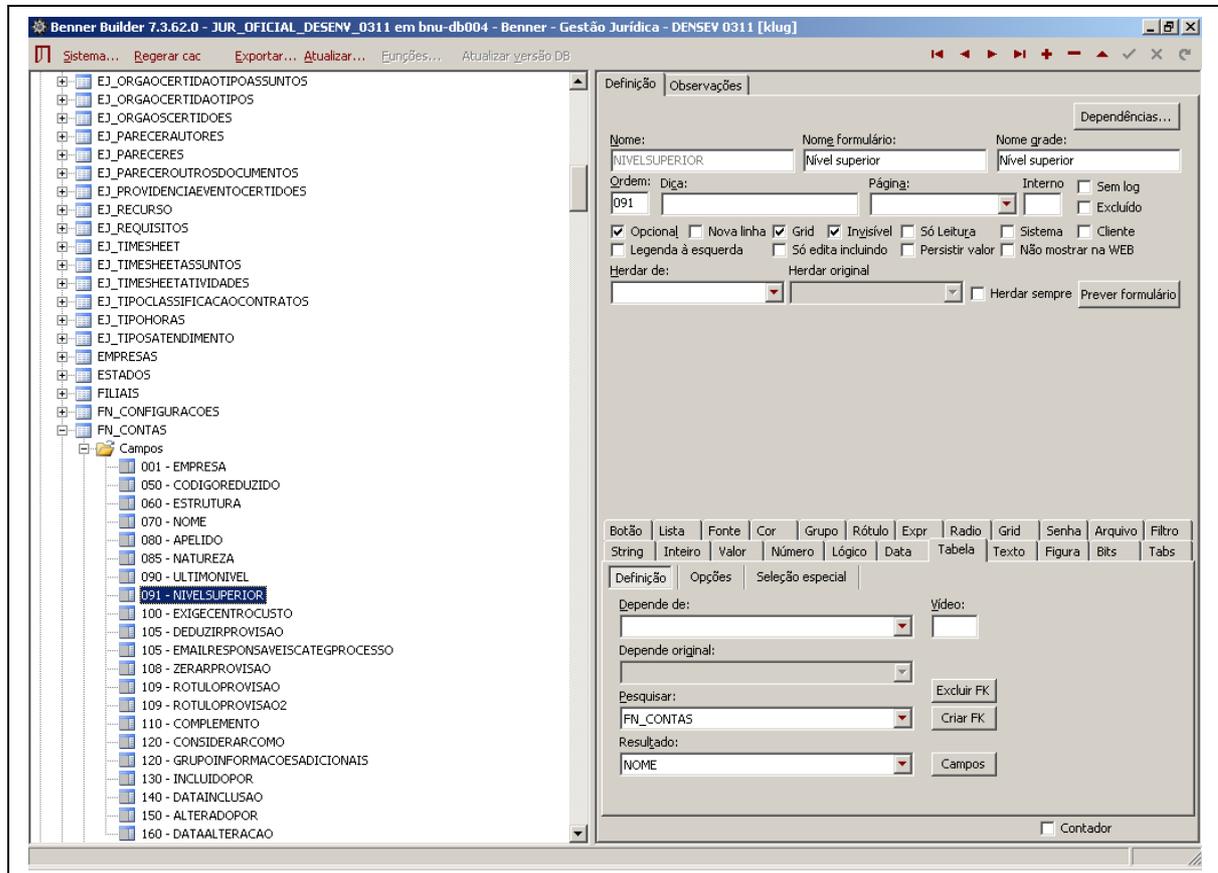
Através do Benner Builder, mesmo usuários com modesta cultura na área de programação, podem, mediante treinamento, realizar manutenções em sistemas produzidos anteriormente. Desta forma, simples personalizações, como a inclusão/exclusão de um campo, a alteração de legendas de campos, inclusão de novas versões de uma tabela, tornam-se muito fáceis de serem implementadas (SMITCH, 1997).

Os sistemas desenvolvidos pela ferramenta Builder basicamente são divididos em módulos, tabelas e campos, cada um desses itens são feitos através de cadastros. Por exemplo, para criar um novo módulo basta fazer um cadastro adicionando seu nome e legenda. Para criar uma nova tabela é a mesma regra, após a inserção das tabelas são criados os campos que podem ser textos, inteiros, chaves estrangeiras, datas e outros.

Na ferramenta Builder existem outras opções como índices, botões, visualização de formulários e outros. Todas estas informações são guardadas em tabelas e quando executado o

o sistema estas tabelas são lidas e geram a interface através do Runner. No sistema ainda pode-se codificar em Visual Basic através de macros ou ainda construir códigos mais complexos em Delphi.

A Figura 2 mostra a tela do sistema de desenvolvimento Builder.



Fonte: Smitch (1997).

Figura 2 – Ferramenta Builder

O sistema Gestão de Processos que é a base para este trabalho começou a ser desenvolvido no ano de 2004 e tem como foco a área jurídica de grandes empresas.

Este sistema no seu processo de desenvolvimento não teve uma análise adequada e foi construído sem modelagem e documentação. Com o crescimento do sistema isto se tornou um problema grave, pois os programadores e analistas que começaram o software não se encontram mais na empresa e as rotinas não foram documentadas. Hoje para realizar a manutenção em uma antiga rotina é muito complicado, pois ao arrumar alguma rotina pode se danificar algo que funciona corretamente.

A empresa está buscando a certificação do *Capability Maturity Model Integration* (CMMI) e umas das dificuldades encontradas para conseguir esta certificação é a falta de modelagem e documentação.

Hoje o sistema possui três camadas de desenvolvimento. A primeira delas é a versão

que é desenvolvida por uma área chamada engenharia. A segunda camada é a específica que é desenvolvida por uma equipe com o nome de entrega e ainda existe a camada cliente onde podem ser desenvolvidas regras de negócio pelo próprio cliente utilizando Visual Basic.

As novas implementações feitas no sistema são geralmente solicitadas pelos clientes onde um comitê decide se a rotina fará parte do sistema aplicativo ou será uma implementação específica. Estes levantamentos são feitos pelos consultores, através de simples escrita e após isto é enviado para o analista de sistema.

## 2.2 ENGENHARIA REVERSA

A engenharia reversa é um processo de análise do software com objetivo de recuperar seu projeto e sua especificação. O programa em si permanece inalterado no processo de engenharia reversa. O código fonte do software geralmente está disponível para entrada do processo de engenharia reversa. Algumas vezes contudo, se até o código fonte foi perdido a engenharia reversa precisa começar com o código executável.

A engenharia reversa não é o mesmo que a reengenharia. O objetivo da engenharia reversa é derivar o projeto ou especificação de um sistema a partir de seu código fonte. O objetivo da reengenharia é produzir um sistema novo, com manutenção mais fácil (SOMMERVILLE, 2003).

Atualmente, existe um grande número de empresas que continuam trabalhando com sistemas implementados em linguagens de programação antigas já em desuso, cujas manutenções são árduas e custosas. Esses sistemas, denominados legados, ainda são de muita utilidade aos seus usuários e muitas vezes, sua reconstrução, usando técnicas modernas de desenvolvimento de software, pode ser a solução para sua reutilização sem ter que construir um novo sistema.

Ao realizar a engenharia reversa podem existir perdas de informações. Para minimizar isto se recomenda a utilização de padrões de engenharia reversa. Estes padrões além de agilizar o processo de engenharia reversa facilitam a obtenção da documentação do projeto (PERES, 2003). Estes padrões são divididos em três partes, identificar, organizar e recuperar. Cada uma dessas atividades contém uma série de passos que auxiliam a engenharia reversa. Os padrões existentes em cada passo do processo são apresentados na Tabela 1, juntamente com suas descrições.

Quadro 1 – Padrões de engenharia reversa

<b>Nome Padrão</b>	<b>Descrição</b>
<b>Passo identificar</b>	
1. Identificar Objetivos	Identificar através de entrevistas os principais objetivos da aquisição do sistema por um cliente e armazená-los como fato na base de conhecimento.
2. Identificar processos de negócio	Identificar os processos de negócio vinculados ao sistema e armazená-los como fatos na base de conhecimento.
3. Identificar Casos de Uso	Identificar casos de uso e armazená-los como fatos na base de conhecimento.
<b>Passo organizar</b>	
1. Organizar a base de conhecimento	Organizar os fatos armazenados na base de conhecimento e dividi-los de acordo com as rotinas do sistema.
2. Criar Diagrama de Pacotes	Criar o Diagrama de Pacotes de acordo com a divisão de módulos, rotinas e diagramas.
<b>Passo recuperar</b>	
1. Criar o modelo de Objetivos	Criar o modelo de objetivos baseados nos fatos documentados na base de conhecimento.
2. Criar Diagrama de Processos	Criar o Diagrama de Processos com base nos processos documentados na base de conhecimento.
3. Criar Diagramas de Atividades	Criar o Diagrama de Atividades representando o fluxo de trabalho dos usuários do sistema.
4. Descrever Requisitos	Descrever os principais requisitos funcionais do sistema.
5. Criar Diagramas de Casos de Uso	Criar os casos de uso do sistema e descrever seus cenários.

Fonte: adaptado de Peres (2003).

## 2.3 MODELAGEM

Modelagem de software é a atividade de construir modelos que expliquem as características ou o comportamento de um software ou de um sistema de software. Na construção do software os modelos podem ser usados na identificação das características e funcionalidades que o software deverá prover (análise de requisitos), e no planejamento de sua construção (XAVIER, 2008). Os modelos revelam as características essenciais de um sistema, detalhes não relevantes e que só aumentariam a complexidade do problema podem ser ignorados.

Existem várias razões para utilizar modelos na construção de sistemas como comunicação entre as pessoas envolvidas, gerenciamento de complexidade, redução de custos no desenvolvimento, previsão de comportamento futuro do sistema e outros.

### 2.3.1 MODELO DE OBJETIVOS

O modelo de objetivos estabelece a razão pela qual a organização existe, o que ela está tentando alcançar e quais são as estratégias para atingir estes objetivos (KREMMER, 2007).

O modelo de objetivos é utilizado para descrever os objetivos que fizeram o cliente adquirir o sistema. Este diagrama possui um alto nível de abstração. Ele deve ser criado através de uma entrevista ao cliente.

Os objetivos são representados como objetos com o estereótipo <<goal>> (objetivo). Um objetivo pode ser dividido em subobjetivos. O atingimento de um subobjetivo contribui para o atingimento do objetivo superior. Esta conexão é representada através de uma dependência no sentido do objetivo para o subobjetivo. A figura 3 mostra um exemplo de uso do Modelo de Objetivos.

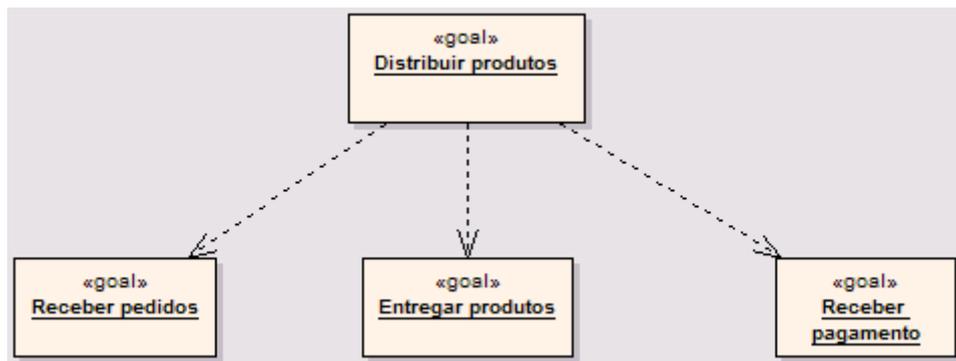


Figura 3 – Exemplo do modelo de objetivos

### 2.3.2 DIAGRAMA DE PROCESSOS

O Diagrama de Processos mostra uma seqüência de atividades que transforma entradas em saídas de valor para a organização. Ele possui um propósito e um objetivo, é afetado por eventos do ambiente externo ou de outros processos, utiliza recursos de entrada e produz recursos de saída (KREMMER, 2007).

A modelagem de processo ainda é usada principalmente na engenharia e reengenharia de processos (MONTEIRO, 2004).

Os processos de negócio são as partes ativas de uma organização. Eles são acionados para atingir os objetivos da organização que são descritos no modelo de objetivos. Normalmente um processo envolve várias unidades organizacionais.

Um método para descobrir os processos de negócio é identificar os eventos e seguir o fluxo de trabalho gerado por estes eventos. Por exemplo, um evento de viagem de um colaborador pode gerar várias atividades como reservas de passagens e hotéis, preenchimento de relatórios de viagens e reembolso de despesas.

Outra abordagem é identificar as atividades executadas e agrupá-las conforme os objetivos a que atendem.

Os processos são modelados como uma atividade com o estereótipo <<process>> (processo) e uma forma especial. Os recursos são modelados como objetos com o estereótipo <<resource>> (recurso). Os eventos são modelados com o elemento Event (evento).

A figura 4 mostra um exemplo de Diagrama de Processos no ciclo de desenvolvimento de um sistema.

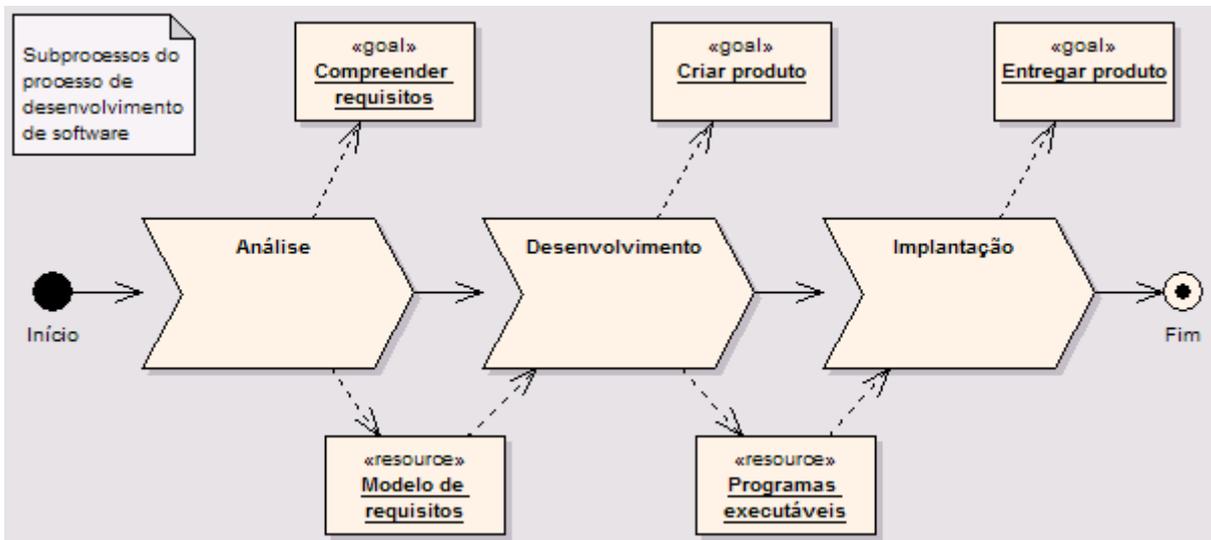


Figura 4 – Diagrama de processos exemplo

### 2.3.3 REQUISITOS FUNCIONAIS

A análise e especificação de requisitos é a forma que os projetistas e engenheiros de sistemas possuem para descrever o que o sistema deverá realizar e como deverá ser (MONTEIRO, 2004).

O principal objetivo do levantamento de requisitos é que usuários e desenvolvedores de sistema tenham a mesma visão do problema a ser resolvido (BEZERRA, 2007).

### 2.3.4 DIAGRAMA ENTIDADE RELACIONAMENTO

Diagrama entidade relacionamento é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração. Ele é a principal representação do Modelo de Entidades e Relacionamentos.

O diagrama mais importante para desenvolvimento de banco de dados é o Diagrama Entidade Relacionamento (MONTEIRO, 2004). O Diagrama Entidade Relacionamento é utilizado para representar as relações estáticas do sistema, ele indica quais as tabelas tem relacionamento e qual a cardinalidade entre elas.

A figura 5 mostra um exemplo de Diagrama Entidade Relacionamento.

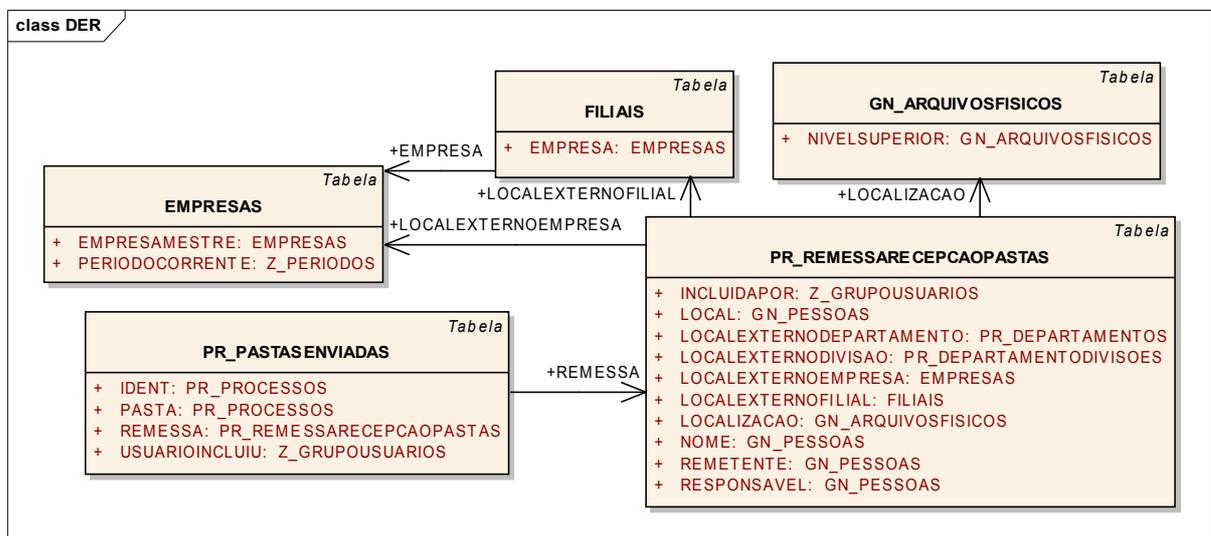


Figura 5 – Exemplo de Diagrama Entidade Relacionamento

### 2.3.5 UML

A UML é uma linguagem visual para documentação e padrões de softwares (PILONE, 2006). A UML permite que os desenvolvedores de sistemas especifiquem, visualizem e documentem os modelos de uma maneira que admita escalabilidade, a segurança e a execução robusta. Como a modelagem UML eleva o nível de abstração por todo o processo de análise e projeto, é mais fácil identificar padrões de comportamento (PENDER, 2004).

A UML é independente tanto de linguagens de programação quanto de processos de desenvolvimento. Isso quer dizer que a UML pode ser utilizada como modelagem de sistemas, não importa que linguagens de programação sejam utilizadas na implementação e

nem a forma de desenvolvimento adotada (BEZERRA, 2007).

A UML pode ser usada para desenvolvimento de quase qualquer tipo de sistemas como: sistemas comerciais, sistemas de tempo real, sistemas distribuídos, etc. Como é amplamente aceita na comunidade de desenvolvedores, fornecedores e fabricantes, é de fácil entendimento e de fácil comunicação.

A UML 2.0 possui vários diagramas, porém para este trabalho será usado o Diagrama de Casos de Uso, Atividades e Pacotes.

### 2.3.5.1 DIAGRAMA DE ATIVIDADES

O Diagrama de Atividades mostra os passos que são seguidos para executar um processo. Eles representam um fluxo de trabalho que integra o trabalho de várias pessoas ao longo de várias etapas para atingir o objetivo do processo. O Diagrama de Atividades é um nível intermediário de detalhamento. Ele é mais detalhado do que os processos e subprocessos, mas não ensina detalhadamente como executar as atividades. Isto poderia ser descrito em uma instrução de procedimento (KREMMER, 2007).

O Diagrama de Atividades pode ser visto como uma extensão dos fluxogramas (BEZERRA, 2007).

A figura 6 mostra um exemplo de Diagrama de Atividades.

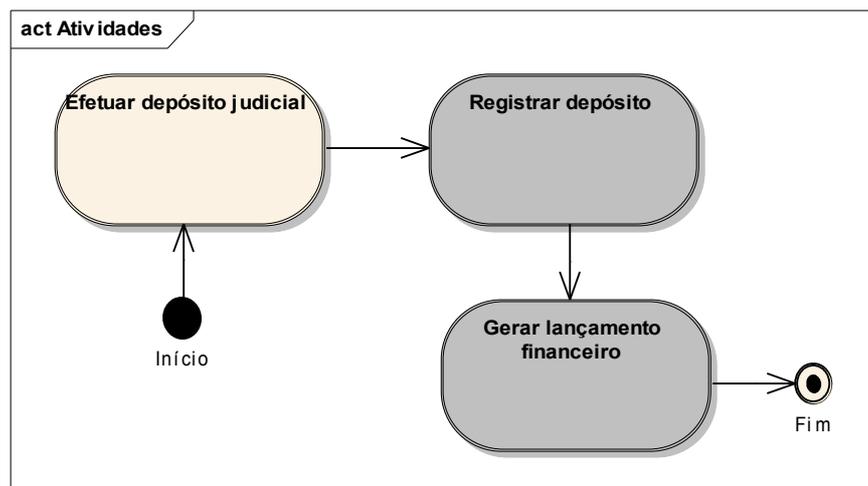


Figura 6 – Diagrama de Atividades exemplo

### 2.3.5.2DIAGRAMA DE CASOS DE USO

O Diagrama de Casos de Uso mostra como o sistema a ser desenvolvido vai interagir com seu ambiente (usuários ou outros sistemas). Ele é bastante importante porque vai ser a base do processo de desenvolvimento do sistema. O Diagrama de Classes especifica a estrutura do domínio e do sistema, os Casos de Usos formalizam as funcionalidades que o sistema deve cumprir (ANQUETIL, 2004).

A figura 7 mostra um exemplo de Diagrama de Casos de Uso.

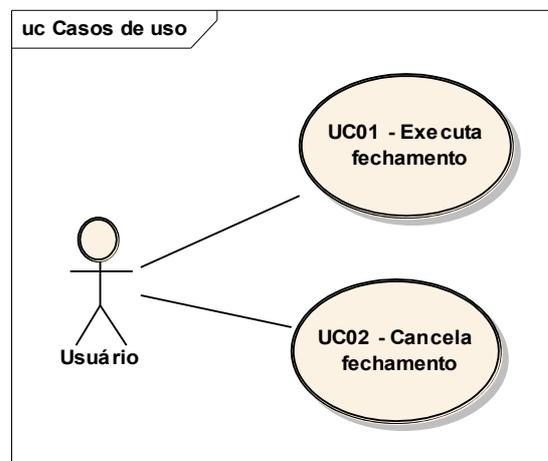


Figura 7 – Diagrama Casos de Uso exemplo

### 2.3.5.3DIAGRAMA DE PACOTES

O Diagrama de Pacotes será utilizado para melhor organizar a documentação do sistema. O Diagrama de Pacotes proporcionam uma ótima maneira para se visualizarem dependências entre partes do sistema (PILONE, 2006).

A figura 8 mostra um exemplo de um projeto de sistema organizado pelos pacotes da UML.

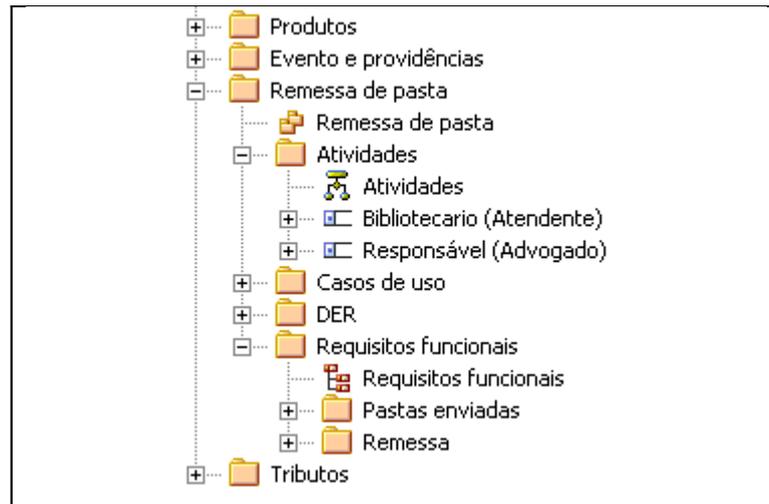


Figura 8 – Exemplo de Diagramas separados em pacotes

## 2.4 ESTRUTURA BÁSICA DO SISTEMA GESTÃO DE PROCESSOS

O sistema Gestão de Processos possui dois módulos: Jurídico e Financeiro. A seguir serão descritos os objetivos de cada módulo.

### 2.4.1 MÓDULO JURÍDICO

O módulo jurídico tem como objetivo o controle de processos, incluindo cadastros de desdobramentos, eventos, providências, pedidos, depósitos, penhoras e outros. A Figura 9 mostra a estrutura básica dos cadastros mais importantes do módulo (CIVIERO, 2007).

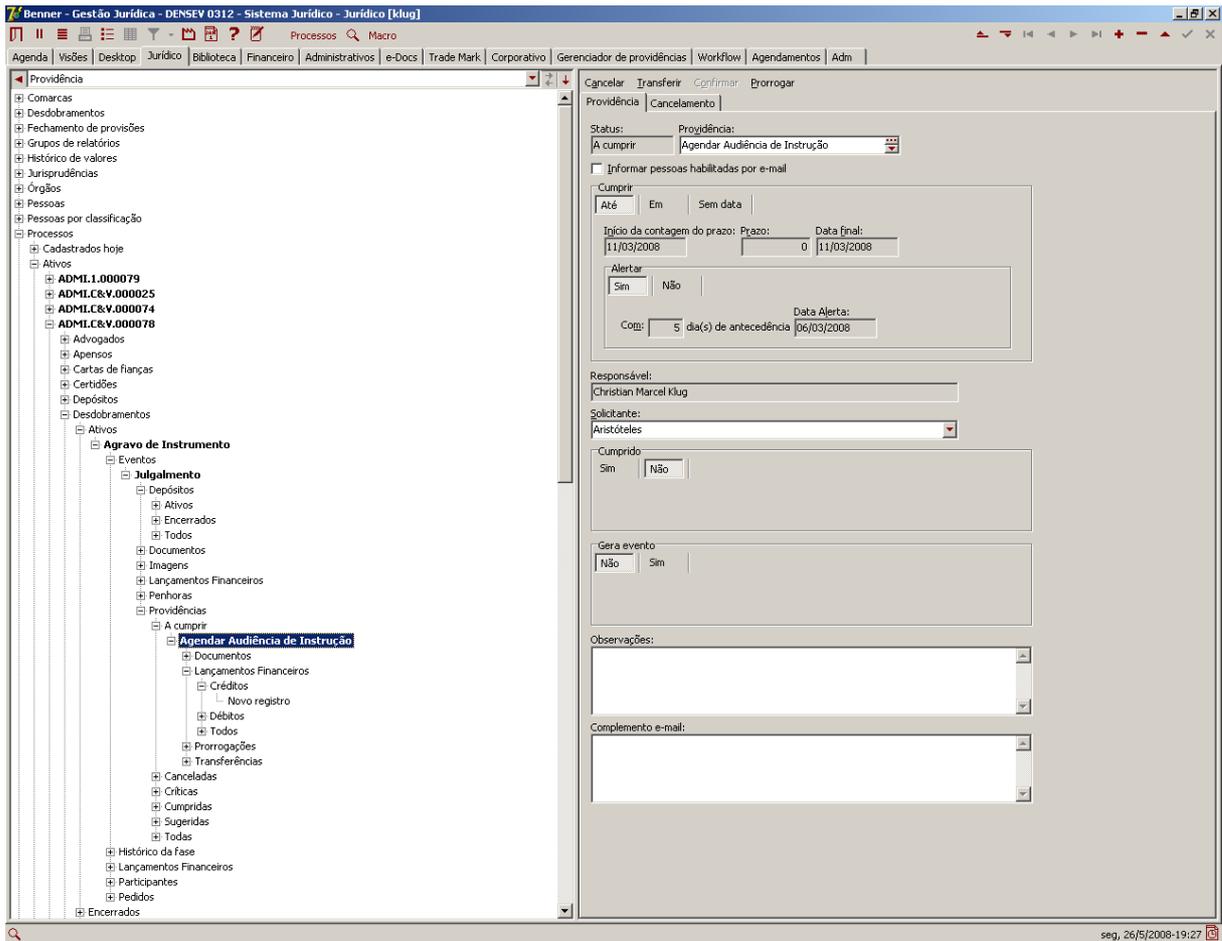


Figura 9 – Tela do módulo financeiro

Os cadastros mais importantes do módulo estão descritos a seguir, e foram adaptados de Civieiro 2007:

- **Pasta do processo:** O cadastro da pasta do processo é como se fosse uma capa do processo onde são cadastradas as informações genéricas, como a categoria, departamento e divisão responsável, se a empresa é passiva ou ativa, localização física do processo. A pasta também mostra um resumo das informações cadastradas abaixo como as partes principais do processo (autor, réu) o número do desdobramento principal, os somatórios dos valores de pedidos, tributos, sentenças, penhora, depósitos, provisões e outros.
- **Desdobramento:** O desdobramento é o processo que está no fórum. Todos os recursos feitos que provoquem redistribuição dos autos são novos desdobramentos. Os demais cadastros importantes estão ligados a estes dois cadastros principais. Abaixo é colocada uma breve explicação dos cadastros.
- **Cartas de fiança:** Usada no direito trabalhista como objeto de penhora pois equivale a

dinheiro.

- Depósitos: Destina-se ao cadastro dos depósitos judiciais e recursais cujo objeto será definido em campo específico.
- Documentos: tem por objetivo a associação de arquivos aos registros do sistema, ou seja, manter em banco de dados as peças e qualquer outro documento digital de diversos formatos (PDF, DOC, XLS, etc.) relacionados ao processo, possibilitando uma busca exata e centralização da alocação física dos arquivos, dentre outras vantagens.
- Eventos: Os eventos são os atos ou acontecimentos que ocorrem dentro ou fora do processo. São os também conhecidos, andamentos processuais.
- Obrigações: é a situação de fato que vincula duas ou mais pessoas que consiste no dever de dar, fazer, ou abster-se de fazer algo em proveito de outrem, que pode ser de ordem moral ou econômica.
- Pedidos: contém os pedidos pleiteados no processo independente de sua categoria (Cível, Trabalhista, etc.).
- Providências: são ações relacionadas ao processo delegadas aos colaboradores internos ou externos.
- Penhoras: é ato pelo qual o devedor demandado numa execução constringe bens em juízo a fim de garantir a dívida objeto do litígio. Caso esta não seja paga, o bem é vendido em leilão ou praça e o produto da venda é revertido em favor do credor.
- Produto: lista dos produtos da empresa envolvidos no processo.
- Participantes: contém todos os participantes contidos no processo como advogados, réu, autor, peritos, escritório terceiros e outros.
- Provisões: são provisões de valores feitas automaticamente com o objetivo de prever gastos com as ações.
- Tributos: são impostos, taxas e contribuições de melhoria.

Existe também o fechamento de provisão de valores. Este fechamento é feito sempre ao final do mês e seu objetivo é provisionar os valores que poderão ser gastos caso a empresa perca os processos. Após o fechamento é impresso um relatório que é enviado à contabilidade onde é usado como um passivo da empresa.

A remessa e recepção de pastas têm como finalidade controlar a remessa e recepção de pastas dos setores responsáveis, para que caso haja a movimentação física de documentos os mesmos possam ser localizados com eficiência.

## 2.4.2 MÓDULO FINANCEIRO

O módulo financeiro tem como objetivo acompanhar os gastos envolvidos nos processos além de controle de faturas e contratos com escritórios terceirizados (CIVIEIRO, 2007). A figura 10 mostra a estrutura básica de tabelas do módulo financeiro:

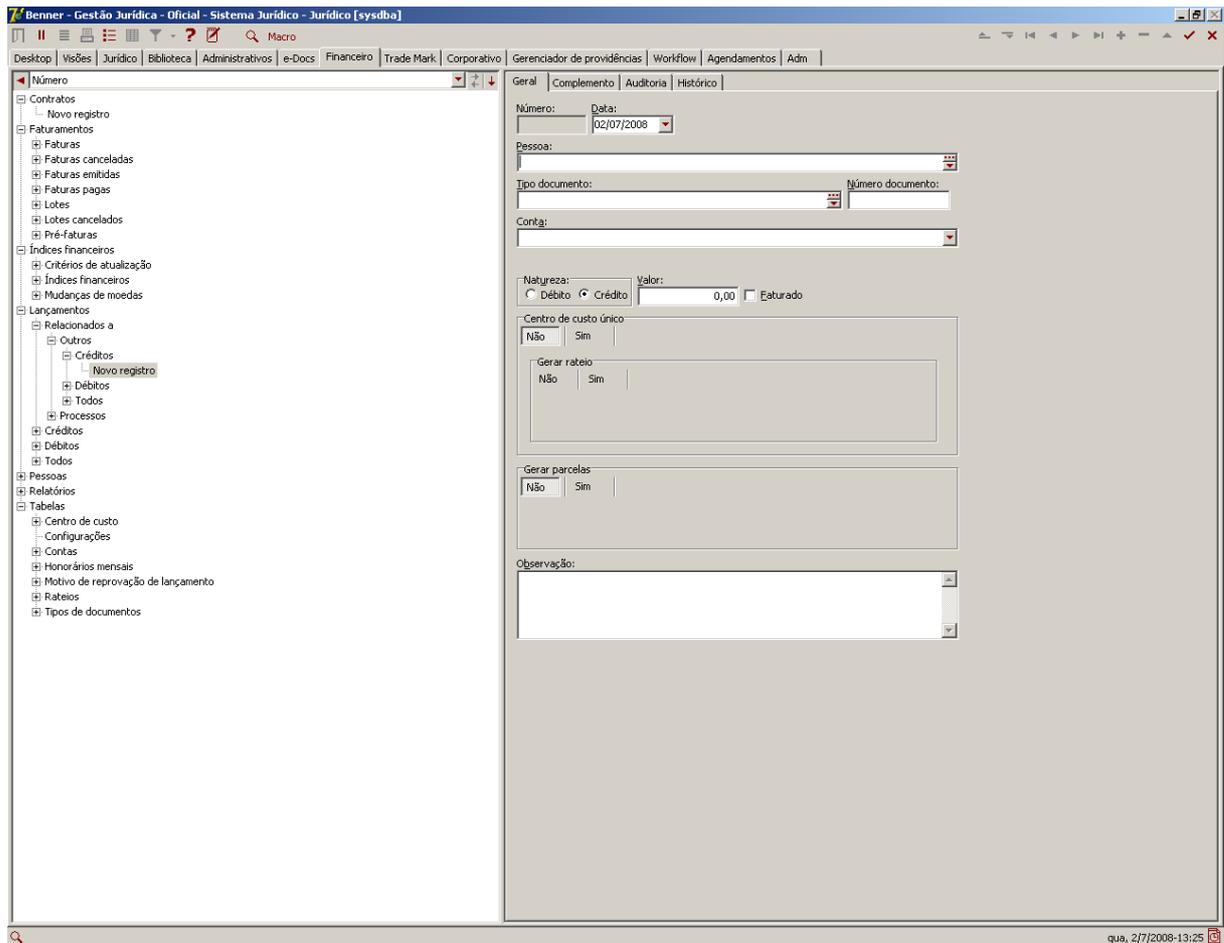


Figura 10 – Tela do módulo financeiro

Os principais cadastros estão descritos a seguir e foram adaptados de Civieiro 2007:

- Os lançamentos financeiros são gastos diversos que ocorrem dentro de um processo, cada um destes gastos deve ser informado à pessoa, conta centro de custo etc. Cada lançamento financeiro gera uma ou mais parcelas.
- Os faturamentos têm a função de agrupar os lançamentos financeiros. Eles devem ser agrupados por data e por pessoa ou centro de custo. Para gerar o faturamento é necessário gerar na seqüência um lote, uma pré fatura e a fatura.
- Contratos com escritório externos: São contratos feitos com escritórios externos que

tem como objetivo gerar lançamentos financeiros automáticos. Através de configurações é definida a forma de pagamento que pode ser por processo, mensal, pagamento por número de processos, êxito e outros.

## 2.5 TRABALHOS CORRELATOS

Existem muitos trabalhos, artigos e ferramentas sobre engenharia reversa. Como o trabalho sobre o Estudo Comparativo da Engenharia Reversa de Dados em Ferramentas CASE (SCARTON, 1997). O autor lista uma série de ferramentas CASE que possuem o recurso de engenharia reversa e faz um comparativo entre elas.

Outro trabalho é o Gerador de Bases de Conhecimento Genexus a Partir de Código Fonte da Linguagem FoxPro (HECKMANN, 1995), onde foi desenvolvida uma ferramenta que faz a engenharia reversa em sistemas desenvolvidos na linguagem FoxPro.

Existem ainda outros trabalhos sobre engenharia reversa como o Protótipo para Geração do Modelo Entidade Relacionamento (E-R) a Partir do Esquema Gerado para o Banco de Dados ORACLE7 (KOLLER, 1996).

### 3 DESENVOLVIMENTO

Este capítulo apresenta a especificação da rotina de geração de arquivos para importação do EA. A construção manual dos diagramas é abordada a partir do item 3.5.

#### 3.1 REQUISITOS DO SISTEMA

Os requisitos são divididos em funcionais e não funcionais. Requisitos Funcionais (RF) compreendem o levantamento das funcionalidades em geral do sistema, em outras palavras, produzir algo. Os Requisitos Não Funcionais (RNF) compreendem aspectos relacionados a atributos, propriedades, comportamentos e restrições.

##### 3.1.1 REQUISITOS FUNCIONAIS

O Quadro 1 apresenta os requisitos funcionais do sistema e sua rastreabilidade, ou seja, vinculação com os Casos de Uso associados.

<b>Requisitos Funcionais</b>	<b>Caso de Uso</b>
RF01: O sistema deverá permitir ao analista inserir, alterar, excluir rotinas do sistema.	UC01
RF02: O sistema deverá permitir ao analista inserir, alterar e excluir várias tabelas em uma rotina.	UC02
RF03: O sistema deverá possuir no cadastro de rotinas uma opção para pesquisar as tabelas que fazem ligação com a tabela principal. Cada tabela encontrada deverá ser inserida automaticamente no cadastro de tabelas.	UC03
RF04: O sistema deverá permitir a geração de arquivos para importação na ferramenta EA.	UC04

Quadro 2 – Requisitos funcionais

### 3.1.2 REQUISITOS NÃO FUNCIONAIS

O Quadro 2 lista os requisitos não funcionais previstos para o sistema.

<b>Requisitos Não Funcionais</b>
RNF01: A rotina de geração de código deverá ser desenvolvida usando Delphi 7.
RNF02: Os arquivos gerados deverão ser gerados com a mesma extensão dos arquivos em Delphi (pás).

Quadro 3 – Requisitos não funcionais

### 3.2 ESPECIFICAÇÃO

Neste item são apresentadas as atividades desempenhadas na fase de especificação do sistema desenvolvido, como o diagrama de Casos de Uso e o dicionário de dados.

#### 3.2.1 CASOS DE USO

A interação entre sistema e o usuário é denominado Caso de Uso, que proporcionam a descrição dos requisitos funcionais do sistema de forma clara e concisa. A figura 11 apresenta os Casos de Uso do sistema. A figura 11 mostra os Casos de Uso da rotina de geração de arquivos.

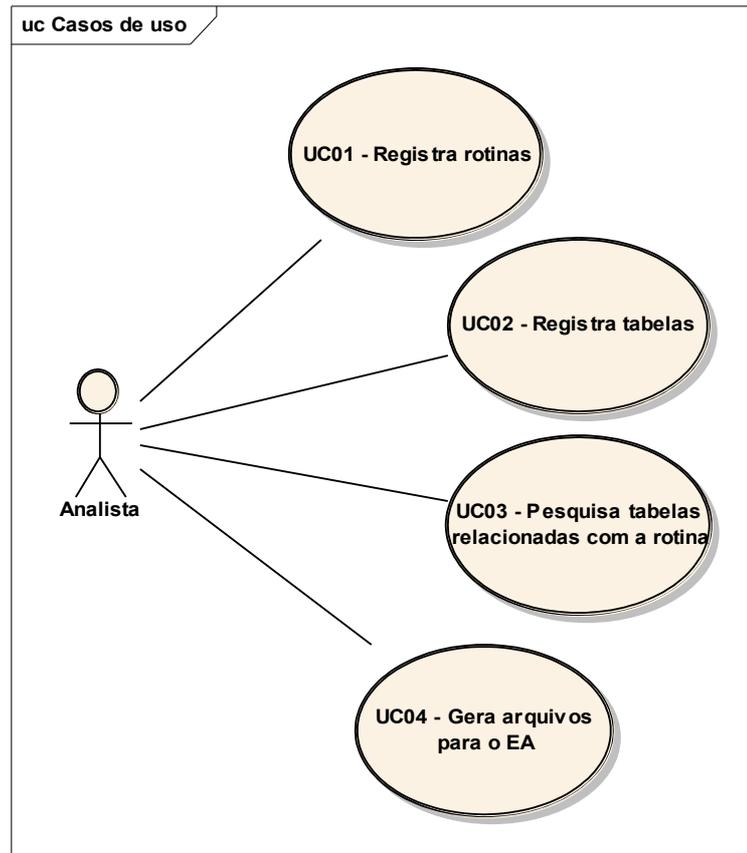


Figura 11 – Diagramas de Casos de Uso do sistema

### 3.2.1.1 CENÁRIOS DOS CASOS DE USO

A seguir nos quadros 3 a 6 são apresentados os cenários.

#### UC01 – Registra rotinas

##### Principal

1. O sistema apresenta a tela para cadastro de rotinas. 2. O usuário preenche o nome da rotina. 3. O usuário preenche a tabela principal da rotina. 4. O usuário preenche o campo incluir tabelas do sistema. 5. O usuário preenche quantos níveis o sistema deverá sugerir tabelas. 6. O usuário preenche o campo observações. 7. O usuário valida e salva o registro.

##### Campos não preenchidos. - Alternativo

No passo 7 caso algum campo obrigatório não foi preenchido o sistema emite uma mensagem indicando o campo que não foi preenchido.

Quadro 4 – Caso de Uso registra rotina

#### UC02 – Registra tabelas

**Principal**

1. O sistema exibe a tela para cadastro de tabelas.2. O usuário preenche o campo tabela.3. O usuário salva o registro.

**Campos obrigatórios - Alternativo**

Caso no passo 2 o usuário não informe a tabela o sistema não sistema mostrara uma mensagem de que falta preencher o campo.

**Campo nível - Alternativo**

Caso a tabela foi inserida através da rotina de sugestões de tabelas o sistema preenche o campo nível de forma automática. Caso a inserção foi manual o campo fica em branco.

Quadro 5 – Caso de Uso registra tabelas

**UC03 – Pesquisa rotina relacionadas a tabelas****Principal**

1. O sistema apresenta a tela de cadastro de rotinas.2. O usuário pressiona o campo gerar ligações.3. O sistema insere as tabelas encontradas abaixo da rotina.4. O sistema apresenta uma mensagem de execução com sucesso.

**Erro de execução - Alternativo**

Caso ocorra algum erro durante a execução o sistema alerta através de uma mensagem.

Quadro 6 – Caso de Uso pesquisa tabelas

**UC04 - Cadastro de rotinas****Principal**

1. O sistema apresenta a tela de cadastro de rotinas.2. O usuário pressiona o campo gerar arquivo.3. O sistema gera um arquivo com a extensão pas na pasta c://TCC.4. O sistema apresenta uma mensagem de execução com sucesso.

**Erro ao gerar o arquivo - Alternativo**

Caso ocorra algum erro ao gerar o arquivo o sistema emite uma mensagem para o usuário.

Quadro 7 – Caso de Uso cadastro de rotinas

### 3.2.2 DICIONÁRIO DE DADOS

Nesta seção é apresentado o dicionário de dados. Os dados apresentados são a descrição do campo (Descrição), o nome físico do atributo (Cód. Atributo), o tipo do atributo (Tipo) e se é chave primária (PK) ou estrangeira (FK).

ROTINAS				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Código	Handle	int	Sim	Não
Nome da rotina	Rotina	varchar(50)	Sim	Não
Tabela principal	Tabela	int	Não	Sim
Incluir tabelas de sistema	incluirtabelasstema	char(1)	Não	Não
Nível	Nível	Int	Não	Não
Observações	Observacoes	varchar(50)	Não	Não

Quadro 8 – Entidade rotinas

ROTINATABELAS				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Código	Handle	int	Sim	Não
Nome da rotina	Rotina	Int	Sim	Sim
Não incluir	Naoincluir	Char(1)	Não	Não
Incluir tabelas de sistema	incluirtabelasstema	char(1)	Não	Não
Nível	Nível	Int	Não	Não

Quadro 9 – Entidade rotinas e tabelas

TABELAS				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Código	Handle	int	Sim	Não
Tabela	Nome	Int	Sim	Não

Quadro 10 – Entidade tabelas

### 3.3 IMPLEMENTAÇÃO

A rotina foi desenvolvida em linguagem Delphi. No sistema Gestão de Processos foi criada uma chamada escrita em VBA para acessar a DLL escrita em Delphi.

O quadro 10 mostra o trecho de código onde o sistema faz a verificação dos relacionamentos entre as tabela principal e as demais.

No início do código é verificado a tabela principal da rotina e após são verificado as tabelas que tem relacionamento com ela. Cada tabela encontrada é inserida no cadastro de tabelas. Após encontrar as tabelas que fazem relacionamento com a tabela principal a rotina cria as ligações para as tabelas inseridas até o nível escolhido no cadastro de rotinas.

```
procedure TGerarLigacoes.GerarTodasLigacoes(pRotina: Integer);
```

```

var
  qTabelas : TBQuery;
  qLigacoes : TBQuery;
  qUpdate : TBQuery;
  qRegras : TBQuery;
  qRotina : TBQuery;
  concluido : Boolean;
  sTabela : String;
  iTabelaCount : Integer;

begin
  try
    try
      qTabelas := TBQuery.Create(nil);
      qTabelas.DatabaseName := FSistema.dataBaseName;
      qLigacoes := TBQuery.Create(nil);
      qLigacoes.DatabaseName := FSistema.dataBaseName;
      qUpdate := TBQuery.Create(nil);
      qUpdate.DatabaseName := FSistema.dataBaseName;
      qRegras := TBQuery.Create(nil);
      qRegras.DatabaseName := FSistema.DataBaseName;
      qRotina := TBQuery.Create(nil);
      qRotina.DatabaseName := FSistema.DataBaseName;
      concluido := False;

      qRotina.Active := False;
      qRotina.SQL.Text := ' SELECT NIVEL, INCLUIRTABELASISTEMA ' +
        ' FROM K9_GN_ROTINAS WHERE HANDLE = :PROTINA ';
      qRotina.ParamByName('PROTINA').AsInteger := pRotina;
      qRotina.Active := true;

      qTabelas.Active := False;
      qTabelas.SQL.Text := ' SELECT TABELA TABELA, NIVEL ' +
        ' FROM K9_GN_ROTINATABELAS ' +
        ' WHERE ROTINA = :PROTINA AND ' +
        ' NIVEL <= :PNIVEL AND ' +
        ' VERIFICADA = :PVERIFICADA AND ' +
        ' TABELA IS NOT NULL ';
      qTabelas.ParamByName('PVERIFICADA').AsString := 'N';
      qTabelas.ParamByName('PROTINA').AsInteger := pRotina;
      qTabelas.paramByName('PNIVEL').AsInteger := qRotina.FieldByName('NIVEL').AsInteger;

      while not concluido do
        begin
          qTabelas.Active := False;

          qTabelas.Active := True;

          if not qTabelas.Eof then
            begin
              qLigacoes.Active := False;
              qLigacoes.SQL.Text := ' SELECT A.TABELA TABELA, ' +
                ' A.PESQUISAR PESQUISAR ' +
                ' FROM Z_CAMPOS A ' +
                ' WHERE A.CLASSE = 6 ' +
                ' AND ((A.TABELA = :PTABELA) OR (A.PESQUISAR = :PTABELA));';
              qLigacoes.ParamByName('PTABELA').AsInteger := qTabelas.FieldByName('TABELA').asInteger;
            end
          end
        end
      end
    end
  end
end

```

```

if qRotina.fieldByName('INCLUIRTABELASISTEMA').AsString = 'N' then
  qLigacoes.SQL.Add(' AND ((A.TABELA IN (SELECT HANDLE FROM Z_TABELAS WHERE NOME NOT ' +
    ' LIKE "Z_%")' +
    ' AND (A.PESQUISAR IN (SELECT HANDLE FROM Z_TABELAS WHERE NOME NOT ' +
    ' LIKE "Z_%"))));

qLigacoes.Active := true;

while not qLigacoes.Eof do
begin

  if qLigacoes.FieldByName('TABELA').asInteger = qTabelas.FieldByName('TABELA').asInteger then
  begin
    if deveIncluirTabela(pRotina, qLigacoes.FieldByName('PESQUISAR').asInteger) then
      InserirTabela(pRotina, qLigacoes.FieldByName('PESQUISAR').asInteger,
        qTabelas.FieldByName('NIVEL').AsInteger + 1);
    end
  else
  begin
    if deveIncluirTabela(pRotina, qLigacoes.FieldByName('TABELA').asInteger) then
      InserirTabela(pRotina, qLigacoes.FieldByName('TABELA').asInteger,
        qTabelas.FieldByName('NIVEL').AsInteger + 1);
    end;
  end;

  qLigacoes.Next;
end;

qUpdate.Active := False;
qUpdate.SQL.Text := ' UPDATE K9_GN_ROTINATABELAS SET VERIFICADA = :PVERIFICADA WHERE ' +
TABELA = :PTABELA AND ROTINA = :PROTINA';
qUpdate.ParamByName('PTABELA').AsInteger := qTabelas.FieldByName('TABELA').AsInteger;
qUpdate.ParamByName('PROTINA').AsInteger := pRotina;
qUpdate.ParamByName('PVERIFICADA').AsString := 'S';
qUpdate.ExecSQL;

end
else
  concluido := True;
end;

except
  raise Exception.Create(sProblema + Exception(ExceptObject).Message);
end;
finally
  FreeAndNil(qTabelas);
  FreeAndNil(qLigacoes);
  FreeAndNil(qUpdate);
  FreeAndNil(qRegras);
  FreeAndNil(qRotina);
end;

end;

```

Quadro 11 – Trecho de código

### 3.4 OPERACIONALIDADE DO SISTEMA

Esta seção apresenta o funcionamento da rotina.

#### 3.4.1 CADASTRO DE ROTINAS

No cadastro de rotinas é informado o nome da rotina, a tabela principal, se a rotina deve incluir tabela do sistema e os níveis de ligações que o analista deseja gerar.

A tabela principal indica a primeira tabela verificada pela rotina, a partir dela a rotina encontra as demais tabelas.

Se a opção incluir tabelas do sistema não estiver marcada o Diagrama Entidade Relacionamento será gerado sem as tabelas de sistema como, por exemplo, a tabela de usuários. O campo nível indica quantos níveis de ligações serão gerados em relação à tabela principal. Por exemplo se for preenchido o campo com valor 1 o sistema irá gerar apenas as tabelas que são FK com a tabela principal.

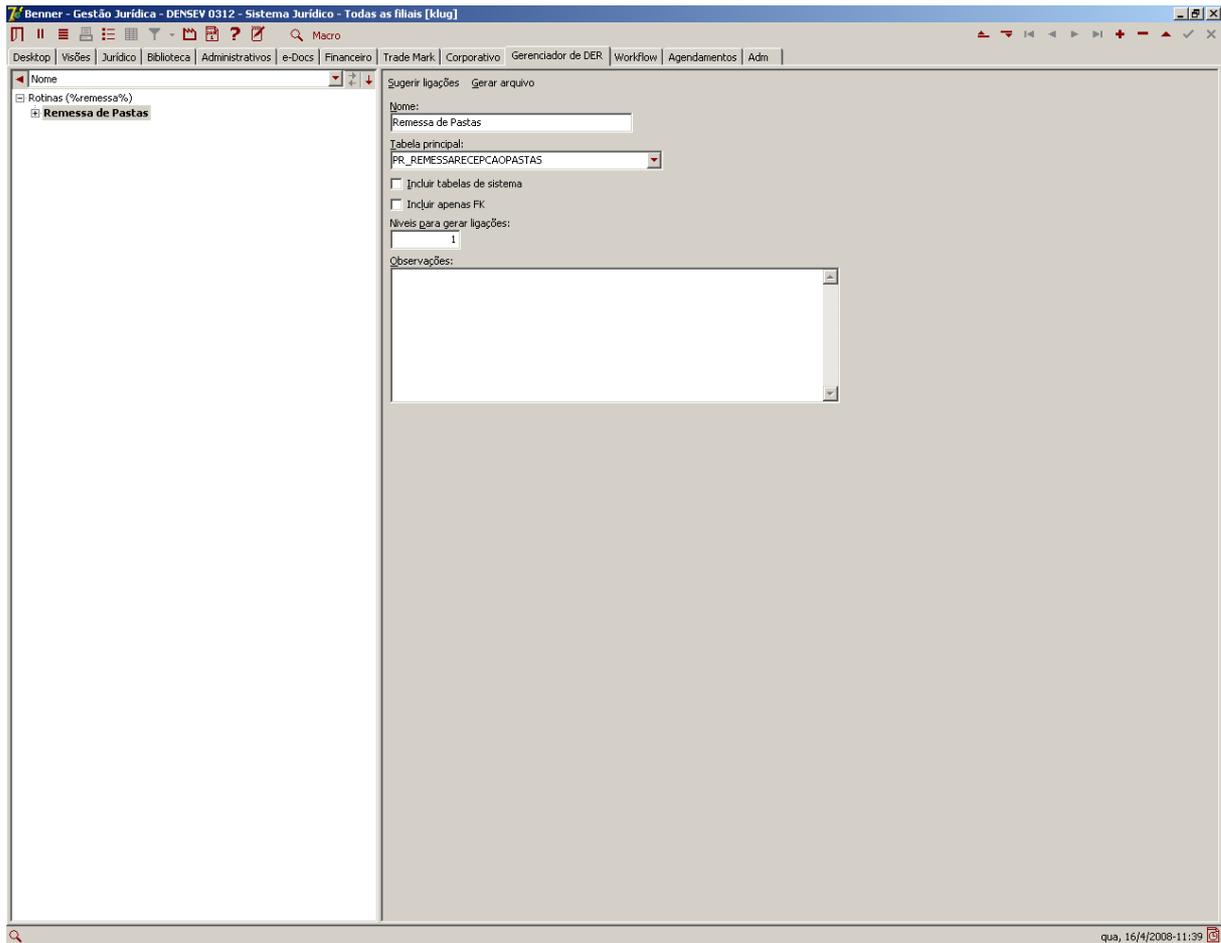


Figura 12 – Cadastro de rotinas

### 3.4.2 CADASTRO DE TABELAS

O cadastro de tabelas serve para definir todas as tabelas que irão ser geradas no arquivo. Existem duas formas de inserir nesta tabela, manualmente e automaticamente. A manual basta adicionar através do botão “+”. A forma automática a tabela incluída através do botão sugerir tabelas que está no cadastro de rotinas (figura 12).

A figura 13 mostra o cadastro de tabelas.

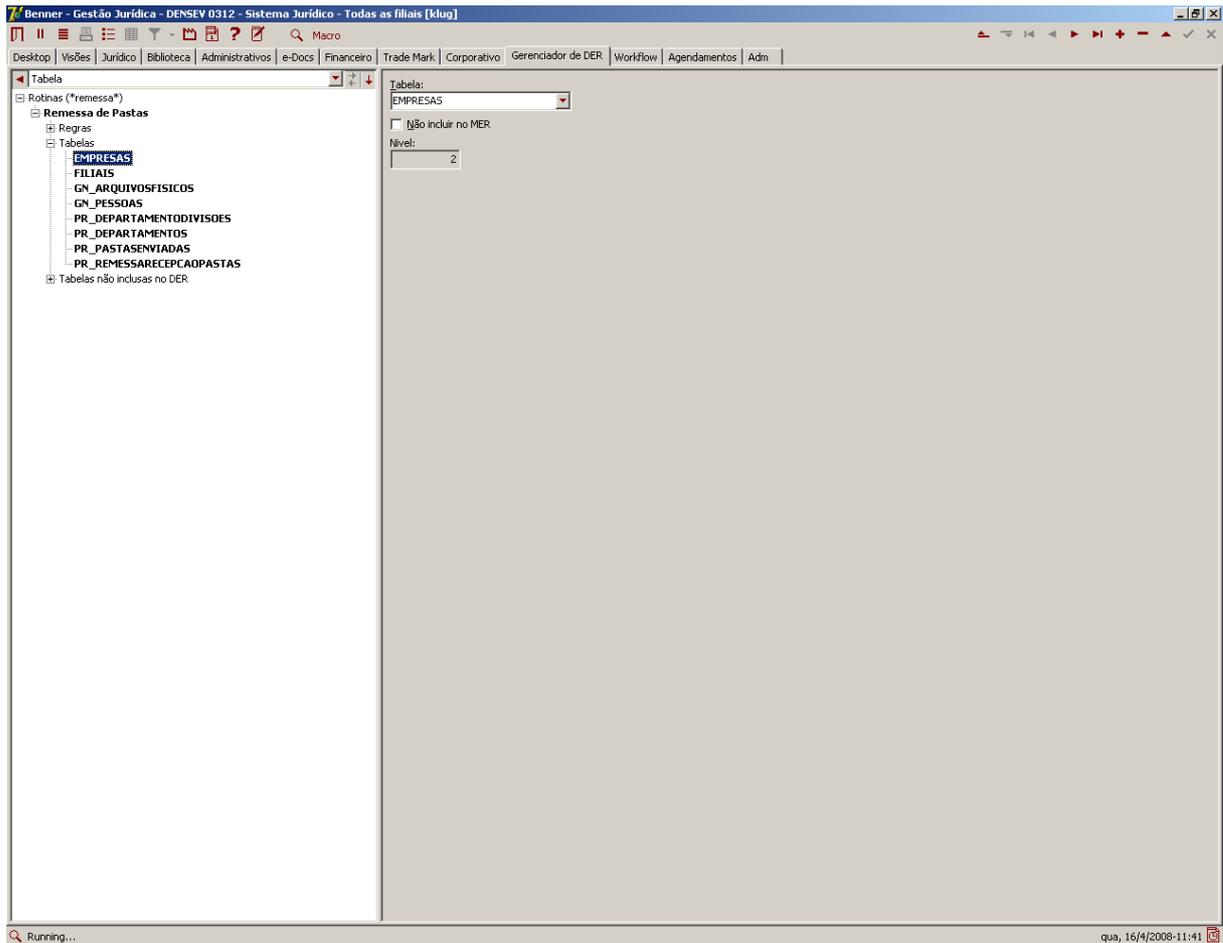


Figura 13 – Cadastro de tabelas

### 3.4.3 EXEMPLO DE FUNCIONAMENTO DA ROTINA

Esta seção apresenta um exemplo que mostra passo a passo o funcionamento da rotina.

No cadastro de rotinas é cadastrada uma nova rotina com o nome de Participantes, o nome da tabela principal é PR\_PROCESSOPARTICIANTEs, a opção incluir tabelas do sistema não está setada e o número de níveis escolhido é um.

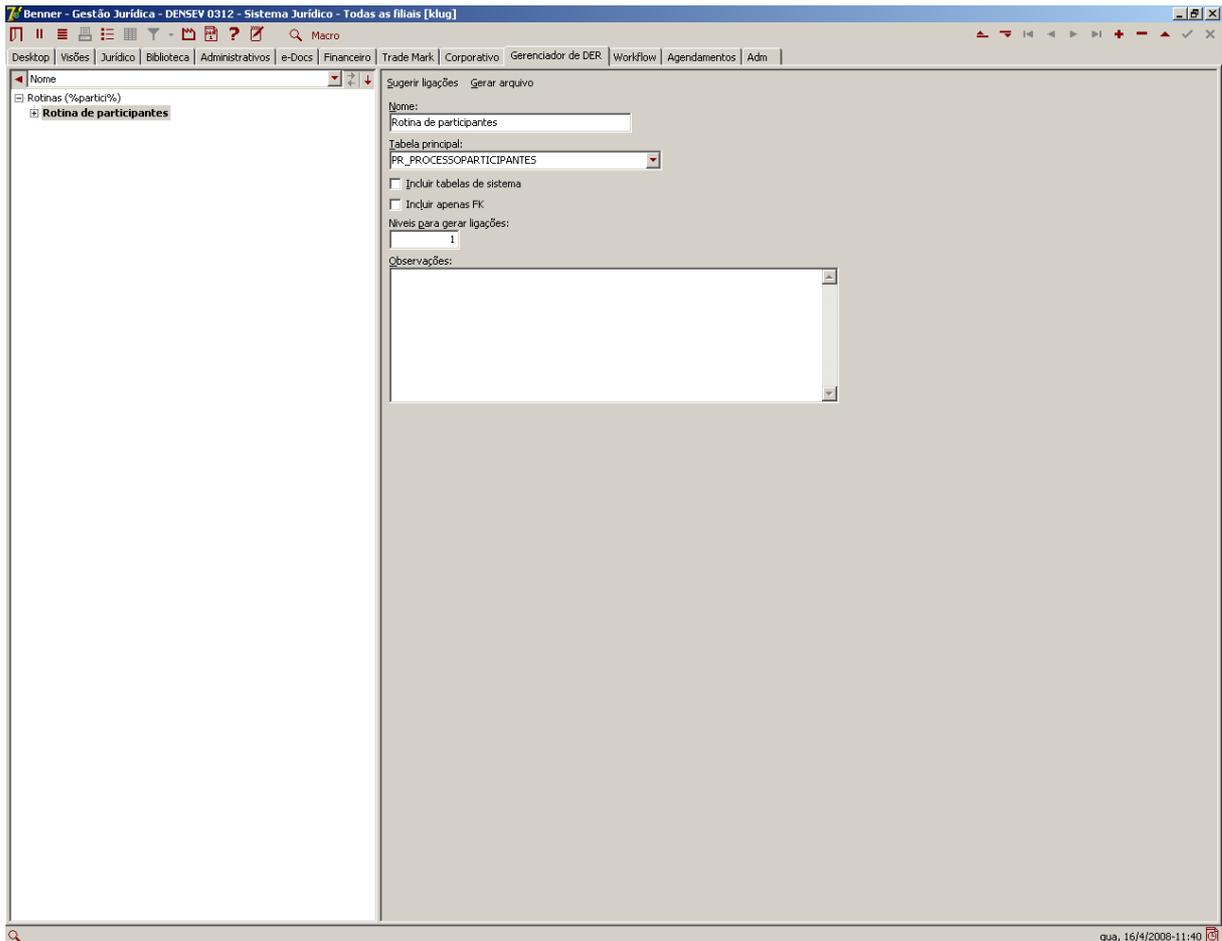


Figura 14 – Cadastro de rotinas exemplo

Ao clicar no botão sugerir ligações, o sistema insere na tabela K9\_GN\_ROTINATABELAS as tabelas que serão geradas no arquivo. A inserção é feita da seguinte forma: No início a rotina insere a tabela principal PR\_PROCESSOPARTICIPANTES, o nível 1 e o campo verificado igual a N. Após esta operação a tabela fica da seguinte forma:

Tabela	Nível	Verificado
PR_PROCESSOPARTICIPANTES	1	N

Quadro 12 – Estado inicial da tabela

Ao clicar no botão sugerir ligações, o segundo passo da rotina é encontrar todas as tabelas que fazem ligação com a tabela principal. A rotina verifica todas as tabelas que são ‘filhas’ ou ‘pais’ da tabela PR\_PROCESSOPARTICIPANTES.

Após esta verificação a tabela fica conforme o quadro 12.

Tabela	Nível	Verificada
PR_PROCESSOPARTICIPANTES	1	S
PR_CATEGORIAS	2	N
PR_CATEGORIATIPOACAOASSUNTOS	2	N
PR_CATEGORIATIPOACOES	2	N
PR_PARTEQUALIFICACAO	2	N
PR_PROCESSODEPOSITOLANCAMENTOS	2	N
PR_PROCESSOPEDIDOS	2	N

Quadro 13 – Estado da tabela no segundo nível

Conforme mostra o quadro 12 é possível notar que o campo verificada foi alterado para S isto indica que já foram encontradas todas as ligações desta tabela, o campo nível igual a 1 indica que é a tabela principal. Após a inserção das tabelas de segundo nível a rotina encerra, pois foi parametrizado apenas para gerar um nível a partir da tabela principal. Se existissem mais níveis a rotina verificaria cada tabela e após isto preencheria o campo verificado com S.

Existe uma consistência na rotina para não incluir tabelas duplicadas. A figura 15 mostra as tabelas inseridas pela rotina.

Após a inserção das tabelas é possível gerar o arquivo para importação no EA. Para isto é preciso pressionar o botão gerar arquivo na tela de cadastros de rotinas. Ao pressionar o botão o sistema cria um arquivo com o nome da rotina e a extensão .pas no endereço c:\TCC.

O quadro 13 mostra o arquivo gerado.

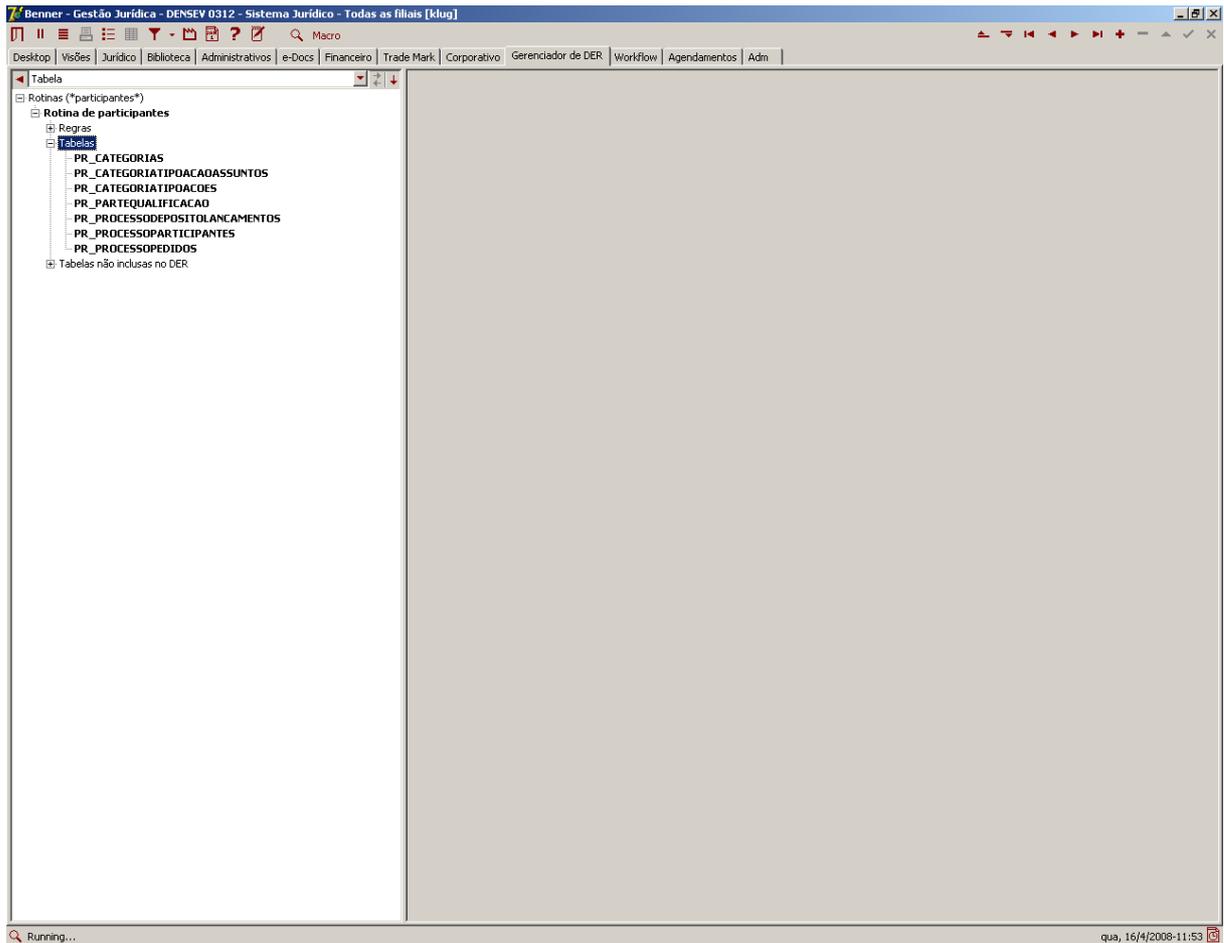


Figura 15 – Cadastro de tabelas exemplo

Unit Rotina\_de\_participantes;

interface

uses

Windows, Messages;

type

PR\_CATEGORIAS= Class(Tabela)

RAMOPROCESSUAL: PR\_RAMOSPROCESSUAIS;

MOEDA: GN\_MOEDAS;

CRITERIO: GN\_CRITERIOS;

end;

PR\_CATEGORIATIPOACAOASSUNTOS= Class(Tabela)

CATEGORIA: PR\_CATEGORIAS;

TIPOACAO: PR\_CATEGORIATIPOACOES;

end;

PR\_CATEGORIATIPOACOES= Class(Tabela)

CATEGORIA: PR\_CATEGORIAS;

end;

```

PR_PARTEQUALIFICACAO= Class(Tabela)

end;

PR_PROCESSODEPOSITOLANCAMENTOS= Class(Tabela)

PROCESSO: PR_PROCESSOS;
DEPOSITO: PR_PROCESSODEPOSITOS;
DEPOSITARIO: PR_DEPOSITARIOS;
BANCO: GN_BANCOS;
RESPONSAVELRESGATE: PR_PROCESSOPARTICIPANTES;
end;

PR_PROCESSOPARTICIPANTES= Class(Tabela)

PROCESSO: PR_PROCESSOS;
PARTICIPANTE: GN_PESSOAS;
DESDOBRAMENTO: PR_PROCESSODESDOBRAMENTOS;
CATEGORIA: PR_CATEGORIAS;
TIPO: PR_CATEGORIATIPOACOES;
ASSUNTO: PR_CATEGORIATIPOACAOASSUNTOS;
CONDICAO: PR_PARTEQUALIFICACAO;
PEDIDO: PR_PROCESSOPEDIDOS;
K9_INCLUIDOPOR: Z_GRUPOUSUARIOS;
end;

PR_PROCESSOPEDIDOS= Class(Tabela)

PROCESSO: PR_PROCESSOS;
PEDIDO: PR_PEDIDOS;
DESDOBRAMENTO: PR_PROCESSODESDOBRAMENTOS;
RISCO: PR_RISCOS;
MOEDA: GN_MOEDAS;
CRITERIO: GN_CRITERIOS;
PEDIDODEPENDEDE: PR_PROCESSOPEDIDOS;
end;

end.

```

Quadro 14 – Arquivo gerado

Após o arquivo gerado já é possível importar as tabelas no EA, para isto é necessário abrir o programa, adicionar um diagrama Entidade Relacionamento e selecionar as opções conforme mostra a seqüência das figuras 16 a 17.

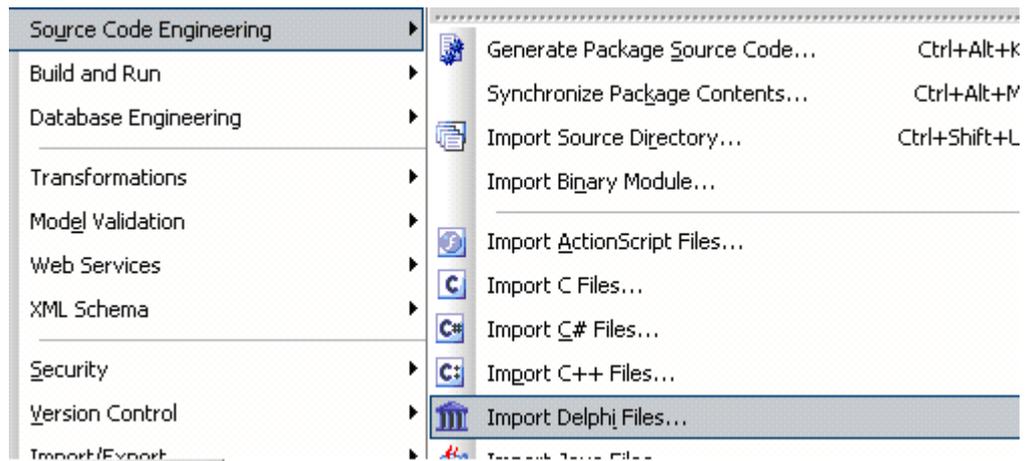


Figura 16 – Opção para importação de arquivo Delphi

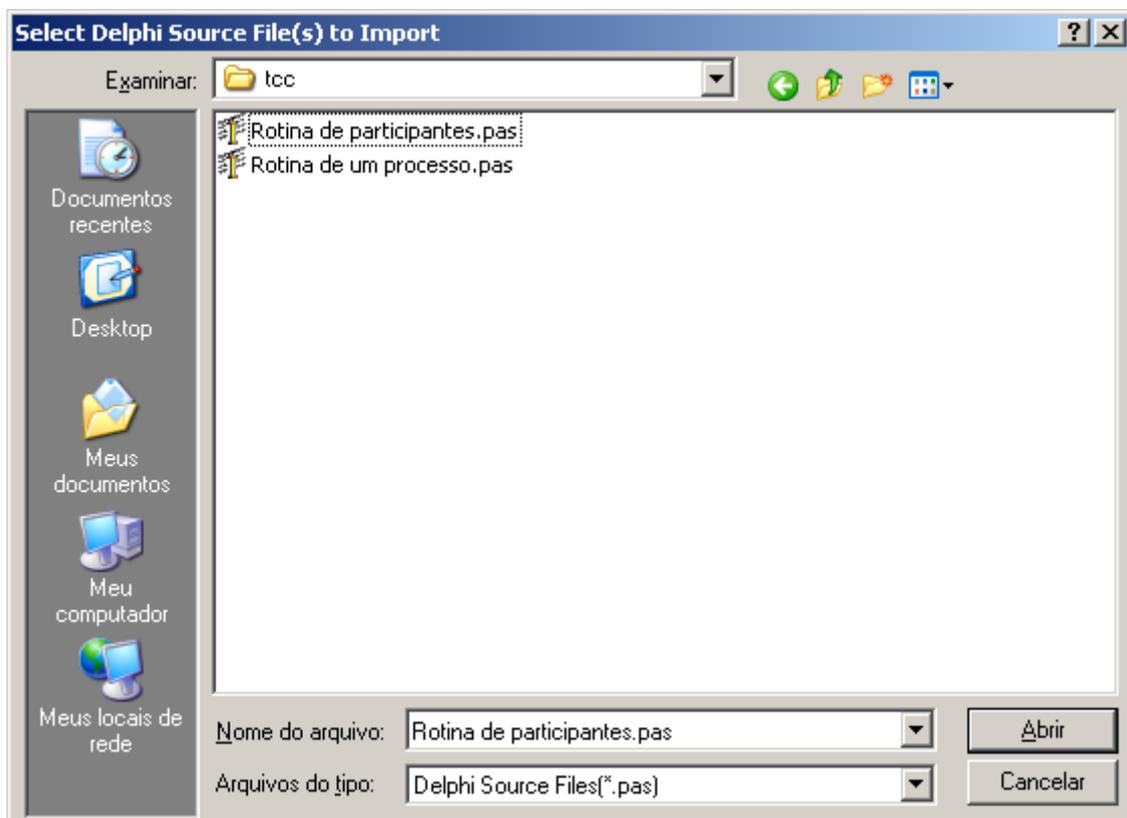


Figura 17 – Selecionando arquivo gerado

Ao pressionar o botão abrir o EA gera o diagrama automaticamente, após isto é necessário organizar o diagrama. A figura 18 mostra o diagrama já organizado.

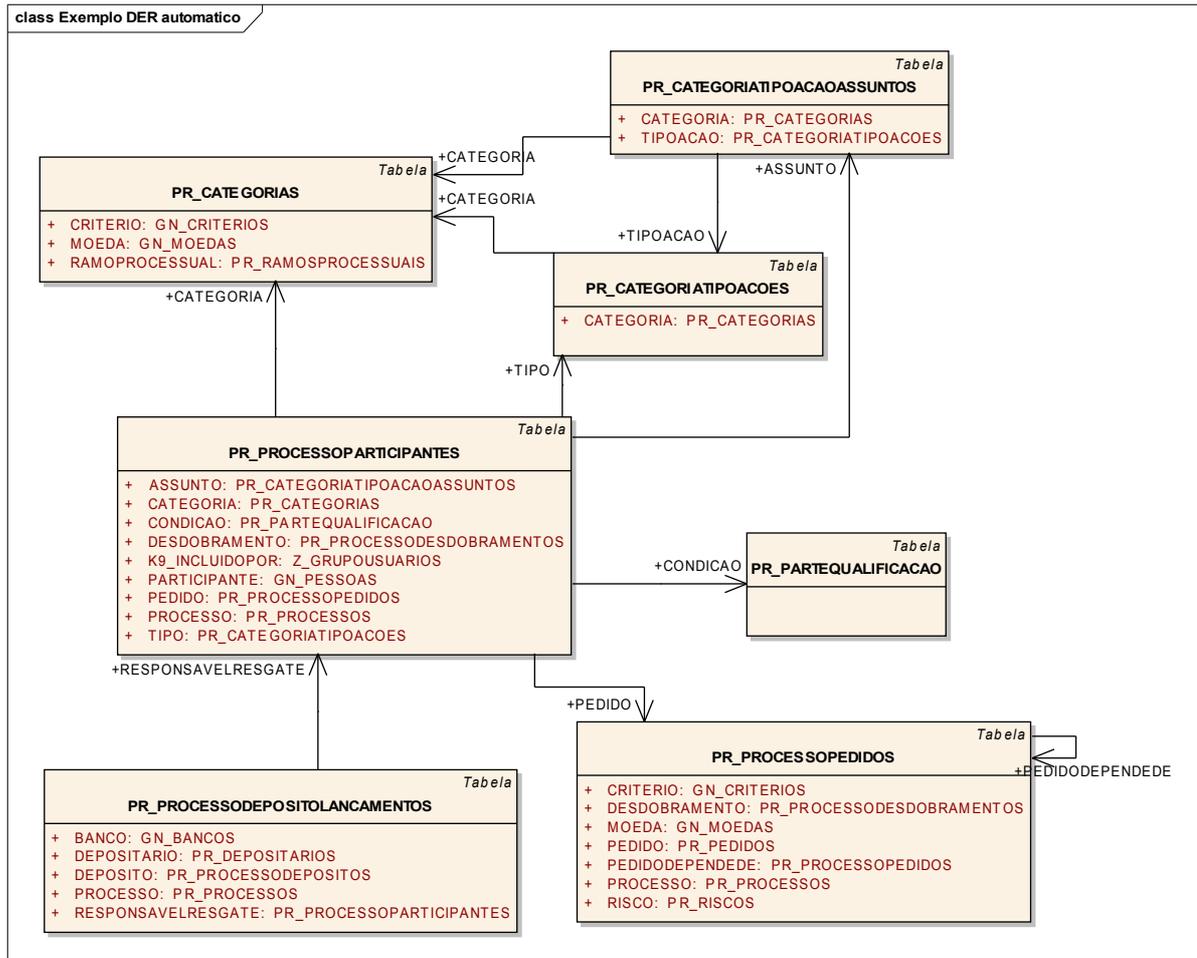


Figura 18 – Diagrama Entidade Relacionamento exemplo

A figura 19 ajuda a entender o funcionamento da rotina. Este diagrama foi feito manualmente apenas para demonstrar algumas opções de geração do Diagrama Entidade Relacionamento.

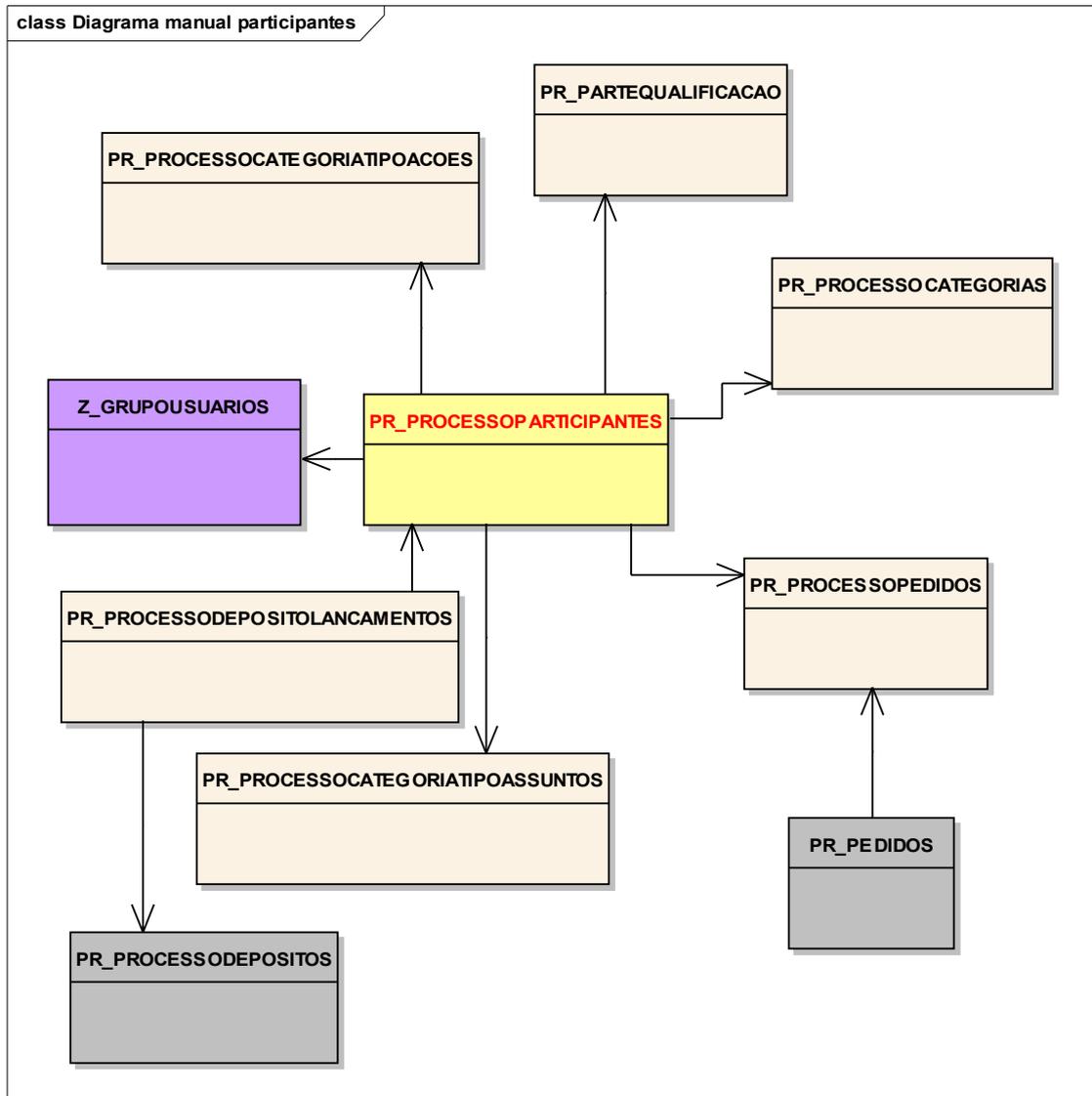


Figura 19 – Figura para demonstração

A tabela PR\_PROCESSOPARTICIPANTES é a principal da rotina, a partir dela que a rotina encontra as demais tabelas. As tabelas PR\_PROCESSOCATEGORIATIPOACOES, PR\_PARTEQUALIFICACAO, PR\_PROCESSOCATEGORIAS, PR\_PROCESSODEPOSITOLANCAMENTOS, PR\_PROCESSOPEDIDOS, PR\_PROCESSOTIPOASSUNTOS são tabelas de primeiro nível, as tabelas PR\_PEDIDOS E PR\_PROCESSODEPOSITOS são de segundo nível e não foram geradas pela rotina automática, pois estão em um nível a acima. A tabela Z\_GRUPOUSUARIOS também não foi inserida, pois não foi setado a opção de incluir tabelas de sistema no cadastro de rotinas.

### 3.5 DESENVOLVIMENTO DOS DIAGRAMAS

Nesta seção são apresentados alguns diagramas construídos. Como a modelagem é algo proprietário da empresa apenas alguns diagramas foram apresentados neste trabalho.

Neste trabalho foram documentadas todas as rotinas do módulo jurídico e financeiro. Em cada módulo foi documentado o modelo de objetivos e o diagrama de processos.

Os outros diagramas foram documentados por rotinas e o nível de detalhamento foi feito de acordo com a necessidade de cada rotina.

Em relação aos Casos de Uso a numeração é sempre reiniciada em cada pacote.

Para a construção dos diagramas foram utilizados alguns padrões da engenharia reversa, os padrões utilizados neste trabalho foram adaptados. Inicialmente foi executado o passo identificar. Este passo visa identificar as principais características do sistema e armazená-los em uma base de conhecimento, a base de conhecimento utilizada neste trabalho foi um arquivo texto.

Para a realização deste passo foi feita uma entrevista com um vendedor do sistema para descobrir quais os principais objetivos dos clientes na aquisição do sistema. Após isto foi feito um levantamento com base em algumas documentações dos processos de negócio atendidos pelo sistema e também foi identificado os Casos de Uso do sistema.

No passo organizar foi dividido a base de conhecimento em módulos e rotinas do sistema, e para finalizar este passo foi criado o modelo de documentação no EA através do Diagrama de Pacotes.

No último passo (recuperar) foram criados os diagramas propostos no modelo de documentação e a importação de arquivos para a geração do Diagrama Entidade Relacionamento.

A figura 20 mostra a tela do EA com a documentação concluída.

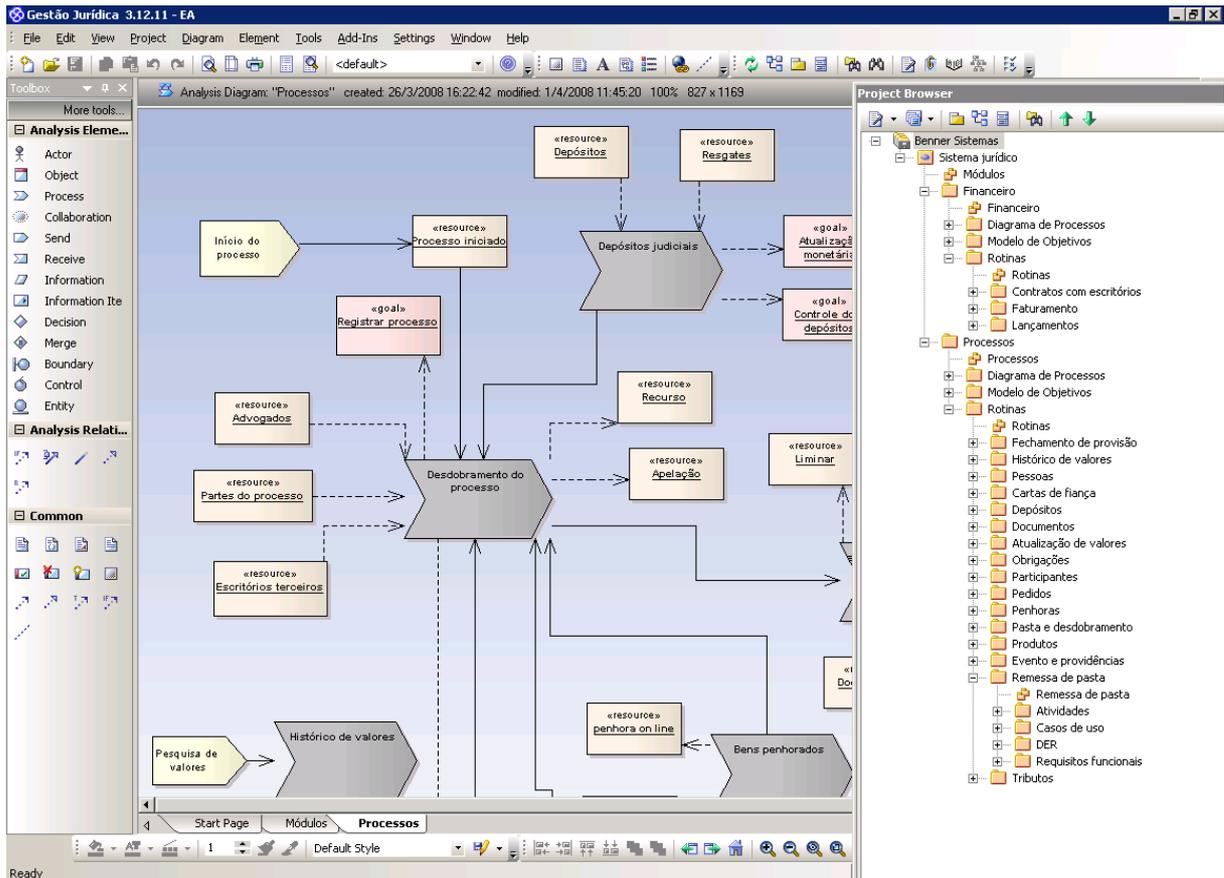


Figura 20 – Visão geral do EA com a documentação concluída

### 3.5.1 DIAGRAMAS DO MÓDULO JURÍDICO

Nesta seção será apresentado o modelo de objetivos, Diagrama de Processos do módulo jurídico.

#### 3.5.1.1 DIAGRAMA DE OBJETIVOS

A figura 21 mostra o diagrama geral de objetivos do módulo jurídico. Sua função é mostrar os problemas que o cliente quer resolver ao adquirir o sistema. Este diagrama deve ser obtido através de entrevista com o cliente.

O diagrama começa pela parte mais abstrata como mostra a figura em que o objetivo principal do cliente é controlar seus processos. Após esta identificação devemos descobrir o que é necessário para atender este objetivo, como por exemplo, para controlar processos

precisamos provisionar valores, registrar processos, aplicar correção monetária etc.

Este diagrama é complementado pelo diagrama de processos que vai mostrar como atingir os objetivos deste diagrama.

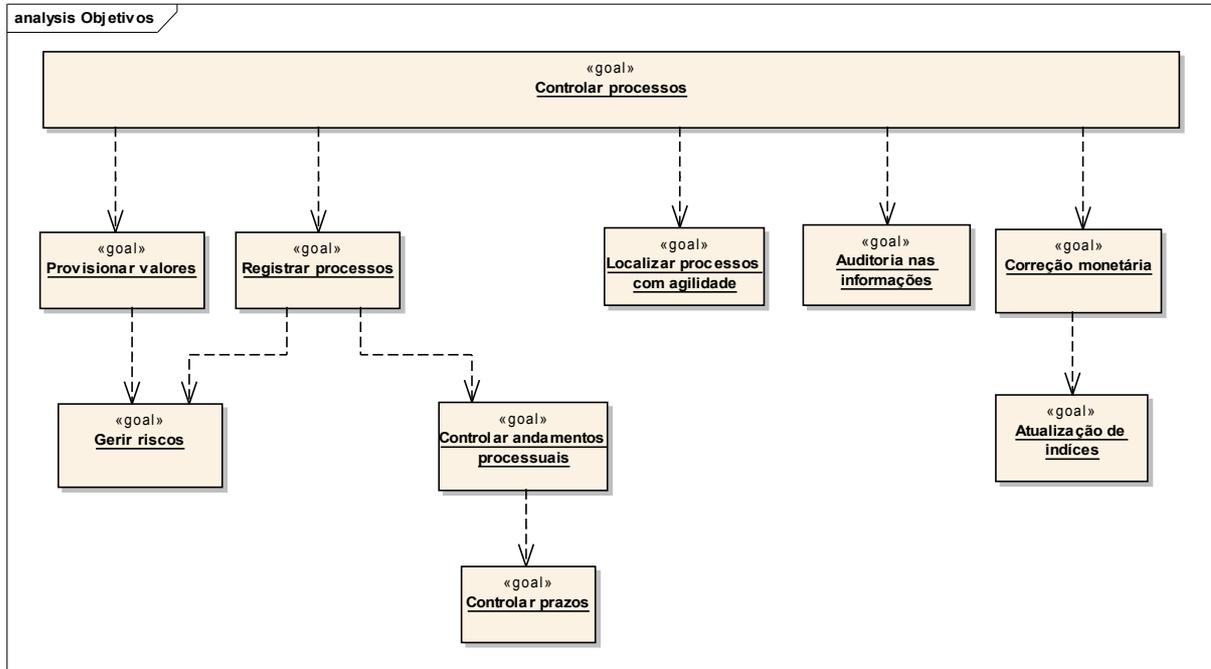


Figura 21 – Modelo de Objetivos do módulo jurídico

### 3.5.1.2 DIAGRAMA DE PROCESSOS

Este diagrama expressa a seqüência de atividades de negócio que transforma entradas em saídas de valor para a organização. Ele possui um propósito e um objetivo, é afetado por eventos do ambiente externo ou de outros processos, utiliza recursos de entrada e produz recursos de saída.

A figura 22 mostra o Diagrama de Processos do módulo jurídico. Pode-se notar um evento que inicia o fluxo dando origem ao desdobramento do processo. Um desdobramento do processo possui recursos como os advogados, as partes do processo (autor e réu), o desdobramento também gera recursos de saída como por exemplo, gerar uma apelação. O desdobramento do processo cumpre os objetivos de auditoria de informações e registros de processos. Após o desdobramento são iniciados os andamentos processuais que possuem seus próprios recursos e objetivos.



### 3.5.2 ROTINAS DO MÓDULO JURÍDICO

Após a construção do modelo de objetivos e do Diagrama de Processos foram construídos os outros diagramas. Estes diagramas foram separados por pacotes, ou seja cada rotina do módulo contém seu pacote de diagrama e o nível de detalhamento é feito de acordo com a necessidade de cada rotina.

#### 3.5.2.1 FECHAMENTO DE PROVISÃO

Na rotina de fechamento de provisão foi desenvolvido o Diagrama de Atividades, Diagrama de Casos de Uso com cenários e os Requisitos Funcionais. A figura 23 mostra o Diagrama de Atividades. A figura 24 mostra o Diagrama de Casos de Uso. A figura 25 mostra os requisitos funcionais e a figura 26 mostra o Diagrama Entidade Relacionamento da rotina concebida automaticamente através da rotina.

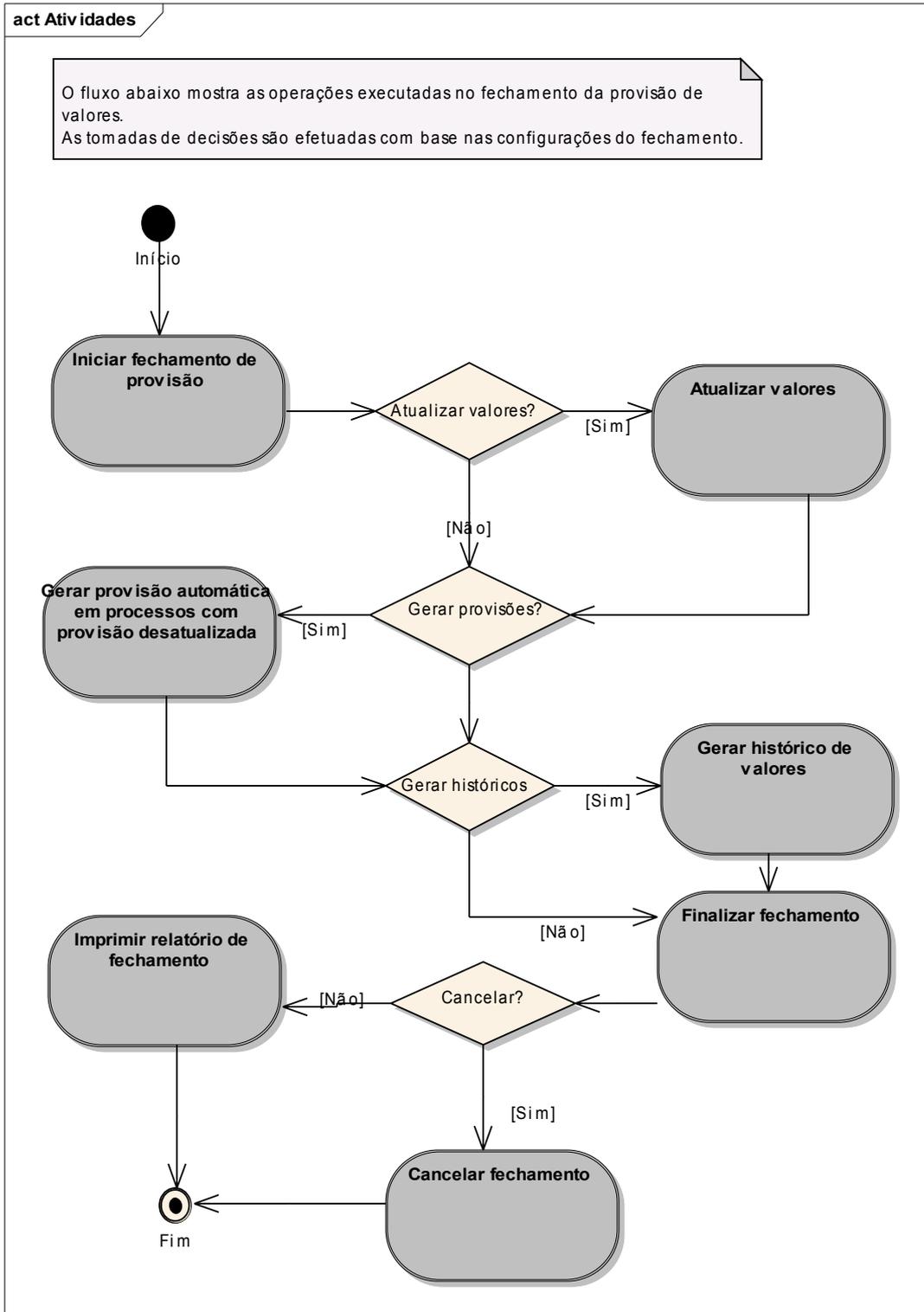


Figura 23 – Diagrama de Atividades do fechamento de provisão

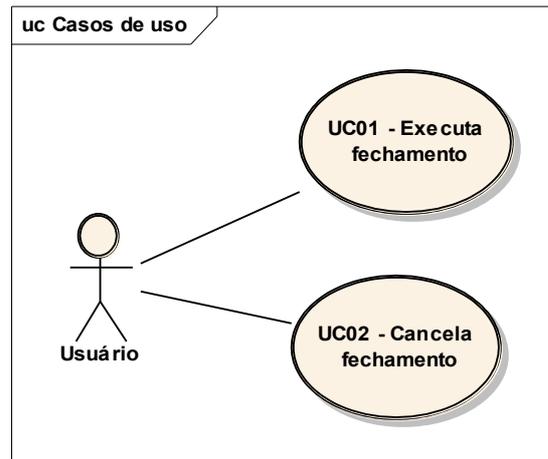


Figura 24 – Diagrama de Casos de Uso do fechamento da provisão

#### Executa fechamento

##### Principal

1. O sistema apresenta a tela para cadastro de fechamento. 2. O usuário informa a competência e salva o registro. 3. O usuário pressiona o botão executar fechamento. 4. O sistema emite uma mensagem avisando que o fechamento foi iniciado. 5. Após gerar todos os históricos de valores o sistema altera o status para executado.

##### Alternativo - Competência já cadastrada

2.1 Se existir um fechamento cadastrado com a mesma competência o sistema não salva o registro e emite uma mensagem ao usuário.

Quadro 15 – Cenários do Caso de Uso executa fechamento

#### Cancela fechamento

##### Principal

1. O sistema apresenta o fechamento cadastrado. 2. O usuário altera o campo cancelar fechamento para sim e salva o registro. 3. O sistema exclui todos os registros referentes ao fechamento e emite uma mensagem.

Quadro 16 – Cenários do Caso de Uso cancela fechamento

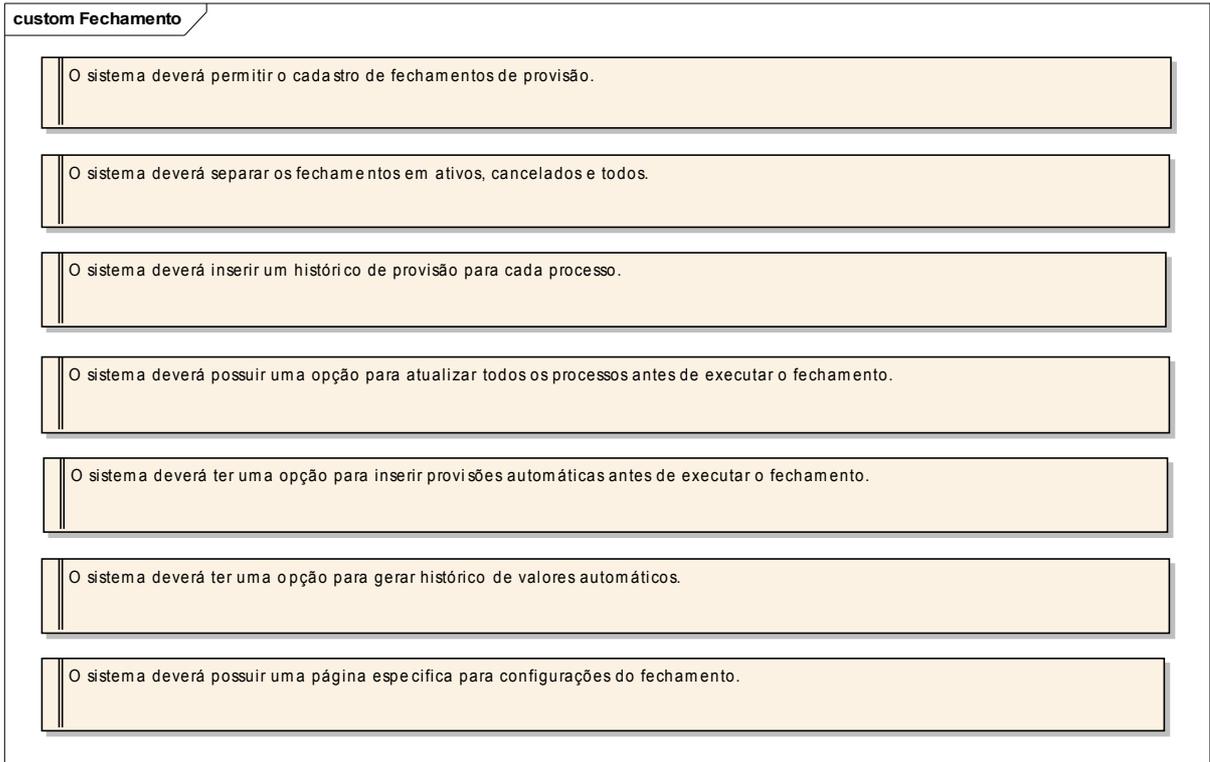


Figura 25 – Requisitos Funcionais do fechamento de provisão

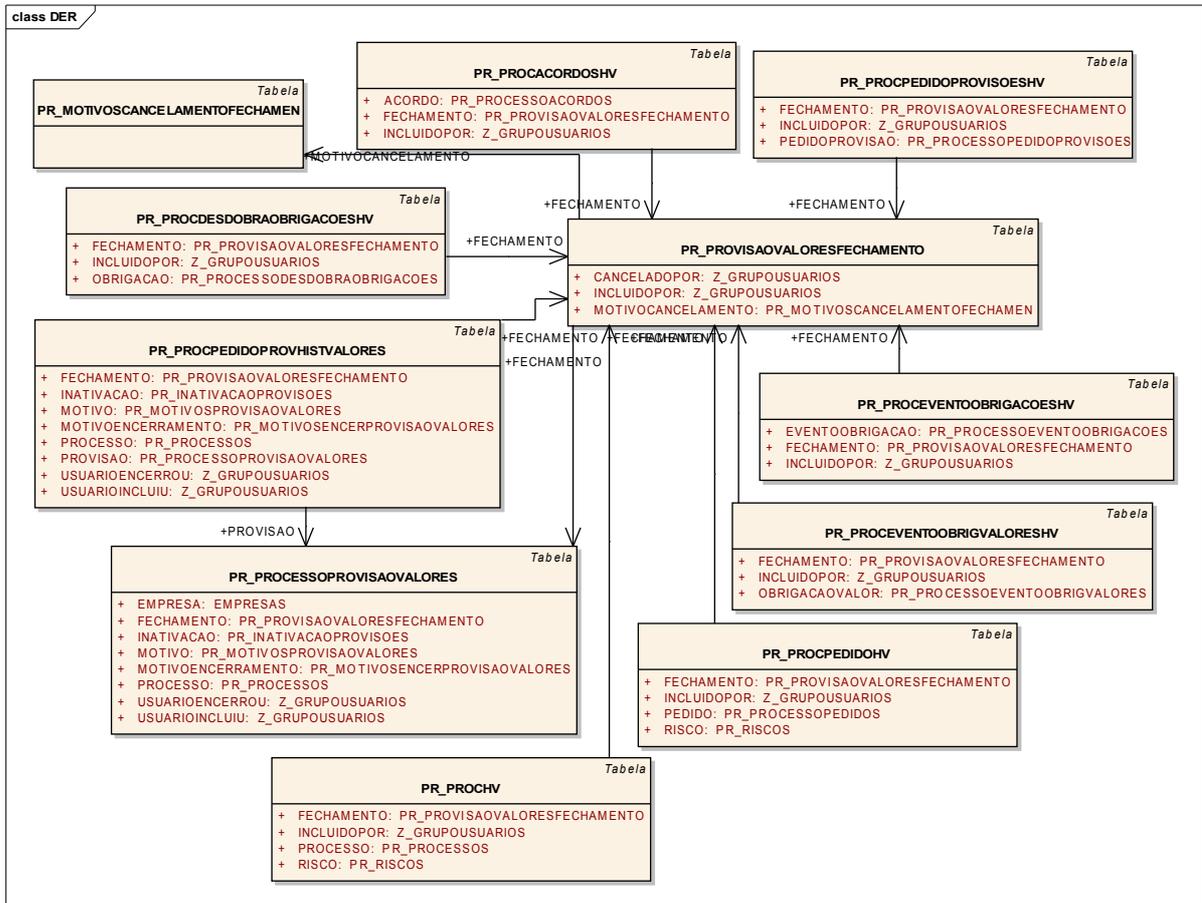


Figura 26 – Diagrama Entidade Relacionamento do fechamento de provisão

### 3.5.2.2 OBRIGAÇÕES

Na rotina de obrigações foi desenvolvido o Diagrama de Casos de Uso e os Requisitos Funcionais. A figura 27 mostra o Diagrama de Casos de Uso, a figura 28 mostra os requisitos funcionais.

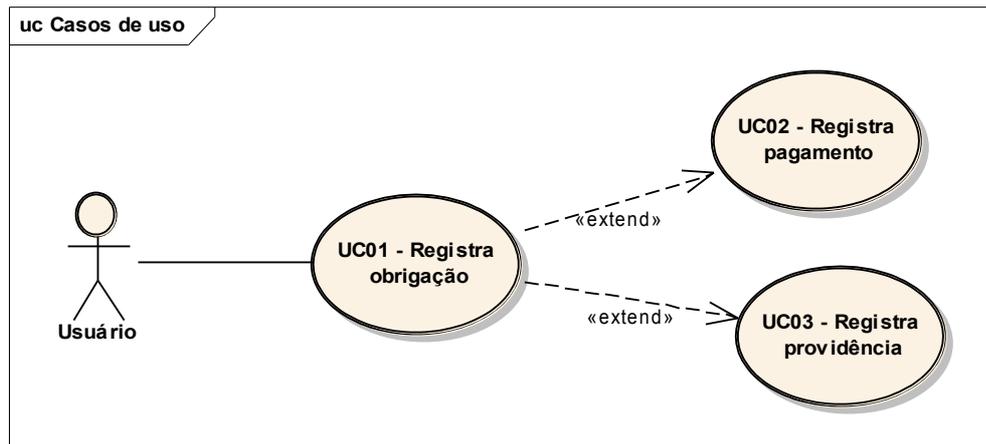


Figura 27 – Diagrama de Casos de Uso de obrigações

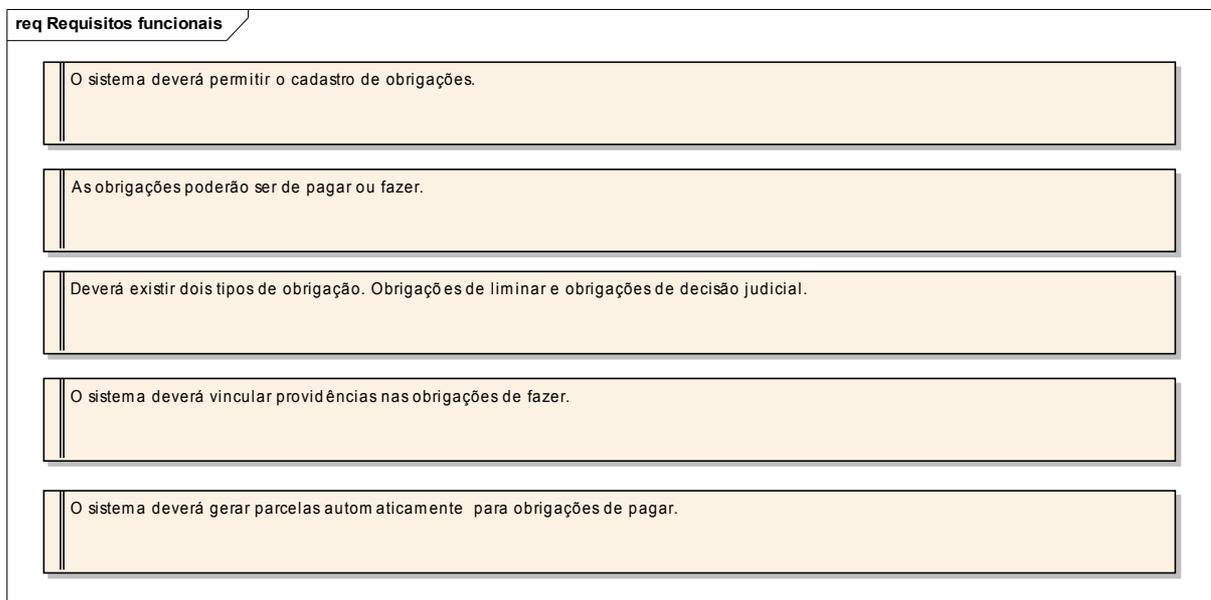


Figura 28 – Requisitos Funcionais de obrigações

### 3.5.2.3 PEDIDOS

Na rotina de pedidos foi desenvolvido o Diagrama de Atividades, Diagrama de Casos de Uso e os Requisitos Funcionais. A figura 29 mostra o Diagrama de Atividades, a figura 30 mostra o Diagrama de Casos de Uso, a figura 31 mostra os requisitos funcionais da rotina, a figura 32 mostra o Diagrama Entidade Relacionamento da rotina concebida automaticamente através da rotina.

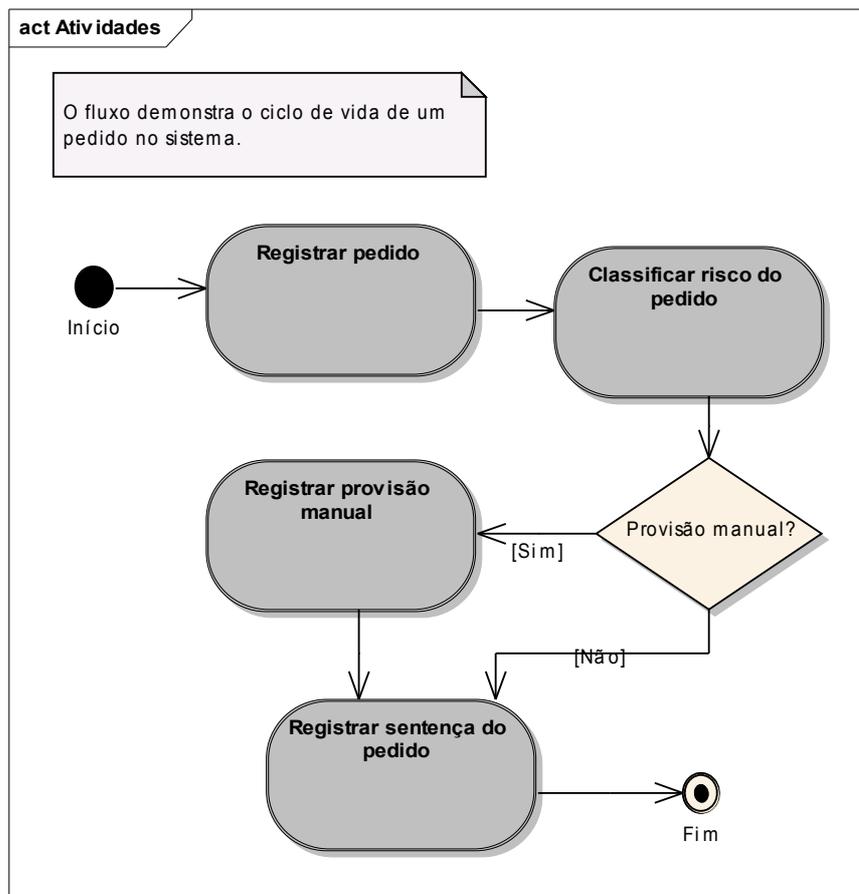


Figura 29 – Diagrama de Atividades de pedidos

No Diagrama de Casos de Uso de pedidos não houve necessidade de documentar os cenários, pois a rotina de pedidos é basicamente cadastral.

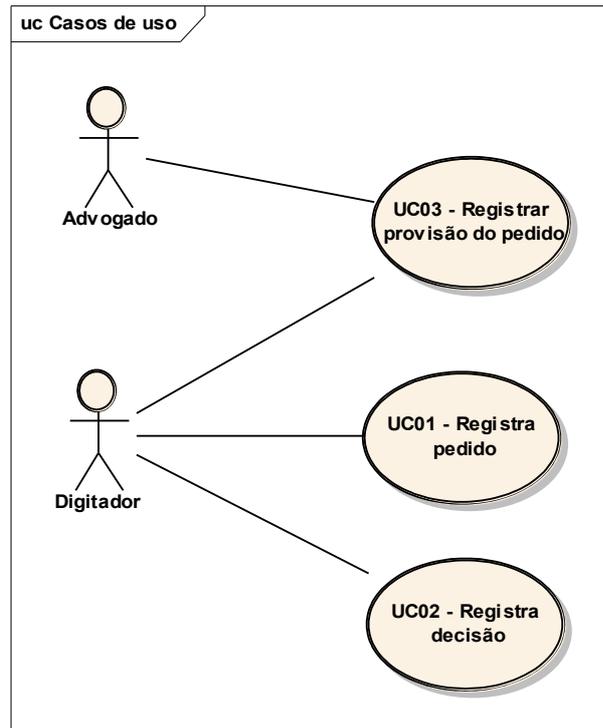


Figura 30 – Diagrama de Casos de Uso de pedidos

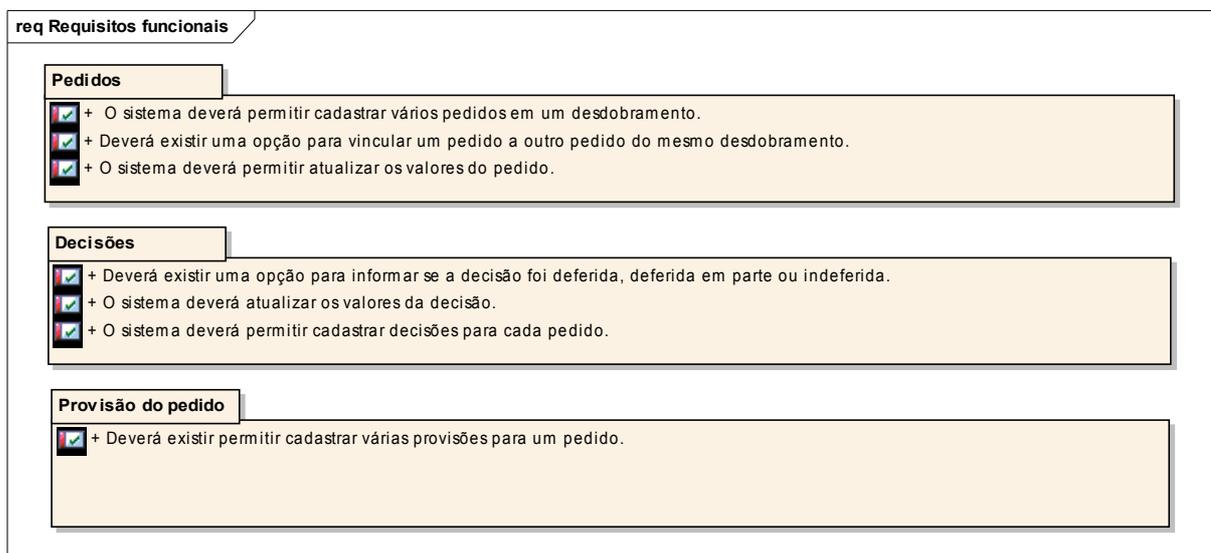


Figura 31 – Requisitos Funcionais da rotina de pedidos

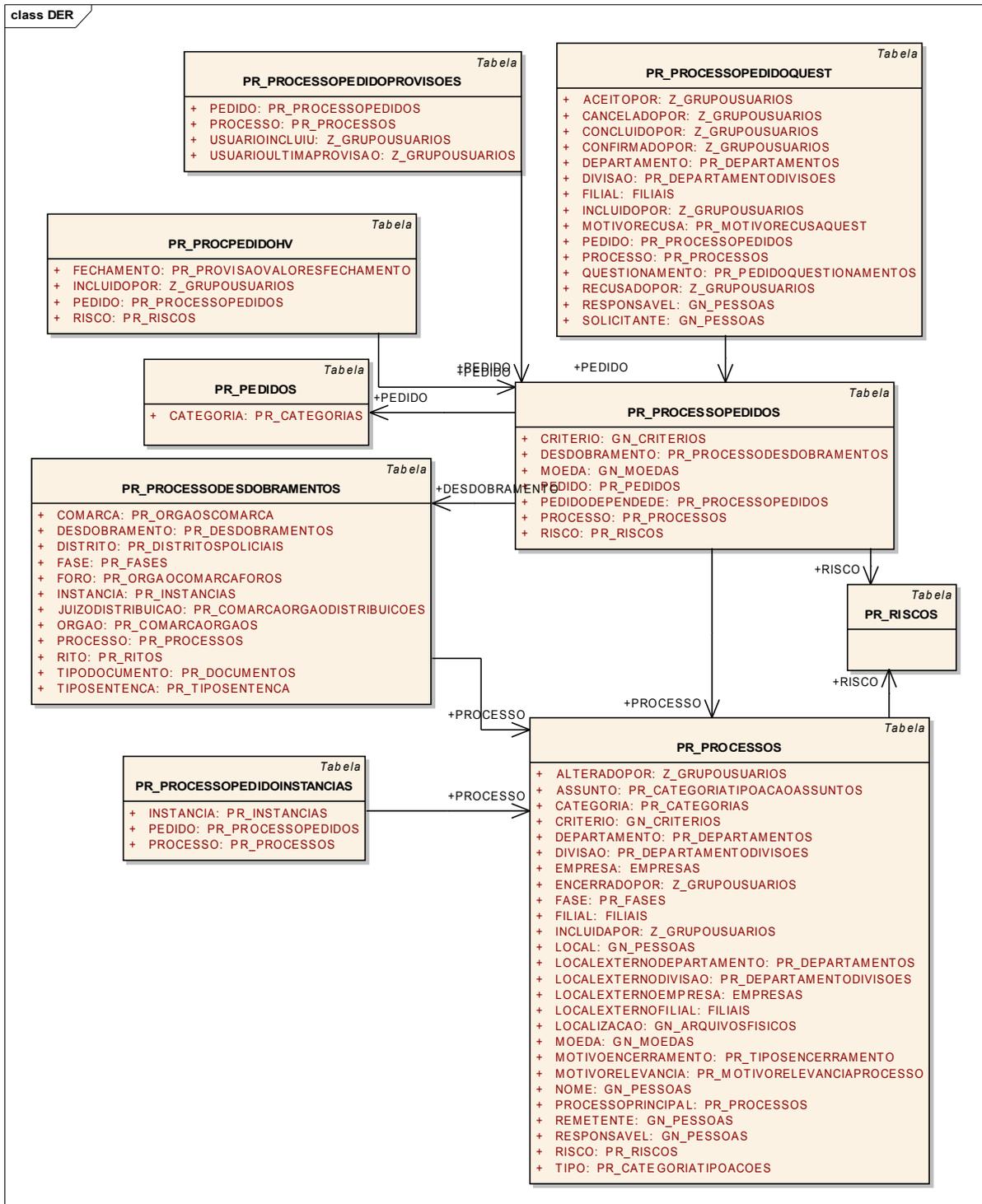


Figura 32 – Diagrama Entidade Relacionamento de pedidos

#### 3.5.2.4 REMESSA E RECEPÇÃO DE PASTAS

Na rotina de remessa e recepção de pastas foi desenvolvido o Diagrama de Atividades, Diagrama de Casos de Uso, Requisitos Funcionais e o Diagrama Entidade Relacionamento. A figura 33 mostra o Diagrama de Atividades, a figura 34 mostra o Diagrama de Casos de Uso, a figura 35 mostra o Diagrama Entidade Relacionamento da rotina concebida automaticamente através da rotina.

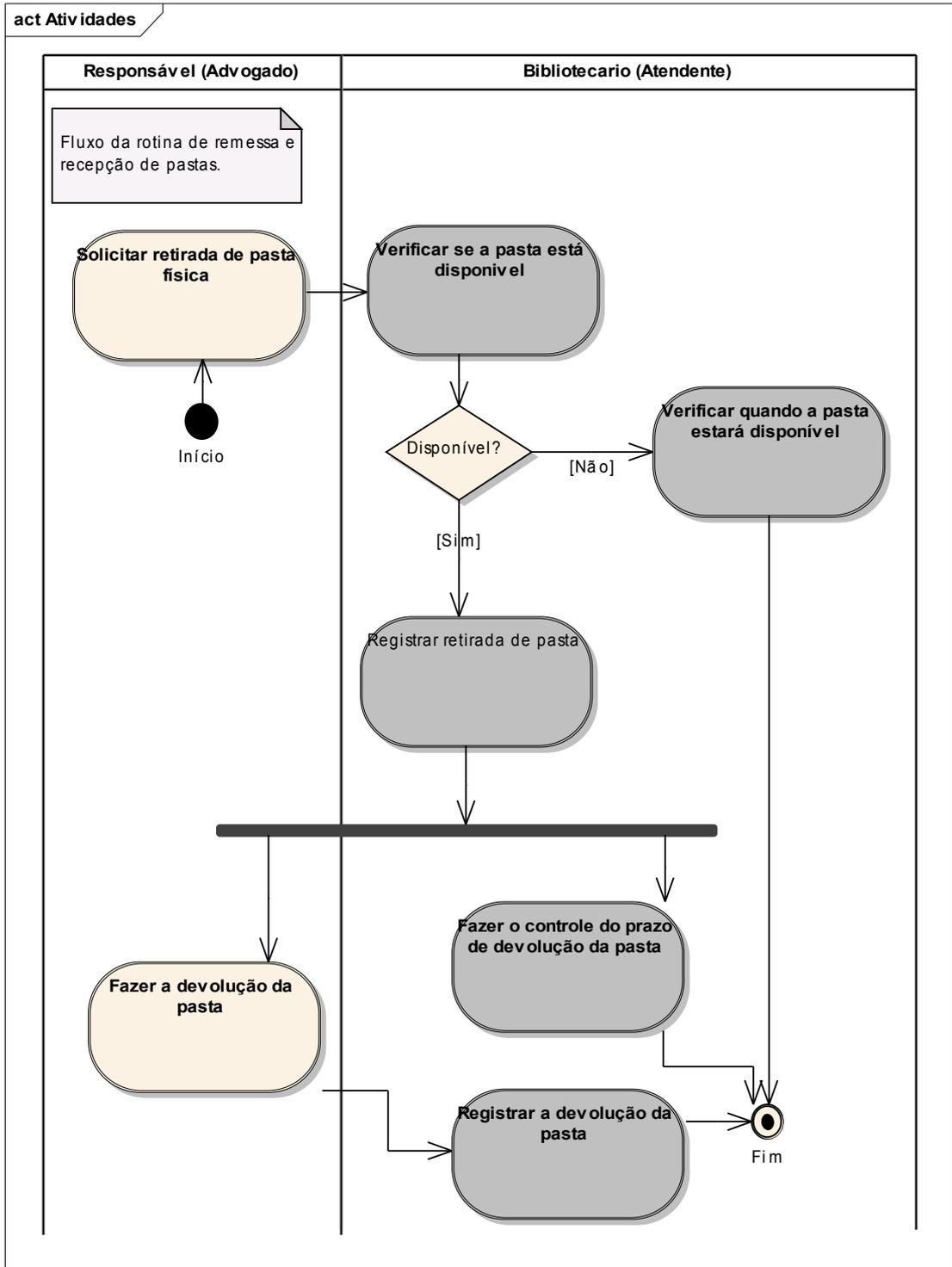


Figura 33 – Diagrama de Atividades de remessa de pastas

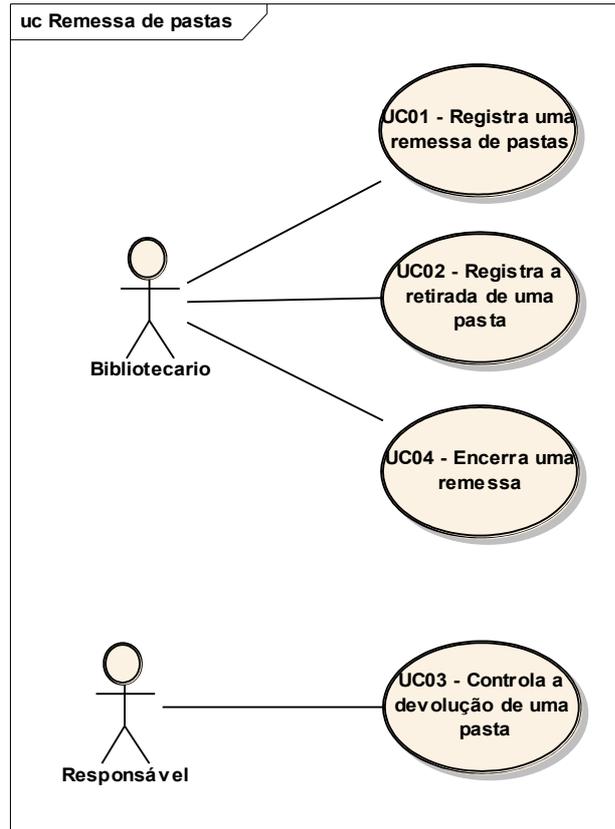
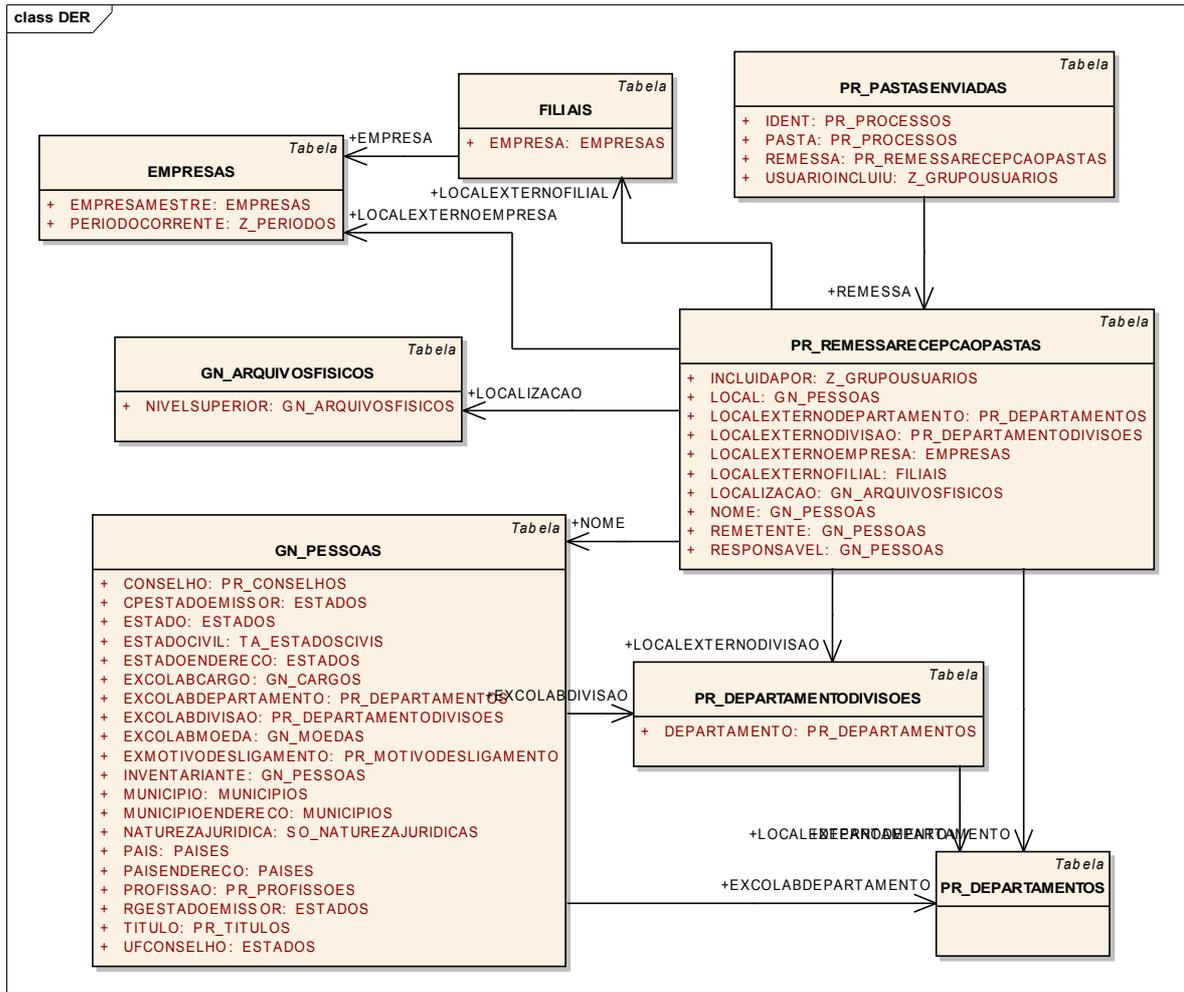


Figura 34 – Diagrama de Casos de Uso de remessa



### 3.5.2.5 EVENTOS E PROVIDÊNCIAS

Na rotina de evento e providências foi desenvolvido o Diagrama de Casos de Uso, Diagrama de Atividades e os Requisitos Funcionais. A figura 36 mostra o Diagrama de Atividades, a figura 37 mostra o Diagrama de Casos de Uso, a figura 38 mostra o Diagrama Entidade Relacionamento da rotina concebida automaticamente através da rotina.

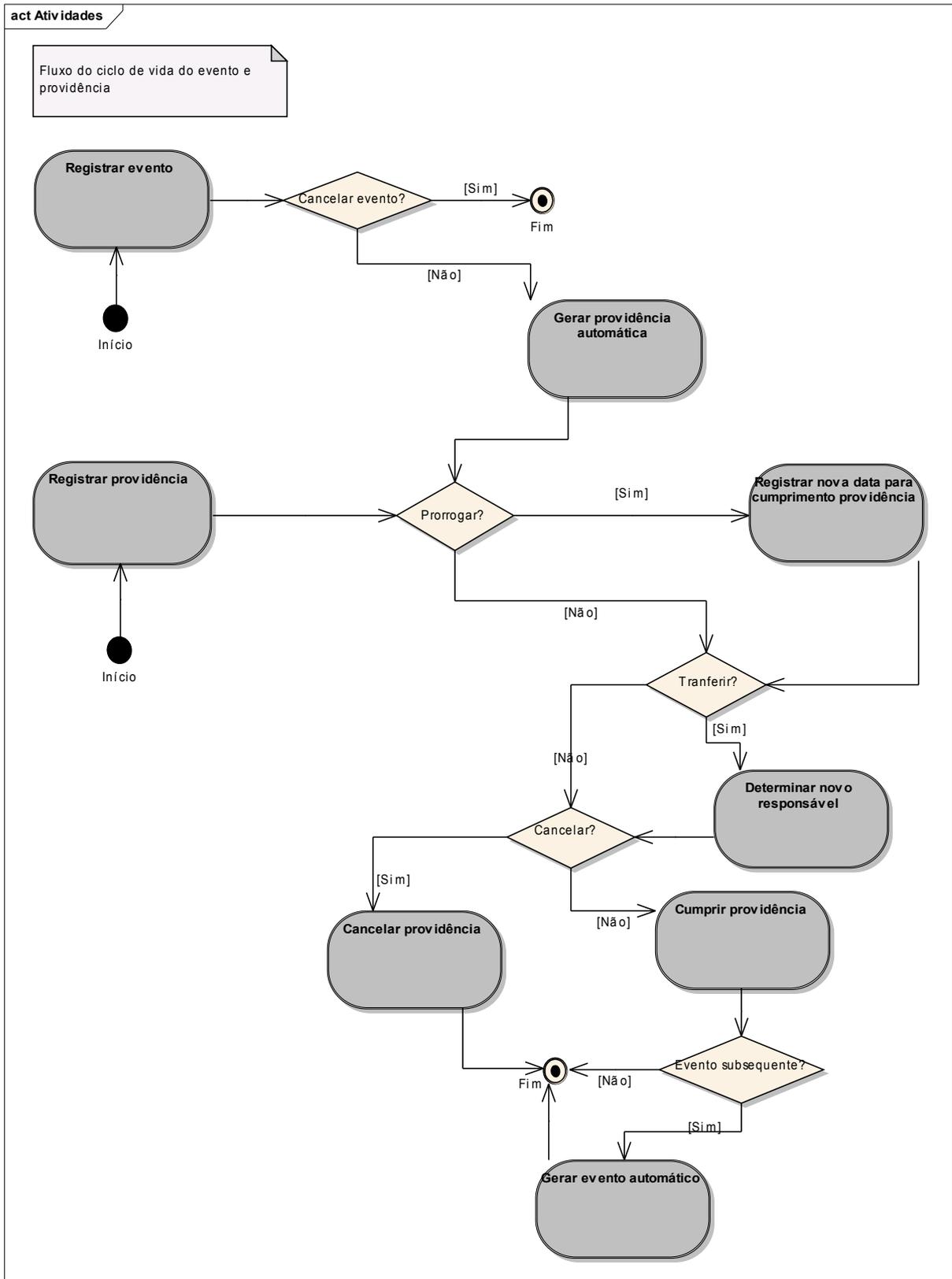


Figura 36 – Diagrama de Atividades de providências

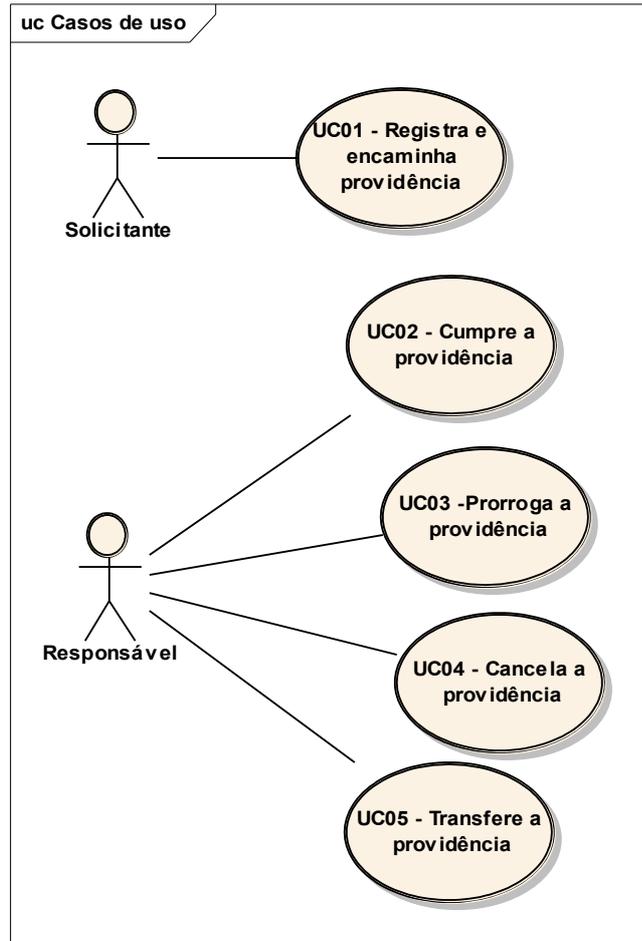


Figura 37 – Diagrama de Casos de Uso de providências

### Cancela providência

#### Principal

1. O sistema apresenta a tela com a providência cadastrada. 2. O usuário aperta o botão cancelar providência. 3. O sistema exibe uma mensagem de confirmação de cancelamento da providência.

Quadro 17 – Caso de Uso cancela providência

### Cumprimento de providência

#### Principal

1. O sistema apresenta a tela com a providência cadastrada. 2. O usuário indica que a providência foi cumprida, informa a data do cumprimento e salva o registro

Quadro 18 – Caso de Uso cumprimento de providência

### Prorroga providência

**Principal**

1. O sistema exibe a tela com a providência cadastrada.2. O usuário aperta o botão prorrogar.3. O usuário escolhe a nova data e salva o registro.4. O sistema inclui abaixo da tabela um log com os dados da prorrogação da providência

Quadro 19 – Caso de Uso prorroga providência

**Registra e encaminha providência****Principal**

1. O sistema apresenta a tela para cadastro da providência.2. O usuário preenche os dados básicos e salva a o registro.3. Após a providência salva ela estará disponível na agenda do responsável

**Tipos de cumprimento de providência – Alternativo**

O campo cumprir define a data de cumprimento da providência2.1 Caso o usuário escolheu a opção cumprir até o responsável terá que cumprir a providência até a data determinado no campo data final.2.2 Caso o usuário escolheu a opção cumprir em o responsável deverá cumprir a providência no dia a hora definido no campo data certa.2.3 Caso o usuário escolheu a opção sem data não é verificada data limite para cumprir a providência.

**Alerta via email - Alternativo**

2.1 Caso o usuário optou pela opção de alerta sobre no cumprimento de providência.2.2 O usuário informa os dias de antecedência para o cumprimento.2.3 O sistema envia um email ao responsável X dias antes do prazo para cumprimento da providência.

Quadro 20 – Caso de Uso registra e encaminha providência

**Transferir providência****Principal**

1. O sistema apresenta a tela com a providência cadastrada.2. O usuário aberta o botão transferir providência.3. O usuário escolhe a nova responsável e salva o registro.4. O sistema exibe a providência na agenda do novo responsável.

Quadro 21 – Cenários do Caso de Uso transferir providência

### 3.5.3 MÓDULO FINANCEIRO

Nesta seção será apresentado o modelo de objetivos, Diagrama de Processos do módulo financeiro.

### 3.5.3.1 MODELO DE OBJETIVOS

A figura 38 mostra o diagrama geral de objetivos do módulo financeiro.

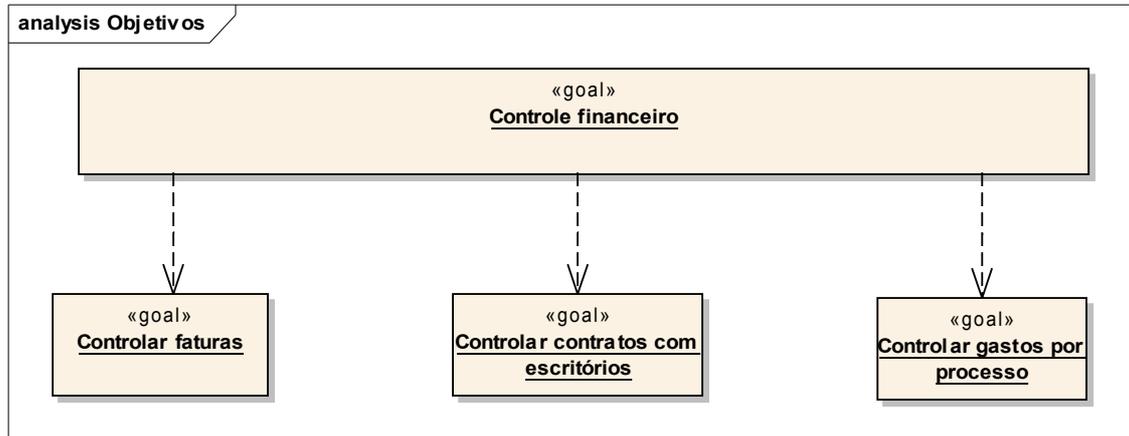


Figura 38 – Modelo de Objetivos do financeiro

### 3.5.3.2 DIAGRAMA DE PROCESSOS

O diagrama de processos da figura 39 mostra a seqüência de atividades de negócio que transforma entradas em saídas de valor para a organização.

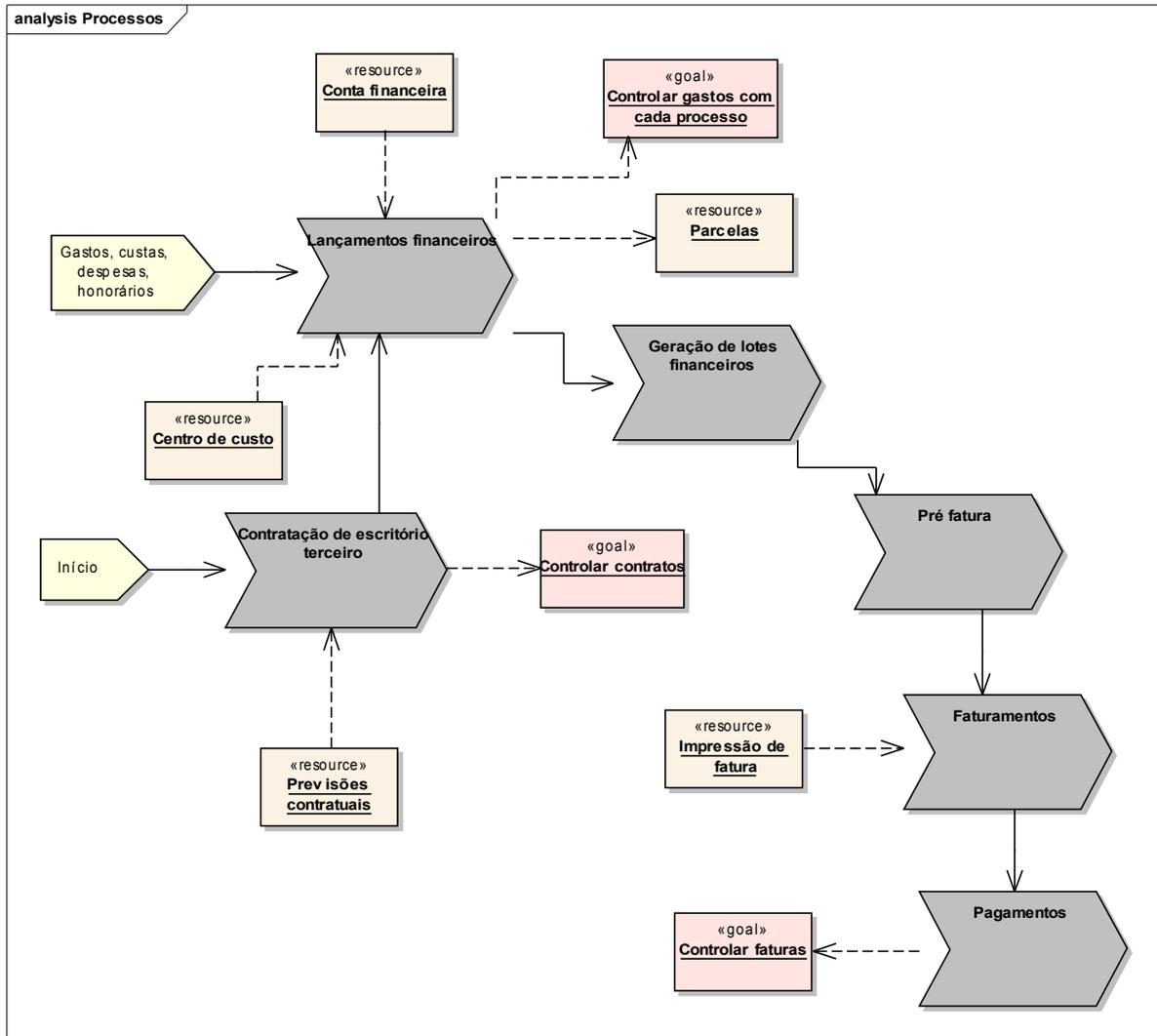


Figura 39 – Diagrama de Processos do financeiro

### 3.5.4 ROTINAS DO MÓDULO FINANCEIRO

Após a construção do modelo de objetivos e do Diagrama de Processos foram construídos os outros diagramas do módulo financeiro.

#### 3.5.4.1 CONTRATOS COM ESCRITÓRIOS EXTERNOS

Na rotina de contratos foi desenvolvido o Diagrama de Casos de Uso e os Requisitos Funcionais. Os cenários dos Casos de Uso não foram documentados, pois a rotina de contratos com escritórios exige pouca interação entre o usuário e o sistema. A figura 40

mostra o Diagrama de Casos de Uso, a figura 41 mostra os requisitos funcionais.

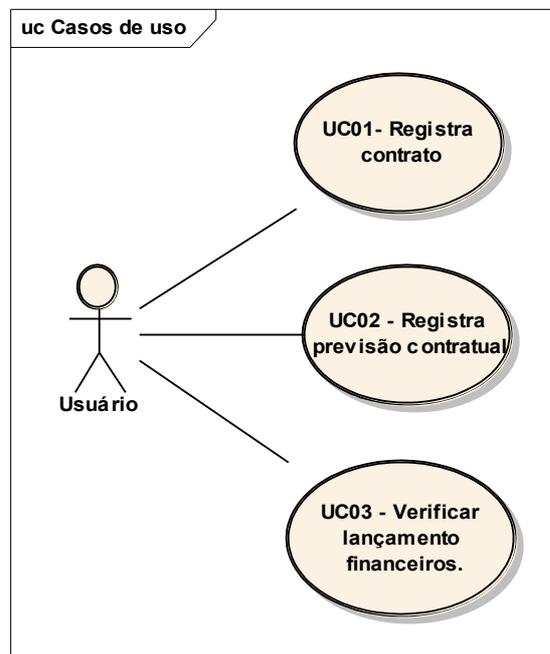


Figura 40 – Diagrama de Casos de Uso dos contratos

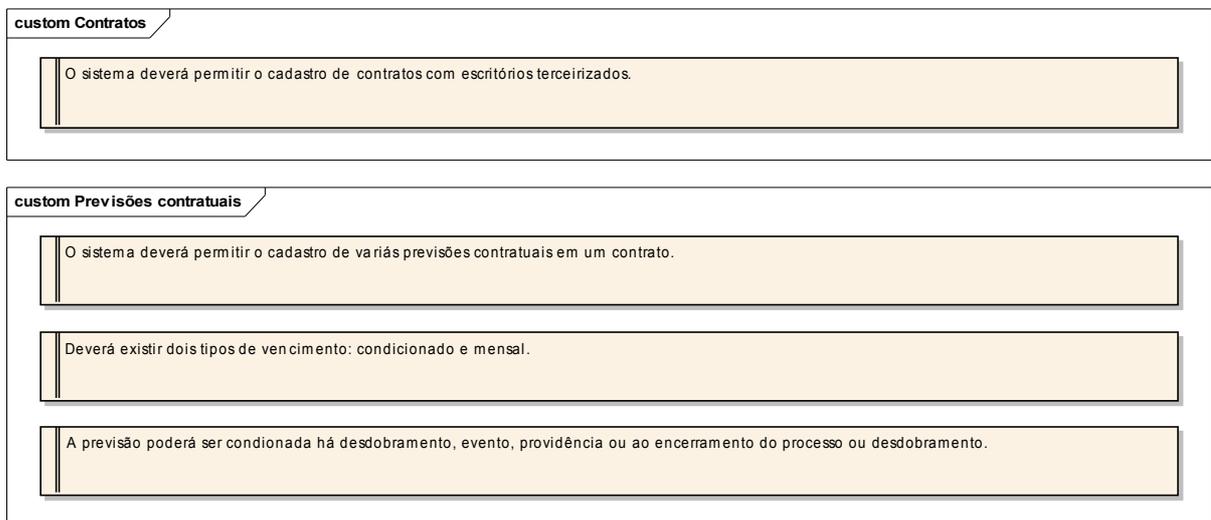


Figura 41 – Requisitos Funcionais dos contratos

### 3.5.4.2 FATURAMENTOS

Na rotina de faturamentos foi desenvolvido o Diagrama de Casos de Uso, Diagrama de Atividades e os Requisitos Funcionais.

A figura 42 mostra o Diagrama de Atividades da rotina. Para gerar um faturamento o

primeiro passo é gerar um lote de lançamentos financeiros, estes lotes são agrupados por data, pessoa e centro de custo. Após gerar o lote o usuário pode conferir os lançamentos que foram associados ao lote, caso os lançamentos estejam certos é necessário gerar uma pré fatura e depois a fatura, pode ser configurado que só determinados usuários poderão gerar a fatura através de grupos de segurança. A figura 43 mostra os Casos de Uso e a figura 44 apresenta o Diagrama Entidade Relacionamento gerado automaticamente pela rotina.

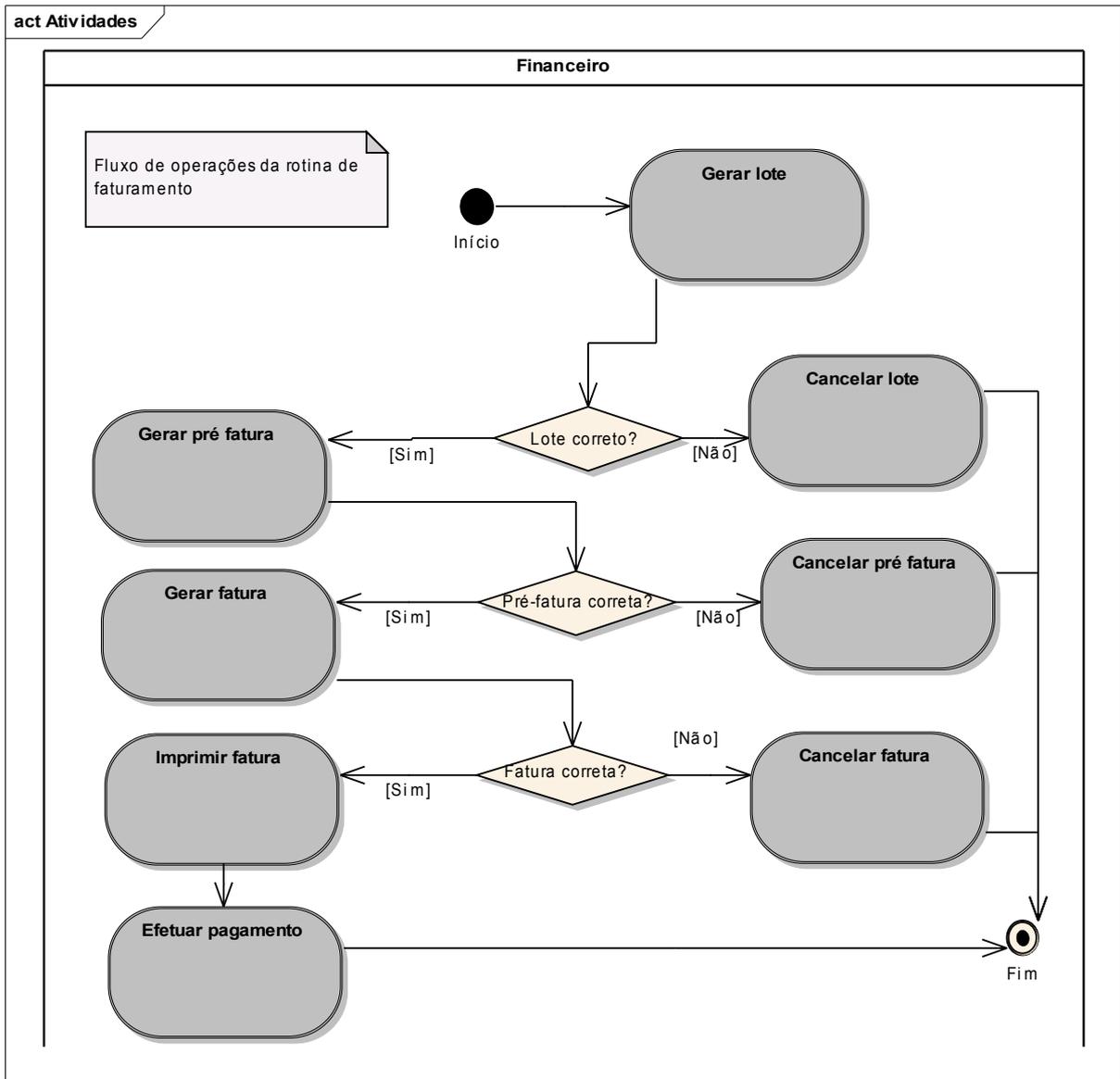


Figura 42 – Diagrama de Atividades do faturamento

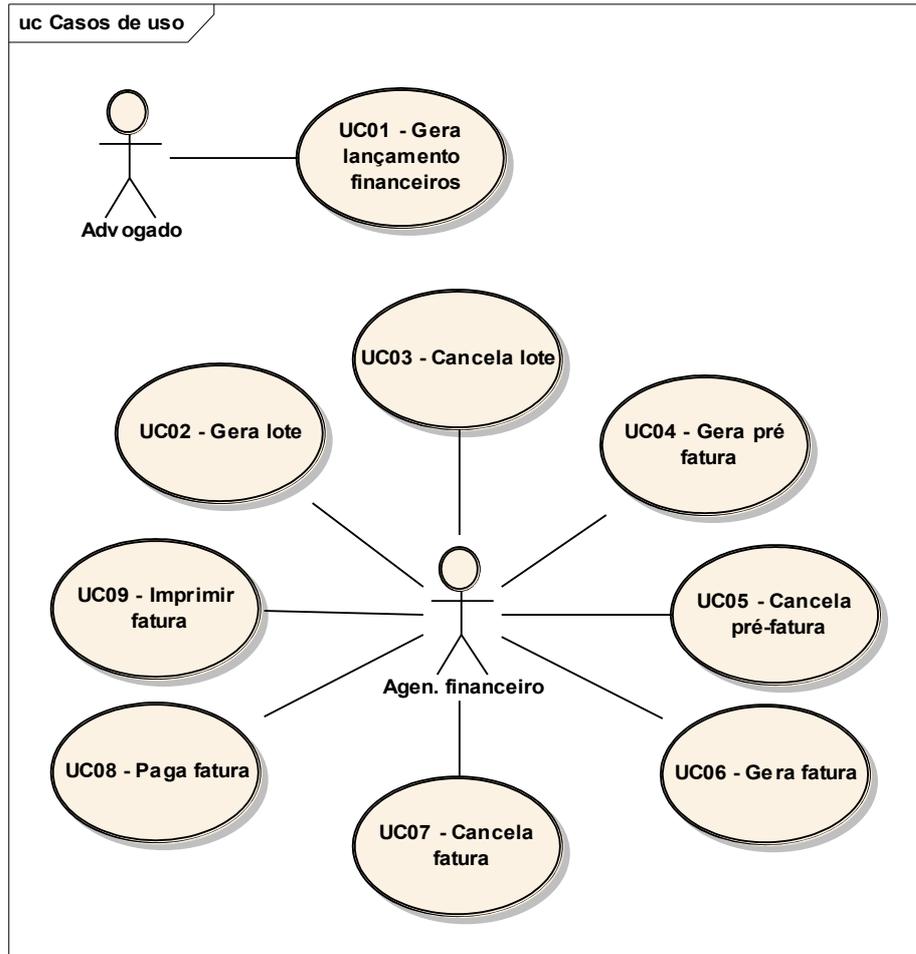
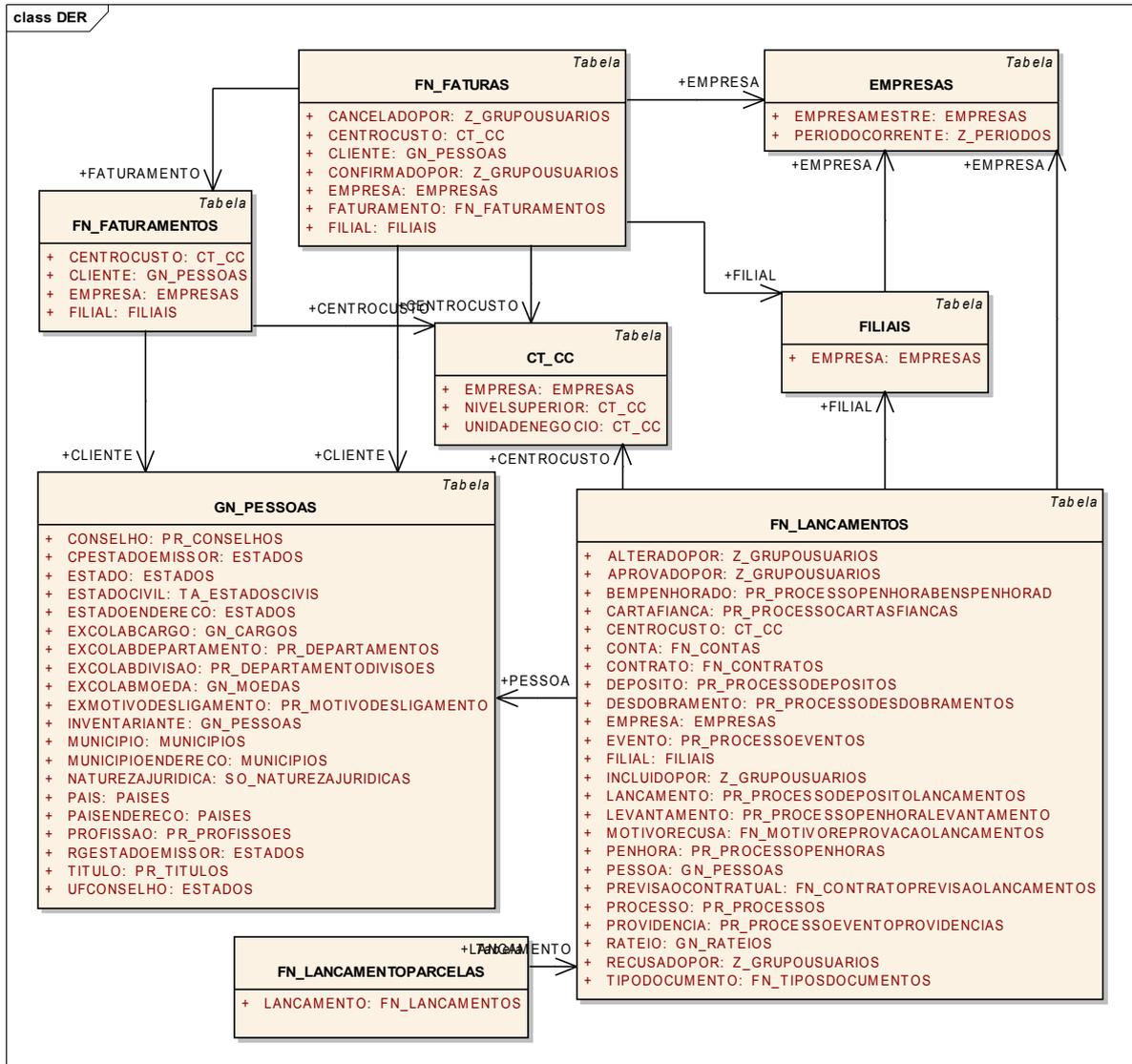


Figura 43 – Diagrama de Casos de Uso do faturamento



### 3.6 DOCUMENTAÇÃO HTML

O EA oferece uma opção para gerar a documentação no formato HTML. Com esta opção é possível visualizar a documentação sem a necessidade de instalação da ferramenta. Além disso, fica fácil a navegação pela documentação. A figura 45 mostra a documentação gerada pela ferramenta.

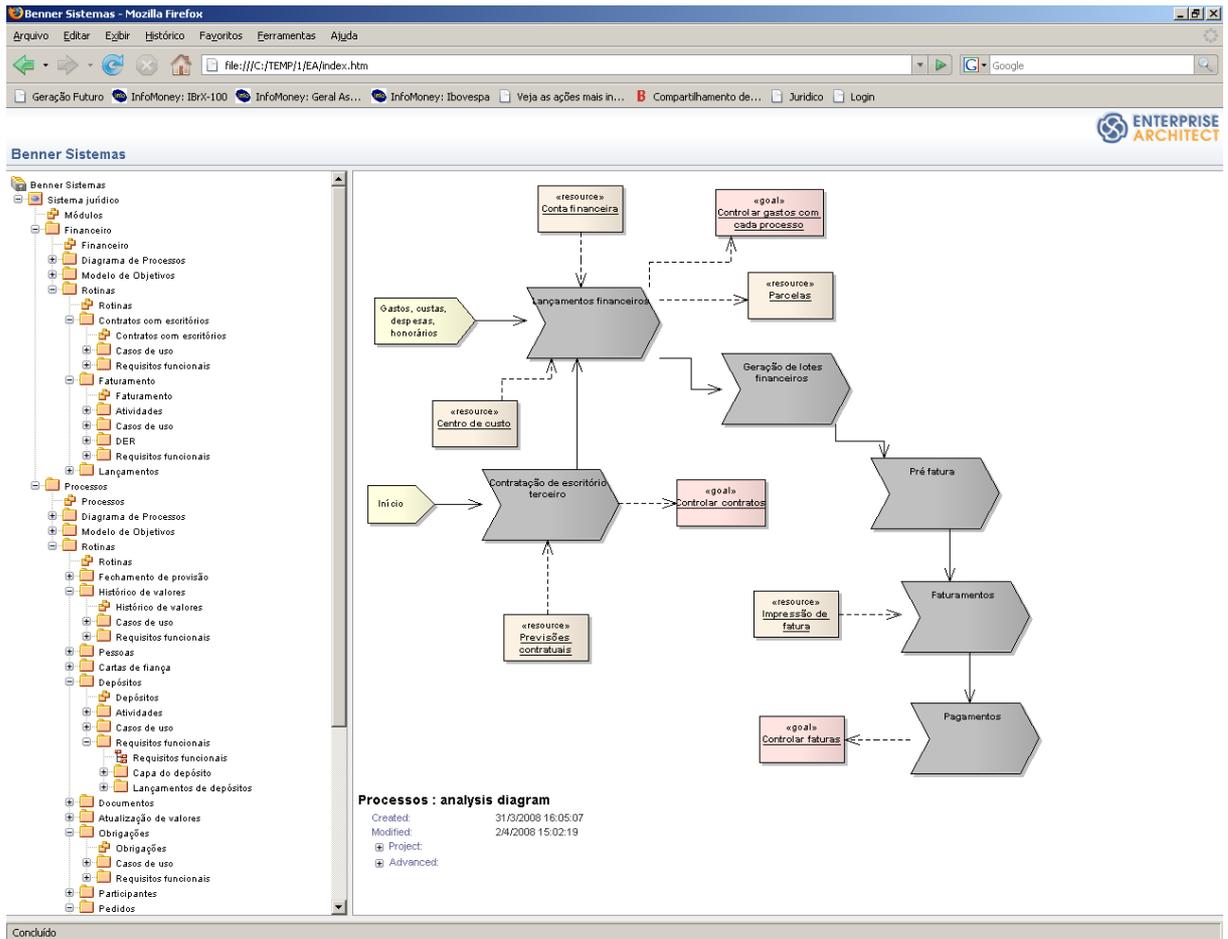


Figura 45 – Documentação HTML gerada pelo EA

### 3.7 RESULTADOS E DISCUSSÃO

Após a engenharia reversa realizada no sistema Gestão de Processos o sistema está documentado. Com isto ficou mais fácil o entendimento do sistema por usuários, desenvolvedores e equipe de suporte. Os programadores terão uma visão mais clara do funcionamento do sistema e poderão fazer manutenções de forma mais segura.

Novas funcionalidades no sistema poderão ser desenvolvidas com base nos diagramas gerados. A documentação servirá também para controlar e centralizar informações.

Nas próximas consultas dos diagramas será necessário, no entanto comparar com as funcionalidades do sistema para verificar possíveis erros no processo de engenharia reversa.

Em termos de tempo de desenvolvimento possivelmente o uso dos diagramas não tornará o processo mais ágil, porém será resolvido o problema de comunicação entre consultores de negócio, analista de sistema e programadores.

A documentação do sistema também funciona como selo de qualidade do sistema. Alguns clientes inclusive exigem a documentação do sistema via UML.

O quadro 21 apresenta a quantidade de alguns componentes utilizados na documentação do sistema.

Componente	Quantidade
Casos de Uso	93
Requisitos funcionais	117
Atividades	80
Pacotes	116
Atores	22
Cenários	101

Quadro 22 – Quantidade de componentes utilizados

Analisando os trabalhos correlatos foi verificado que no trabalho feito por Scarton é feito um comparativo entre algumas ferramentas CASE com o recurso de engenharia reversa, porém o foco do trabalho não é realizar a engenharia reversa de um sistema e sim comparar ferramentas.

Heckmann em sua monografia construiu uma ferramenta para realizar engenharia

reversa de sistemas desenvolvidos na linguagem FoxPro. Esta ferramenta não gera diagramas UML e pode ser utilizada apenas para sistemas desenvolvidos em FoxPro.

Koller desenvolveu uma ferramenta específica para geração de diagrama E-R a partir de um esquema gerado para o Banco ORACLE, mas neste trabalho não é gerado diagramas UML.

## 4 CONCLUSÕES

A engenharia reversa do sistema Gestão de Processos contribuiu para a documentação do sistema através da construção de diagramas.

O primeiro passo da construção dos Diagramas foi criar um modelo de documentação utilizando o Diagrama de Pacotes da UML. Com este modelo os diagramas ficaram organizados de maneira que ficou fácil a localização da documentação de alguma rotina.

A construção do Diagrama Entidade Relacionamento foi automatizada através de uma rotina construída na linguagem DELPHI. Esta rotina lê as informações na ferramenta de desenvolvimento Builder possibilitando o usuário gerar um arquivo. Este arquivo é importado na Ferramenta CASE EA gerando o Diagrama. Como o sistema é de médio porte e existe um número muito grande de tabelas foi criada uma opção para gerar o diagrama por rotinas, com isto é possível visualizar apenas as ligações entre as tabelas da rotina consultada.

Foi constatado que a rotina precisa de algumas alterações para o uso da empresa, por exemplo, o Diagrama Entidade Relacionamento deverá ser gerado por módulo do sistema e não um para cada rotina como foi feito neste trabalho.

Os diagramas foram criados em diferentes níveis de abstração, o de mais alto nível é o modelo de objetivos que visa apenas dar uma visão inicial dos problemas que o cliente irá resolver ao adquirir o sistema.

O Diagrama de Processos foi construído para demonstrar a seqüência das atividades de negocio do cliente. Seus processos atendem os objetivos levantados no modelo de objetivos.

Após estes diagramas foram levantados os principais requisitos do sistema, porém os requisitos não foram levantados de forma muito detalhada, pois o tempo gasto seria muito grande para um sistema que já está em uso.

Os diagramas da UML desenvolvidos foram o Diagrama de Casos de Uso e Diagrama de Atividades. O Diagrama de Atividades foi utilizado para descrever as regras de negócio do sistema. O Diagrama de Casos de Uso foi construído para representar a interação dos usuários com o sistema.

A engenharia reversa foi realizada seguindo alguns padrões que contribuíram com a qualidade deste trabalho.

Esta documentação foi feita visando as regras de negócios, o entendimento desta regras era a principal dificuldade na hora de realizar as manutenções do sistema.

#### 4.1 EXTENSÕES

Apesar da documentação ter sido concluída, apenas com sua utilização serão verificados os pontos onde a documentação deverá ser mais ou menos detalhada, além disto, pode se chegar a conclusão que algum diagrama não seja necessário.

A documentação das regras de negócio não foi feita neste trabalho. Esta documentação seria importante de ser construída. Em relação a novas implementações seria interessante como mencionado na conclusão a geração do Diagrama Entidade Relacionamento por módulos e não mais por rotinas.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANQUETIL, Nicolas. **Diagramas de casos de uso**, São Paulo, 2004. Disponível em: <http://www.ucb.br/ucbtic/mgcti/paginapessoalprof/Nicolas/Disciplinas/UML/node5.html>  
Acesso em: 1 mai. 2008.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistema UML**. 2. ed. Rio de Janeiro: Elsevier, 2007.

CIVIERO, Ewerton. **Manual do sistema jurídico**, 80 f, Manual do sistema, 2007.

HECKMANN, Jacques R. **Gerador de bases de conhecimento genexus a partir de código fonte da linguagem FoxPro**. 1995. 44f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

KOLLER, Karen F. **Um protótipo para geração do modelo E-R a partir do esquema gerado para o banco de dados ORACLE7**. 1996. 100f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

KREMMER, Fernando, **Desenvolvimento de projetos em UML**. 2007. Blumenau. Disponível em: [http://www.benner.com.br/wiki/index.php?title=Modelagem\\_de\\_atividades\\_de\\_neg%C3%B3cio](http://www.benner.com.br/wiki/index.php?title=Modelagem_de_atividades_de_neg%C3%B3cio) Acesso em: 1 mai. 2008.

MONTEIRO, Emiliano. **Projeto de sistemas e banco de dados**. 1. ed. Rio de Janeiro: Brasport, 2004.

PENDER, Tom. **UML a Bíblia**. 1. ed. Rio de Janeiro: Campus, 2004.

PERES, Darley R. **Padrões de processo de engenharia reversa baseado em transformações**. 1. ed. São Paulo: Alta Books, 2003.

PRESSMAN, Roger S. **Engenharia de software**. 5. ed. Rio de Janeiro: McGraw-Hill, 2002.

PFLEEGER, Shari L. **Engenharia de software**. 2. ed. São Paulo: Pearson Education, 2004.

PILONE, Dan. **UML 2**. 1. ed. Rio de Janeiro: Alta Books, 2006.

SCARTON, Evandro M. **Estudo comparativo da engenharia reversa de dados em ferramentas Case**. 1997. 90f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SOMMERVILLE, Ian. **Engenharia de software**. 1. ed. São Paulo: Pearson Education, 2003.

SMITCH, Carlos D. **Manual Builder**, 20 p, Manual da ferramenta Builder, 1997.

XAVIER, Jonas, **Modelagem de sistemas**. 2008. São Paulo. Disponível em: <<http://pt.wikipedia.org/wiki/Modelagem>> Acesso em: 1 mai. 2008.