



Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"

**LUIZ ANGELO FRANCISCATTI**

**SISTEMA PARA GERENCIAMENTO FINANCEIRO**

Assis

2010

LUIZ ANGELO FRANCISCATTI

## SISTEMA PARA GERENCIAMENTO FINANCEIRO

Trabalho de conclusão de  
Curso apresentado ao Instituto  
Municipal de Ensino Superior de  
Assis, como requisito do curso  
de Graduação.

Orientador: Luiz Carlos Begosso

Área de Concentração: Desenvolvimento de Sistemas

Assis  
2010

## FICHA CATALOGRÁFICA

FRANCISCATTI, Luiz Angelo

Sistema para Gerenciamento Financeiro / Luiz Angelo Franciscatti. Fundação Educacional do Município de Assis, 2010.

57p.

Orientador: Dr. Luiz Carlos Begosso.

Trabalho de Conclusão de Curso - Instituto Municipal de Ensino Superior de Assis - IMESA.

CDD: 001.61

Biblioteca da FEMA

# SISTEMA PARA GERENCIAMENTO FINANCERIO

LUIZ ANGELO FRANCISCATTI

Trabalho de conclusão de Curso  
apresentado ao Instituto Municipal  
de Ensino Superior de Assis,  
como requisito do curso de  
Graduação, analisado pela  
seguinte comissão examinadora.

Orientador: Luiz Carlos Begosso

Analisador: Luiz Ricardo Begosso

ASSIS

2010

## AGRADECIMENTOS

Primeiramente a Deus, por me ajudar a cumprir mais esta importante etapa da minha vida e por me encorajar-me a enfrentar os desafios encontrados ao longo do caminho.

Ao meu orientador Dr. Luiz Carlos Begosso, pelas suas orientações, ideias, opiniões e por sempre me incentivar durante este trabalho.

Aos familiares, meu pai Angelo e minha mãe Luzia meus irmãos, Amanda, Fernando e minha noiva Juliana por ter acreditado na minha capacidade e sempre tiveram me ajudando e apoiando em todos os momentos da minha vida e que confiaram e acreditaram no meu potencial.

Aos meus Amigos do curso TPD, Mayquel, Priscila, Jefferson, André, João, Nando, Moacir e todos que colaboraram ajudando trocando informações e amizade sincera e aos Amigos que conviveram comigo durante meu estágio no CEPEIN, Mariana, Hellen, Tamiris, Leonardo, Wilson e a todos que ajudaram nos momento de dificuldade e acreditaram que essa jornada é a caminhada para sucesso.

## RESUMO

Nesse trabalho será apresentada a análise e o projeto de um sistema para gerenciamento financeiro para entidade Associação Filantrópica Nosso Lar. O Software será desenvolvido com intuito de atender e suprir todas as necessidades da entidade, manter controle sobre os dados referentes a situações financeiras, armazenar histórico financeiro de cada projeto, além de controlar as consultas e emitir relatórios de contas a pagar e receber. As tecnologias utilizadas no desenvolvimento do sistema foram: linguagem de programação Java, banco de dados PostgreSQL e a ferramenta de desenvolvimento NetBeans.

Palavras - chaves: Java, NetBeans, PostgreSQL.

## ABSTRACT

In this work will be presents the analysis of a system for financial management for a entity. The software will be installed with the intention to attend and meet all the needs of the organization, keeping track of data on financial situations, storing financial history of each project, and track queries and reporting of accounts payable and receivable. Introducing the technologies used in developing the system, the Java programming language, PostgreSQL database and development tool NetBeans.

Key - words: Java, NetBeans, PostgreSQL.

## LISTA DE ILUSTRAÇÃO

Figura 01 - <i>Work Breakdown Structure</i> .....	14
Figura 02 - Sequenciamento de Atividades.....	15
Figura 03- Diagrama Use Case .....	22
Figura 04- Diagrama Use Case Informar <i>Login</i> .....	23
Figura 05- Diagrama Use Case Manter Usuário .....	24
Figura 06- Diagrama Use Case Manter Fontes de Recursos .....	25
Figura 07- Diagrama Use Case Manter Projeto .....	26
Figura 08- Diagrama Use Case Manter Serviço .....	27
Figura 09- Diagrama Use Case Movimentar Contas a Receber .....	28
Figura 10- Diagrama Use Case Movimentar Contas a Pagar .....	29
Figura 11- Diagrama Use Case Movimentação Financeira .....	30
Figura 12- Diagrama Use Case Emitir Relatório de Contas a Pagar.....	31
Figura 13- Diagrama Use Case Emitir Relatório de Contas a Receber.....	32
Figura 14- Diagrama Use Case Emitir Relatório do Projeto.....	33
Figura 15- Diagrama Use Case Emitir Relatório Mensal.....	34
Figura 16- Diagrama Use Case Emitir Relatório Geral.....	35
Figura 17- Diagrama de Classe .....	36
Figura 18- Diagrama de Sequência Cadastrar Usuário.....	37
Figura 19- Diagrama de Sequência Cadastrar Fontes de Recursos.....	38
Figura 20- Diagrama de Sequência Cadastrar Projeto.....	38
Figura 21- Diagrama de Sequência Cadastrar Serviço.....	39
Figura 22- Diagrama de Sequência Cadastrar Contas a Pagar.....	40
Figura 23- Diagrama de Sequência Cadastrar Contas a Receber.....	40
Figura 24 - Diagrama de Sequência Emitir Relatório Contas a Pagar .....	41
Figura 25 - Diagrama de Sequência Emitir Relatório Contas a Receber.....	42
Figura 26 - Diagrama de Sequência Emitir Relatório do Projeto .....	42
Figura 27 - Diagrama de Sequência Emitir Relatório do Projeto Geral.....	43
Figura 28 - Diagrama de Entidade e Relacionamento.....	44

## LISTA DE TABELAS

Tabela 01- Acompanhamento de Custos .....	18
Tabela 02 - Manter Usuário .....	24
Tabela 03 - Manter Fontes de Recursos.....	25
Tabela 04 - Manter Projeto.....	26
Tabela 05- Manter Serviço .....	27
Tabela 06- Movimentar Contas a Receber .....	28
Tabela 07- Movimentar Contas a Pagar .....	29
Tabela 08- Movimentação Financeira .....	30
Tabela 09- Emitir Relatório de Contas a Pagar.....	31
Tabela 10- Emitir Relatório de Contas a Receber.....	32
Tabela 11- Emitir Relatório do Projeto.....	33
Tabela 12- Case Emitir Relatório Mensal.....	34
Tabela 13- Emitir Relatório Geral.....	35

## Sumário

<b>1- INTRODUÇÃO</b> .....	<b>12</b>
1.1 OBJETIVO DO TRABALHO.....	13
1.2 PÚBLICO ALVO.....	13
1.3 JUSTIFICATIVA.....	13
<b>2 - METODOLOGIA DE DESENVOLVIMENTO</b> .....	<b>14</b>
2.1 METODOLOGIA UTILIZADAS.....	14
2.1.1 Java .....	14
2.1.2 PostgreSQL.....	15
2.1.3 Hibernate .....	16
2.1.4 IReport .....	16
2.1.5 NetBeans .....	16
2.1.6 Uml.....	17
2.1.8 Orientação à Objetos .....	17
<b>3 - ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA</b> .....	<b>18</b>
<b>4- WORK BREAKDOWN STRUCTURE (WBS)</b> .....	<b>19</b>
<b>5- SEQUENCIAMENTO DE ATIVIDADES</b> .....	<b>20</b>
<b>6- ORÇAMENTO</b> .....	<b>21</b>
<b>6.1 Pessoal</b> .....	<b>21</b>
<b>6.2 Equipamento</b> .....	<b>22</b>
<b>7 - ANÁLISE E ESPECIFICAÇÃO DO SISTEMA</b> .....	<b>24</b>
7.1 METODOLOGIA DE ANÁLISE .....	24
7.2 LEVANTAMENTO DE REQUISITOS.....	24
7.3 DECLARAÇÃO DOS OBJETIVOS.....	25
7.4 Caso Uso (UC).....	25
7.5 LISTA DE EVENTOS .....	26
7.6 Diagrama Use Case.....	27
7.7 Diagrama Use Case Informar Login.....	28
7.8 DIAGRAMA DE CLASSE.....	41

7.9 Diagrama de Sequência.....	42
7.10 Diagrama de Entidade e Relacionamento.....	49
<b>8- CRONOGRAMA .....</b>	<b>50</b>
<b>9- IMPLEMENTAÇÃO DA APLICAÇÃO.....</b>	<b>51</b>
<b>CONCLUSÃO .....</b>	<b>56</b>
<b>REFERÊNCIAS.....</b>	<b>57</b>

## 1- INTRODUÇÃO

Desde o início das atividades a Entidade Associação Filantrópica Nosso Lar executa todos os registros financeiros de forma manual. A utilização de um *software* traria segurança, agilidade e confiabilidade para a entidade.

Com o passar do tempo, novas tecnologias vem surgindo em conjunto com o crescimento da internet. Surge então a necessidade de ter informações em um menor tempo. Sendo assim, é de suma importância que a entidade procure informatizar seus processos para manter um maior controle sobre os fluxos de dados.

Um dos problemas enfrentados pela Entidade é a falta de um *software* que gerencie essas informações. Neste sentido, este Trabalho de Conclusão de Curso propõe o desenvolvimento do mesmo para seu gerenciamento.

O Sistema para Gerenciamento Financeiro tem como o principal objetivo administrar e manipular detalhadamente seus dados, a fim de controlar o setor financeiro da entidade e dos projetos envolvidos, além de contas a receber, a pagar, as fontes de recursos e histórico dos projetos.

## 1.1 OBJETIVO DO TRABALHO

O objetivo deste trabalho é desenvolver um software para facilitar o gerenciamento das informações relacionadas à entidade. O software será denominado de “Sistema para Gerenciamento Financeiro”, que controlará a contas a receber, a pagar, as fontes de recursos bem como o histórico dos projetos elaborados pela Entidade. Diminuir custos operacionais, evitar possíveis erros e falhas humanas, além de manter uma melhor organização das informações referentes a todos os dados em geral, permitindo assim fácil acesso às informações.

O objetivo geral do sistema é fornecer ao contador segurança e controle do processo financeiro com sucesso nos resultados.

O objetivo da informatização no setor será para o gerenciamento financeiro dos projetos envolvidos e para entidade, obtendo assim qualidade segurança e confiabilidade.

## 1.2 PÚBLICO ALVO

Empresa de pequeno e médio porte e entidades sociais.

## 1.3 JUSTIFICATIVA

O sistema surgiu devido ao fato de que os principais processos da entidade eram realizados manualmente.

As necessidades que levaram o surgimento do presente sistema foram: a falta do controle financeiro, a necessidade do controle mensal e anual dos relatórios,

referente às finanças e as contas a pagar e receber e as fontes de recursos de onde vem a verba relacionada a cada projeto.

## 2 - METODOLOGIA DE DESENVOLVIMENTO

Para o desenvolvimento do sistema será utilizada a linguagem de programação Java. O sistema será desenvolvido utilizando a ferramenta Netbeans e para a persistência com o banco de dados PostgreSQL, foi utilizado o *framework* Hibernate. A ferramenta que foi usada para a preparação da Análise *UML* e Caso de Uso.

### 2.1 METODOLOGIA UTILIZADAS

#### 2.1.1 Java

Em 1991, na *Sun Microsystems*, foi iniciado um projeto *Green Project*. Os mentores do projeto eram *Patrick Naughton*, *Mike Sheridan* e *James Gosling*. O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar à “próxima onda” do mundo digital. Eles acreditavam que em algum tempo haveria uma convergência dos computadores com os equipamentos eletrodomésticos comumente usados pelas pessoas no seu dia a dia. *James Gosling* a denominou de *Oak*, em homenagem a uma árvore que ele via da janela do seu escritório, porém mais tarde, teve que mudar de nome, pois já existia uma linguagem com esse nome. O nome Java, surgiu por acaso durante em uma visita da equipe da

Sun a uma cafeteira (Java era o nome de uma cidade de origem de um café importado).

Já foi anunciado em maio de 1995, pela empresa *Sun Microsystems*, no qual lançava naquele momento uma linguagem de programação que trabalhava em sites produzido na *World Wide Web*. Mas a tecnologia Java não funciona somente para aplicações *Web*, por ser uma linguagem totalmente orientada a objetos e independente de plataforma também é muito utilizada em desenvolvimento de *software*.

Desde seu lançamento, a plataforma Java foi adotada mais rapidamente do que qualquer outra linguagem de programação na história da computação. Segundo *Milkvicz* (2003), Java atingiu a marca de 4 milhões de desenvolvedores em todo mundo. Java continuou e continua crescendo e hoje é com certeza um padrão para o mercado oferecendo qualidade, desempenho e segurança ainda sem nenhum competidor a altura.

### 2.1.2 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto relacional(SGBDOR), ele foi o pioneiro em muitos conceitos objetos-relacionais que agora estão se tornando disponíveis em alguns bancos de dados comerciais.

Desenvolvido no Departamento de Ciência da Computação da Universidade da *Califórnia* em *Berkeley*. O projeto POSTGRES, liderado pelo Professor Michael Stonebraker, foi patrocinado pelas seguintes instituições: *Defense Advanced Research Projects Agency*(DARPA); *Army Research Office*(ARO), *National Science Foundation*(NSF); e ESL. Inc.

O PostgreSQL descende deste código original de *Berkeley*, possuindo o código fonte aberto. (Gonzaga, Jorge Luiz 2007).

### 2.1.3 Hibernate

O Hibernate é um projeto que procura ter uma completa solução para o problema de gerenciamento de dados persistentes em Java. O Hibernate é um framework que se relaciona com o banco de dados, conhecido como mapeamento objeto/relacional para Java.

O Hibernate se integra ao sistema comunicando com o banco de dados como se fosse diretamente feito por sua aplicação (Gonçalves, Edson 2008).

### 2.1.4 iReport

O iReport é um programa Open Source, capaz de criar relatórios para aplicações Java no formato da biblioteca JasperReports. Com ele é capaz de criar qualquer tipo de relatório simples de forma rápida (Gonçalves, Edson 2008).

### 2.1.5 NetBeans

NetBeans é um projeto Open Source que foi desenvolvido pela Sun Microsystems e é um ambiente de desenvolvimento de Software, escrito em Java, é usado por muitos desenvolvedores que utilizam esta linguagem. Esta ferramenta pode ser adquirida gratuitamente e auxiliam programadores escrever, compilar, depurar e implementar programas (Gonçalves, Edson 2008).

### 2.1.6 Uml

UML é chamada de linguagem de modelagem, não é um método. A maioria dos métodos consiste, pelo menos em princípio, de uma linguagem de modelagem e de um processo. A linguagem de modelagem é a notação (principalmente gráfica) utilizada por métodos para expressar projetos. O processo é a sugestão de quais passos a serem seguidos na elaboração de projeto.

### 2.1.7 Jude

*JUDE – Java and UML Developers' Environment - JUDE* é uma *IDE* para Modelagem de Dados (*UML*) criada com Java e de uso fácil e intuitivo. Com a *IDE JUDE* é possível realizar uma modelagem de dados complexa, apresenta os dados para o usuário de forma clara e ainda possui a vantagem de seu *layout* ser bem intuitivo. A *IDE* possui à sua esquerda uma Árvore de Estrutura com todos os Dados à disposição do usuário para se criar diagramas, mapas entre outros.

### 2.1.8 Orientação à Objetos

Projeto Orientado a objetos é uma técnica de programação que concentra-se nos dados e nas interfaces com esse objeto. Facilita a comunicação do profissional modelador e do usuário da área alvo, na medida em que a correlação da simbologia e conceitos abstratos do mundo real e da ferramenta de modelagem (conceitos, terminologia, símbolos, grafismo e estratégias) fosse a mais óbvia, natural e exata possível.

### **3 - ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA**

- Levantamento de Requisito
  
- Análise de Requisito
  
- Gerenciamento de Requisito
  
- Análise de Requisito
  
- Declaração de Objetivos
  
- Diagrama de Casos de Uso
  
- Diagrama de Classe
  
- Diagrama de Sequência
  
- Diagrama de Entidade e Relacionamento
  
- Implementação
  
- Teste
  
- Instalação

#### 4- WORK BREAKDOWN STRUCTURE (WBS)

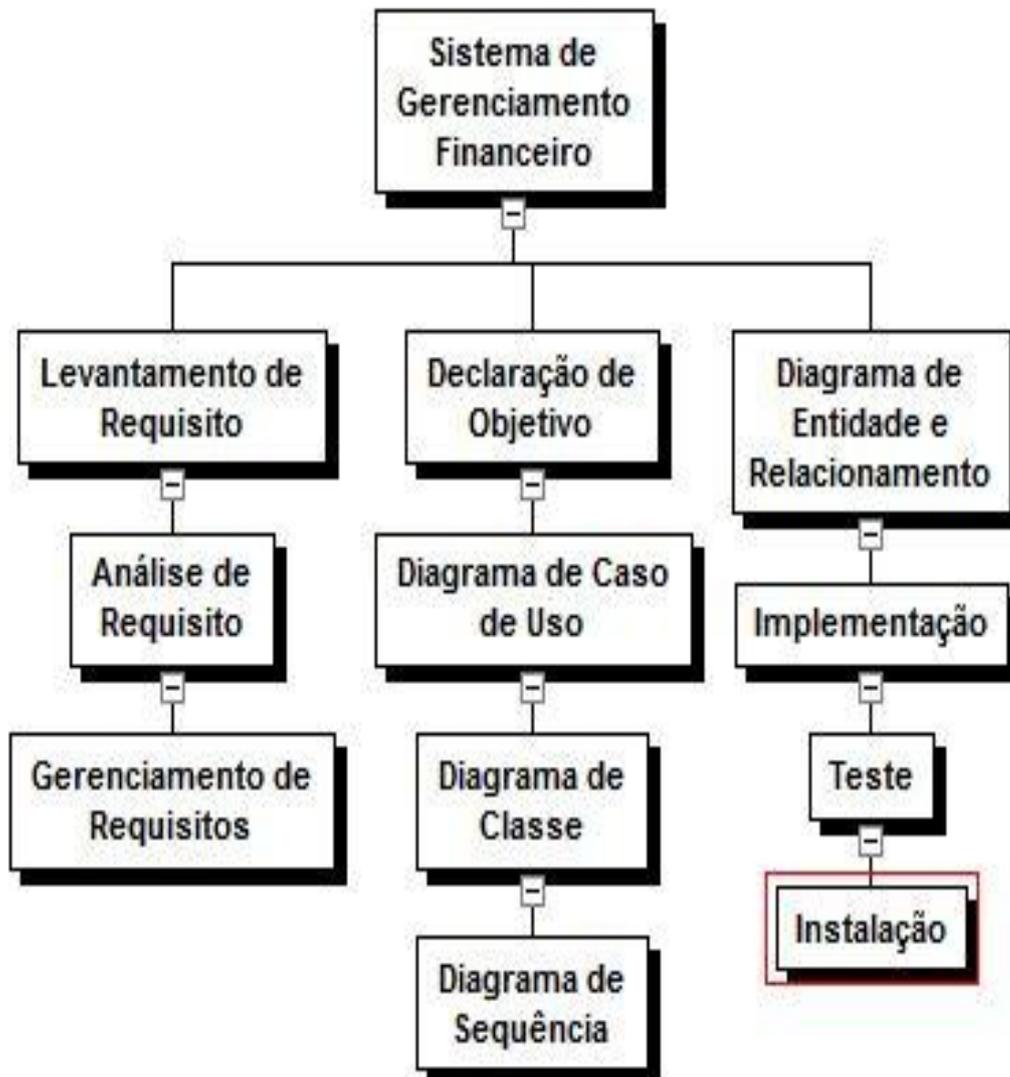


Figura 01 – Work Breakdown Structure

## 5- SEQUENCIAMENTO DE ATIVIDADES

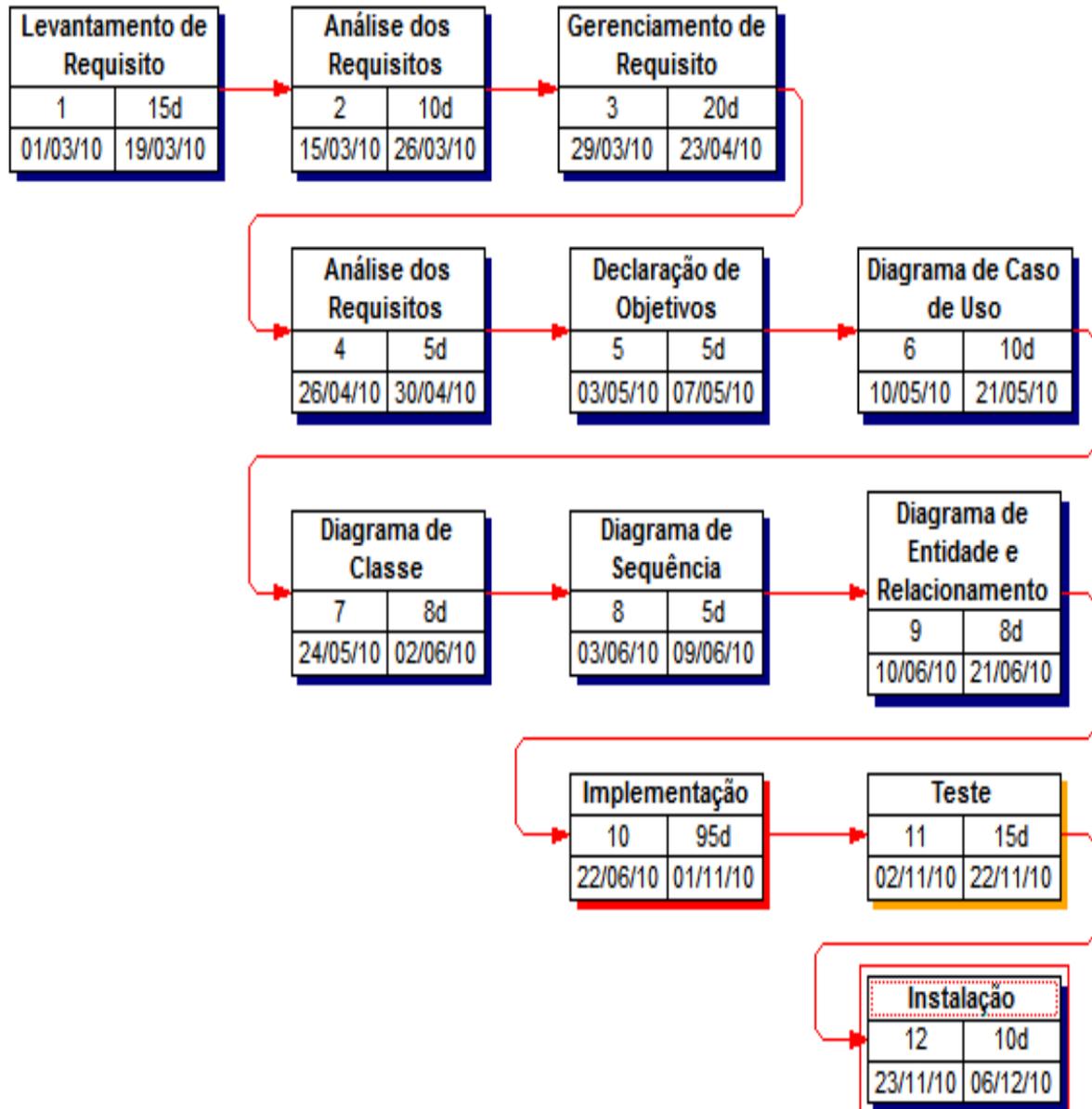


Figura 02 – Sequenciamento de Atividades

## 6- ORÇAMENTO

### Recursos Necessários ao Desenvolvimento

Serão necessários para o desenvolvimento do Projeto, os recursos descritos abaixo:

01 Analista de Sistemas

01 Programador

01 Microcomputador

01 Impressora Jato de Tinta

**Orçamento do Projeto:** Estimativa de custos para as atividades e Estimativa de custos para os recursos.

#### 6.1 Pessoal

<b>Analista</b>	<b>Qtde Horas</b> Somar as horas relativas ao trabalho do analista	<b>Custo/hora (R\$)</b>	<b>Total (R\$)</b>
Luiz Angelo	52	30,00	1.560,00
<b>Custo Analista</b>			<b>1.560,00</b>

<b>Programador</b>	<b>Qtde Horas</b> Somar as horas relativas ao trabalho do programador	<b>Custo/hora (R\$)</b>	<b>Total (R\$)</b>
Luiz Angelo	80	20,00	1.600,00

<b>Total Custo Pessoal</b>	<b>3.160,00</b>
----------------------------	-----------------

## 6.2 Equipamento

- **01 computadores**

- Valor unitário = R\$2.000,00
- Dias (de uso) = 26 dias (considerar: 78h / 03 de trabalho/dia = 26)
- Depreciação = R\$2.000,00 / 12meses (01anos. Tempo de depreciação) = R\$41,67/mês
- Custo dia = R\$41,67 / 26 (dias) = R\$1,60 (ao dia)
- Custo de cada computador = R\$1,60 \* 78 (total de horas do projeto) = R\$124,80.

**Custo de 01 computador \* R\$124,80**

- **01 impressora**

- Valor = R\$400,00
- Dias (de uso) = 26 dias
- Depreciação = R\$400,00 / 12 = R\$33,33
- Custo dia = R\$33,33 / 26 = R\$1,28

**Custo impressora = R\$1,28 \* 78h = R\$99,84**

**Custo Total Equipamento = R\$124,80 + R\$99,84 = R\$224,64**

**Custo Total do Projeto = R\$3.160,00 + R\$224,64 = R\$3.384,64**

### Acompanhamento de custos

<b>Atividades</b>	<b>Data (início)</b>	<b>Data (fim)</b>	<b>Previsto (dias)</b>	<b>Realizado (dias)</b>	<b>Saldo</b>
Levantamento de Requisito	04/03	05/03	02	02	00
Análise de Requisito	08/03	10/03	03	03	00
Gerenciamento de Requisito	22/03	26/03	05	06	-01
Análise de Requisito	26/03	28/03	03	03	00
Declaração de Requisito	05/04	07/04	03	03	00
Diagrama de Caso de Uso	12/04	16/04	05	08	-03
Diagrama Classe	30/05	06/06	07	06	+01
Diagrama Sequência	01/06	09/06	08	08	00
Diagrama de Entidade Relacionamento	09/06	14/06	05	05	00
Implementação	23/07	23/11	95	105	-15
Teste					
Instalação					
<b>Total</b>			21	25	-18

**Tabela 01 – Acompanhamento de Custos**

## **7 - ANÁLISE E ESPECIFICAÇÃO DO SISTEMA**

Este capítulo contém informações das ferramentas utilizadas para a modelagem do sistema proposto e a modelagem do sistema que será desenvolvido.

### **7.1 METODOLOGIA DE ANÁLISE**

Para o auxílio na modelagem do sistema, foi utilizada a metodologia de Análise Orientada a Objeto, juntos com as técnicas de UML (Linguagem de Modelagem Unificada). Depois de modelado, o sistema será desenvolvido utilizando a ferramenta "Netbeans IDE" (Gonçalves, Edson 2008), com base na linguagem Java. Para o armazenamento de dados, será utilizado o banco de dados PostgreSQL, atendendo perfeitamente a implementação do sistema. Para o gerenciamento de relatórios será utilizada a ferramenta "iReports"(Gonçalves, Edson 2008).

### **7.2 LEVANTAMENTO DE REQUISITOS**

Os requisitos foram levantados através de entrevistas com Contador Sidinei Frederico Diniz e a Assistente Administrativa Letícia, afim de, suprir as necessidades da entidade. Durante as entrevistas foram elencados alguns requisitos, Cadastros das Fontes de Recursos e da movimentação financeira de cada projeto, controlar contas a pagar e receber e relatórios gerais.

### 7.3 DECLARAÇÃO DOS OBJETIVOS

Este trabalho tem por objetivo o desenvolvimento de um sistema que gerencie as movimentações financeiras relacionadas à entidade. O sistema utilizará cadastros, movimentações e relatórios para informar a assistente administrativa sobre as operações.

Além de informatizar à entidade, o sistema diminuirá os custos operacionais e evitará possíveis erros e falhas humanas, além de manter melhor a movimentação financeira a todos os dados da entidade em geral, permitir o fácil acesso as finanças da entidade.

### 7.4 Caso Uso (UC)

O Diagrama de Caso de Uso tem como elemento básico no desenvolvimento de *software*, Jacobson (1994) também introduziu um diagrama para a visualização de casos de uso, o qual não faz parte da UML. É importante lembrar que casos de uso representam uma visão externa do sistema.

## 7.5 LISTA DE EVENTOS

Para o desenvolvimento do sistema foram seleccionadas as seguintes funcionalidades, destacando o que foi desenvolvido no 1º módulo.

- 1 Manter Usuário
- 2 Manter Fontes de Recursos
- 3 Manter Projeto
- 4 Manter Serviço
- 5 Movimentar Contas a Receber
- 6 Movimentar Contas a Pagar
- 7 Movimentação Financeira
- 8 Gerar Relatório de Contas a Pagar
- 9 Gerar Relatório de Contas a Receber
- 10 Gerar Relatório do Projeto
- 11 Gerar Relatório Mensal
- 12 Gerar Relatório Geral

## 7.6 Diagrama Use Case

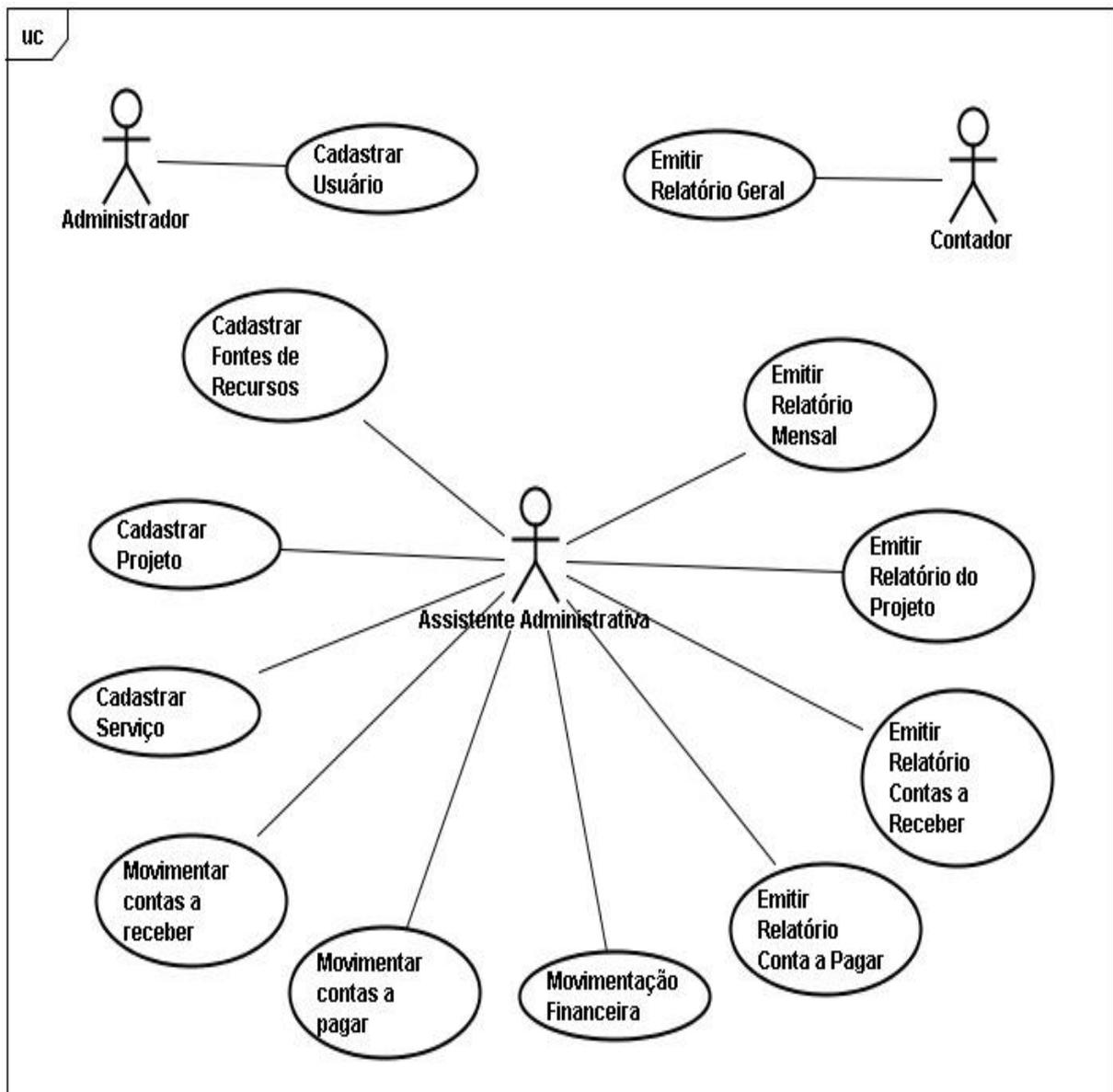


Figura 03- Diagrama de Use Case

## 7.7 Diagrama Use Case Informar Login

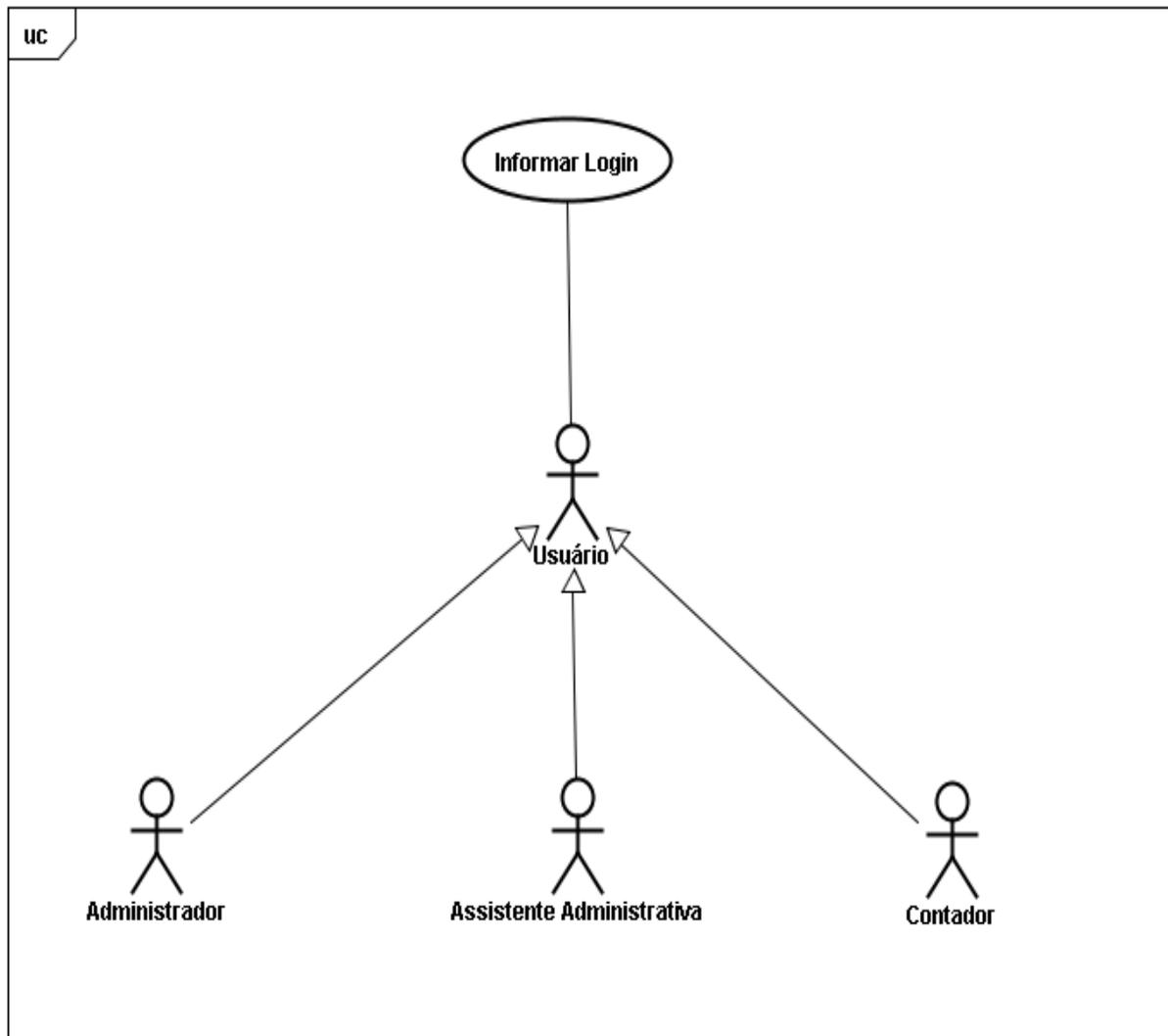
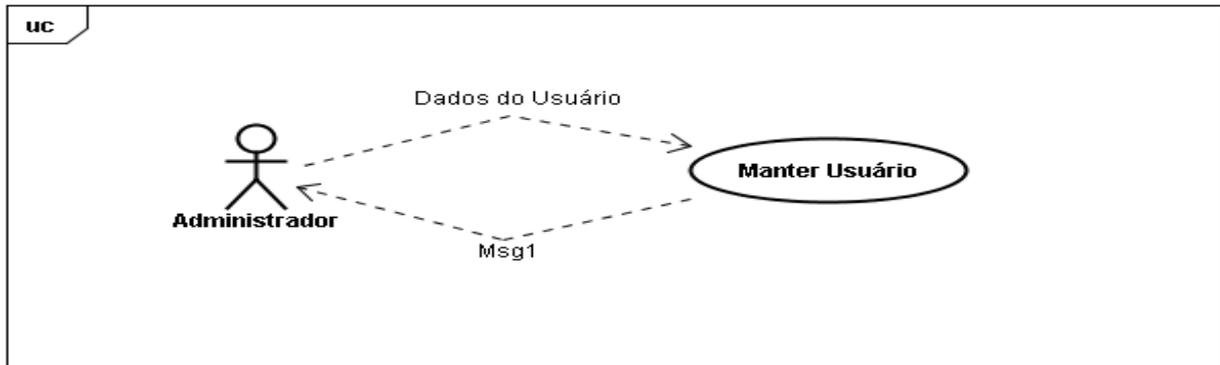


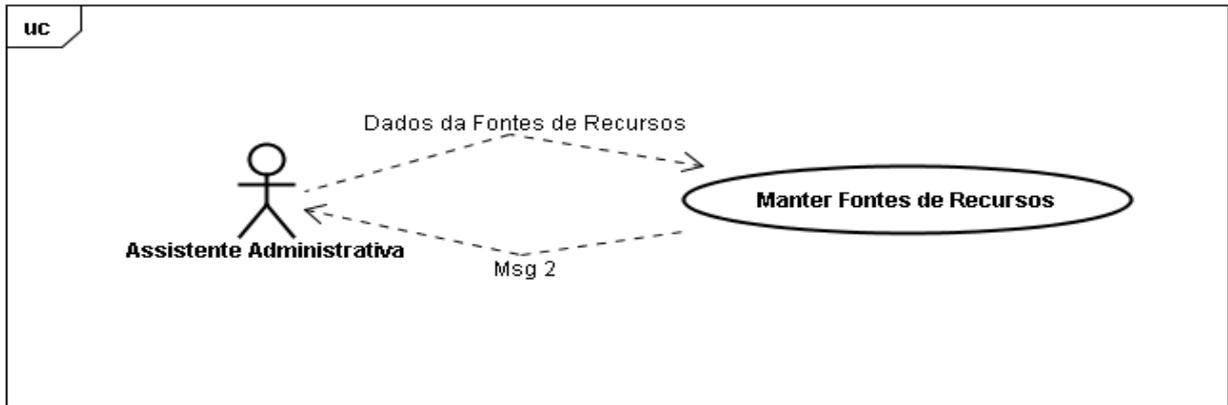
Figura 04- Diagrama de Use Case Informar Login



**Figura 05- Diagrama Use Case Manter Usuário**

<b>Nome da Use Case (01)</b>	Manter Usuário
<b>Ator(es)</b>	Administrador
<b>Pré-Condição</b>	O administrador deverá estar logado no sistema.
<b>Cenário Principal</b>	<p>1- O sistema solicita os dados necessários para o cadastro do usuário.</p> <p>2- O administrador informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O administrador clica em cadastrar.</p> <p>4- O sistema emite a mensagem “Usuário cadastrado com sucesso”.</p> <p>5- O sistema cadastrar o usuário.</p>
<b>Cenário Alternativo</b>	O administrador poderá cancelar o processo durante o cadastro.
<b>Casos de Teste</b>	<p>3.1- O sistema não confirma o cadastro e emite uma mensagem de erro na tela.</p> <p>3.2- O sistema cancela a operação.</p>

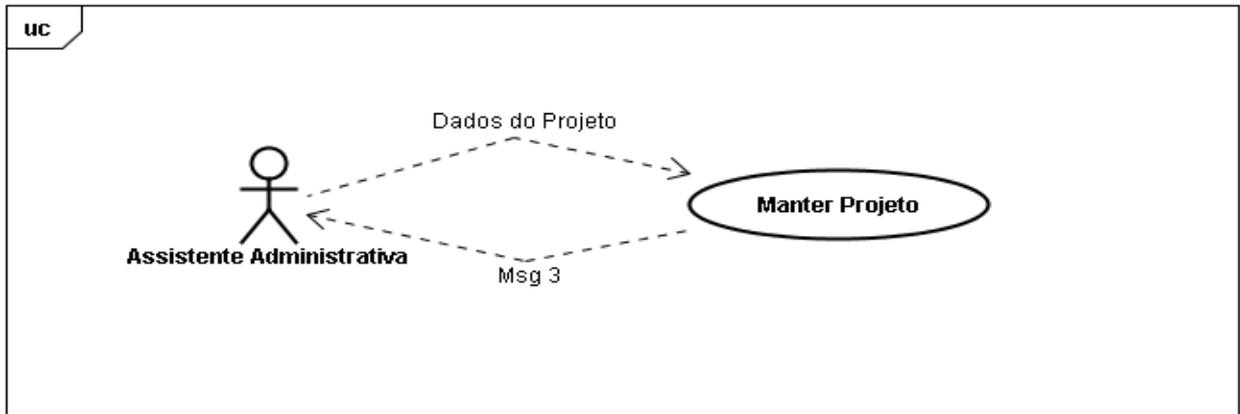
**Tabela 02 – Manter Usuário**



**Figura 06- Diagrama Use Case Manter Fontes de Recursos**

<b>Nome da Use Case (02)</b>	Manter Fontes de Recursos
<b>Ator(es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Cenário Principal</b>	<p>1- O sistema solicita os dados necessários para o cadastro de fontes de recursos.</p> <p>2- A assistente administrativa informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- A assistente administrativa clica em cadastrar.</p> <p>4- O sistema emite a mensagem “Fontes de Recurso cadastrado com sucesso”.</p> <p>5- O sistema cadastra a Fontes de Recursos.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante o cadastro.
<b>Casos de Teste</b>	<p>3.1- O sistema não confirma o cadastro e emite uma mensagem de erro na tela.</p> <p>3.2- O sistema cancela a operação.</p>

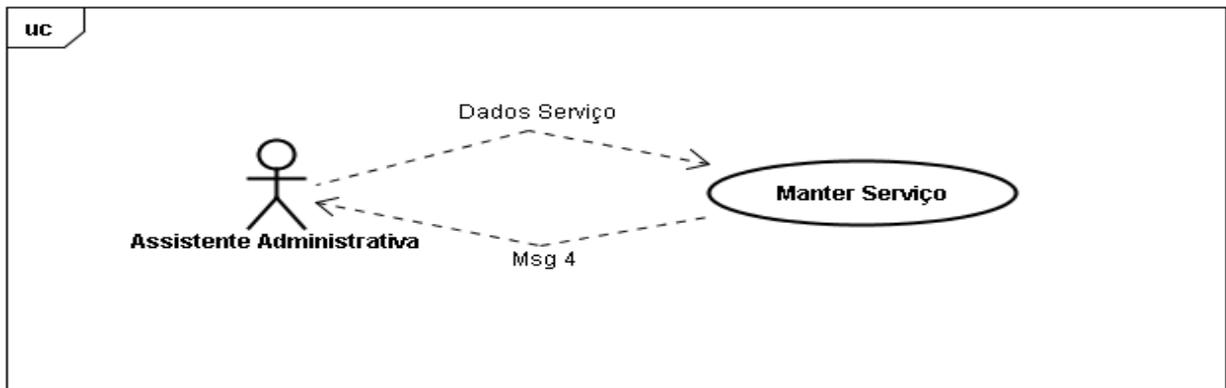
**Tabela 03 – Manter Fontes de Recursos**



**Figura 07- Diagrama Use Case Manter Projeto**

<b>Nome da Use Case (03)</b>	Manter Projeto
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Cenário Principal</b>	<p>1- O sistema solicita os dados necessários para o cadastro do projeto.</p> <p>2- A assistente administrativa informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- A assistente administrativa clica em cadastrar.</p> <p>4- O sistema emite a mensagem “Projeto cadastrado com sucesso”.</p> <p>5- O sistema cadastra o Projeto.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante o cadastro.
<b>Casos de Teste</b>	<p>3.1- O sistema não confirma o cadastro e emite uma mensagem de erro na tela.</p> <p>3.2- O sistema cancela a operação.</p>

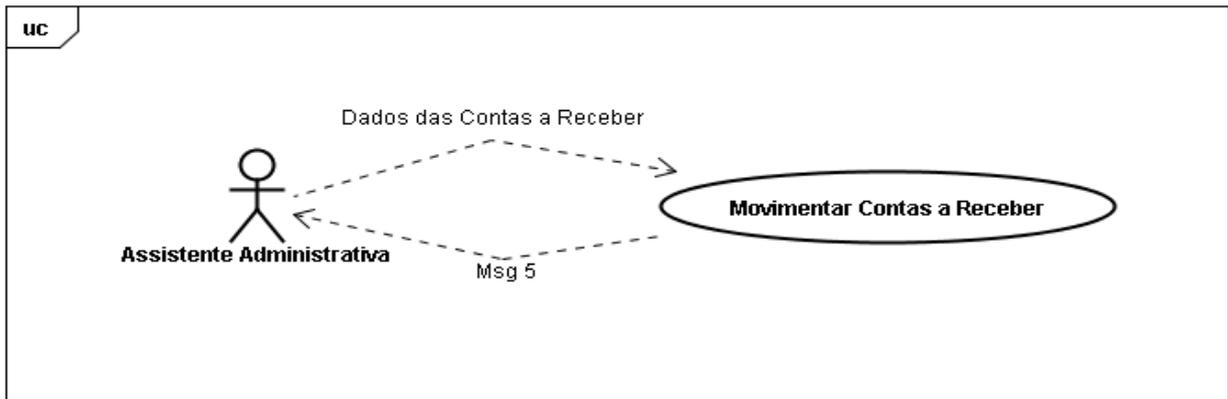
**Tabela 04 – Manter Projeto**



**Figura 08- Diagrama Use Case Manter Serviço**

<b>Nome da Use Case (04)</b>	Manter Serviço
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Cenário Principal</b>	<p>1- O sistema solicita os dados necessários para o cadastro do serviço.</p> <p>2- A assistente administrativa informa os dados de acordo com os campos serem preenchidos.</p> <p>3- A assistente administrativa clica em cadastrar.</p> <p>4- O sistema emite a mensagem “Serviço cadastrado com sucesso”.</p> <p>5- O sistema cadastra o Serviço.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante o cadastro.
<b>Casos de Teste</b>	<p>3.1- O sistema não confirma o cadastro e emite uma mensagem de erro na tela.</p> <p>3.2- O sistema cancela a operação.</p>

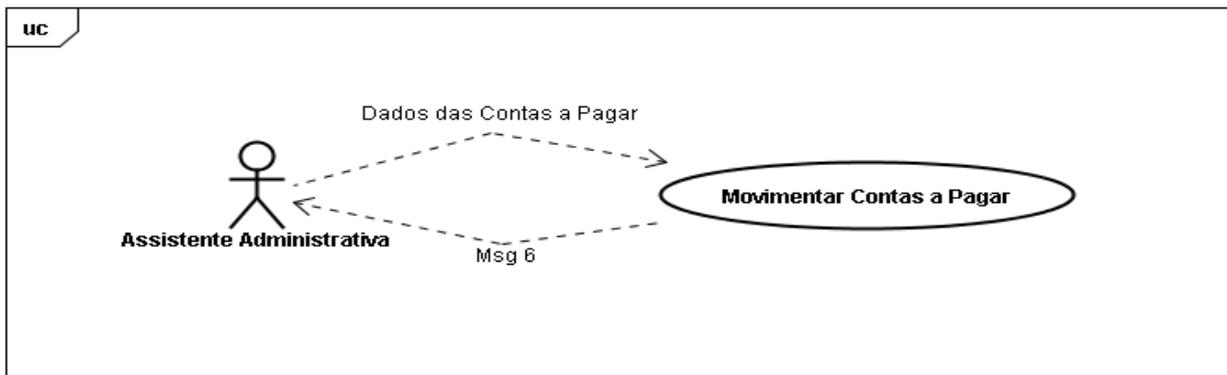
**Tabela 05 – Manter Serviço**



**Figura 09 – Diagrama Use Case Movimentar Contas a Receber**

<b>Nome da Use Case (05)</b>	Movimentar Contas a Receber
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Cenário Principal</b>	<p>1- O sistema solicita os dados necessários para movimentar contas a receber.</p> <p>2- A assistente administrativa informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- A assistente administrativa clica “Salvar”.</p> <p>4- O sistema emite a mensagem “Operação realizada com sucesso”.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante a movimentação.
<b>Casos de Teste</b>	<p>3.1- O sistema não confirma a movimentação e emite uma mensagem de erro na tela.</p> <p>3.2- O sistema cancela a operação.</p>

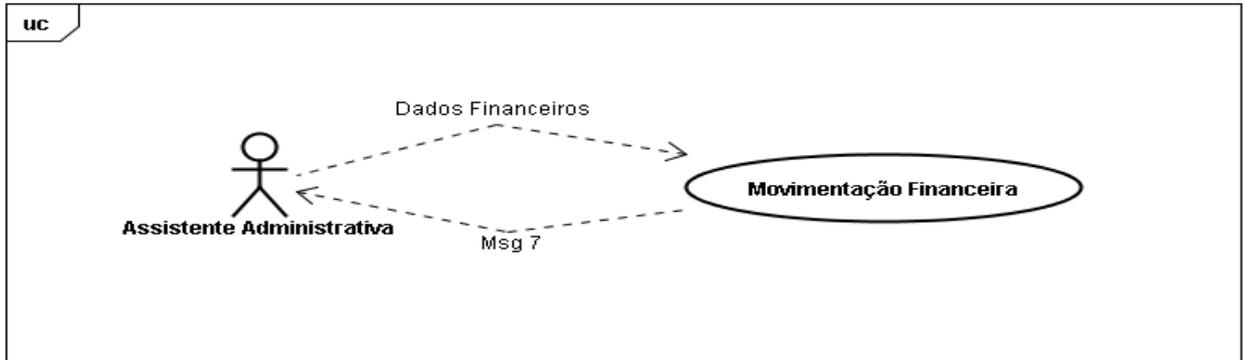
**Tabela 06 – Movimentar Contas a Receber**



**Figura 10- Diagrama Use Case Movimentar Contas a Pagar**

<b>Nome da Use Case (06)</b>	Movimentar Contas a Pagar
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condção</b>	A assistente administrativa deverá estar logado no sistema.
<b>Cenário Principal</b>	<p>1- O sistema solicita os dados necessários para movimentar contas a pagar.</p> <p>2- A assistente administrativa informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- A assistente administrativa clica em “Salvar”.</p> <p>4- O sistema emite a mensagem “Operação realizada com sucesso”.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante a movimentação.
<b>Casos de Teste</b>	<p>3.1- O sistema não confirma a movimentação e emite uma mensagem de erro na tela.</p> <p>3.2- O sistema cancela a operação.</p>

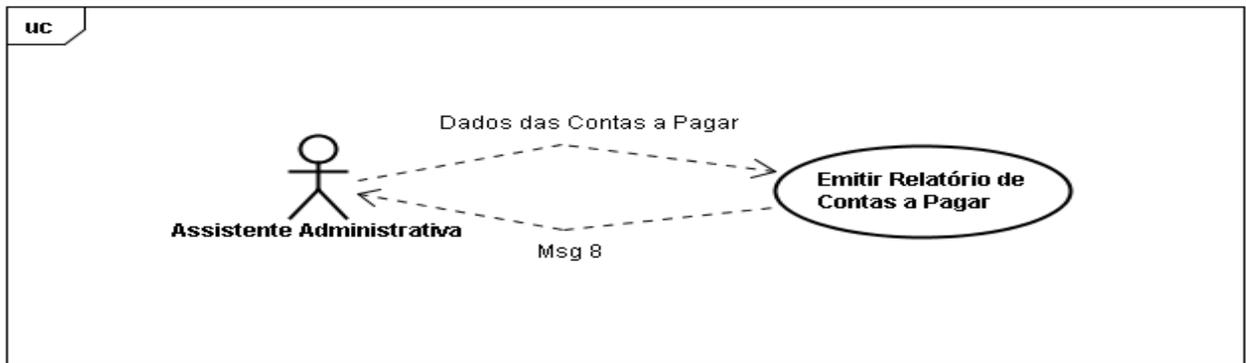
**Tabela 07 – Movimentar Contas a Pagar**



**Figura 11- Diagrama Use Case Movimentação Financeira**

<b>Nome da Use Case (07)</b>	Movimentação Financeira
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Cenário Principal</b>	<p>1- O sistema solicita os dados necessários para fazer a movimentação.</p> <p>2- A assistente administrativa informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- A assistente administrativa clica em “Salvar”.</p> <p>4- O sistema emite a mensagem “Operação realizada com sucesso”.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante a movimentação.
<b>Casos de Teste</b>	<p>3.1- O sistema não confirma a movimentação e emite uma mensagem de erro na tela.</p> <p>3.2- O sistema cancela a operação.</p>

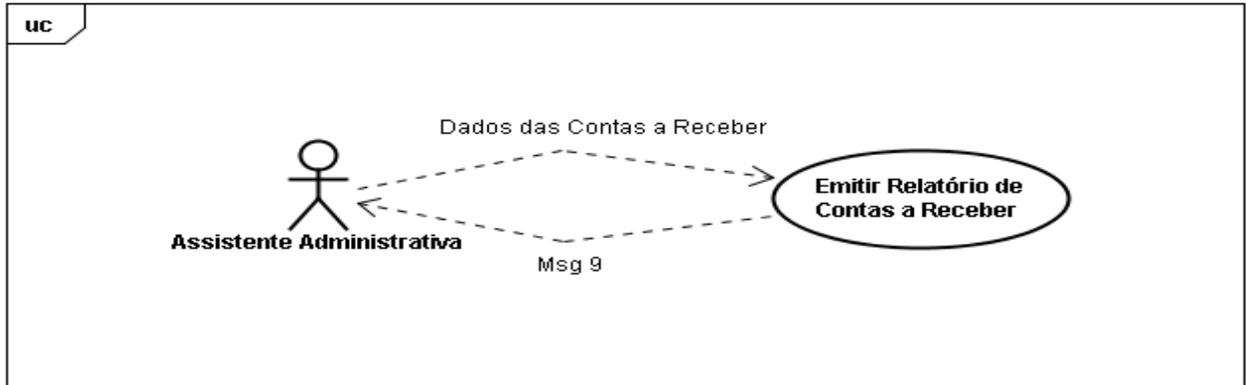
**Tabela 08 – Movimentação Financeira**



**Figura 12- Diagrama Use Case Emitir Relatório de Contas a Pagar**

<b>Nome da Use Case (08)</b>	Emitir Relatório de Contas a Pagar
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Descrição</b>	Os relatórios deverão ser impressos conforme forem necessitados.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1- O sistema disponibiliza informações necessárias a assistente administrativa.</li> <li>2- A assistente administrativa clica em visualizar relatório.</li> <li>3- A assistente administrativa clica em Imprimir.</li> <li>4- O sistema imprime o relatório com sucesso.</li> </ol>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante a visualização.
<b>Casos de Teste</b>	4.1- O sistema cancela a operação.

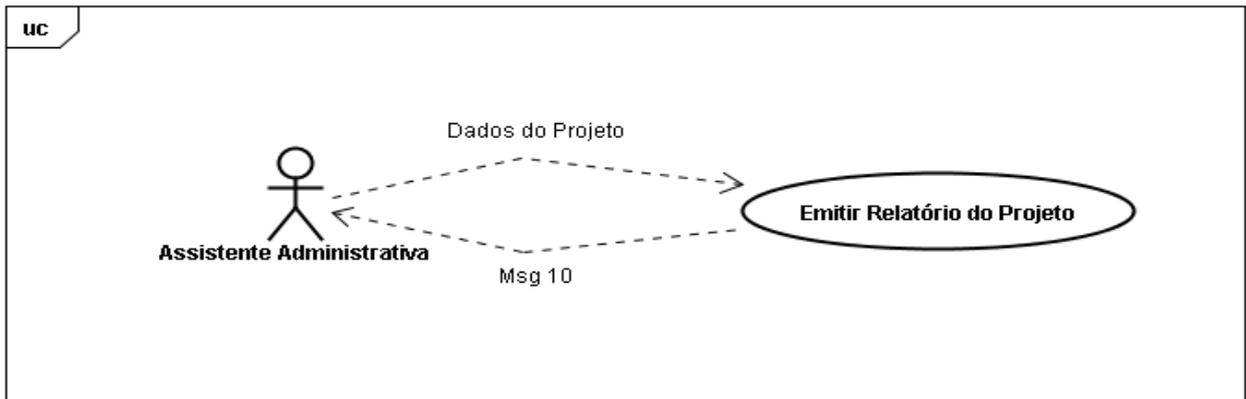
**Tabela 09 – Emitir Relatório de Contas a Pagar**



**Figura 13- Diagrama Use Case Emitir Relatório de Contas a Receber**

<b>Nome da Use Case (09)</b>	Emitir Relatório de Contas a Receber
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Descrição</b>	Os relatórios deverão ser impressos conforme forem necessitados.
<b>Cenário Principal</b>	<p>1- O sistema disponibiliza informações necessárias a assistente administrativa.</p> <p>2- A assistente administrativa clica em visualizar relatório.</p> <p>3- A assistente administrativa clica em Imprimir.</p> <p>4- O sistema imprime o relatório com sucesso.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante a visualização.
<b>Casos de Teste</b>	4.1- O sistema cancela a operação.

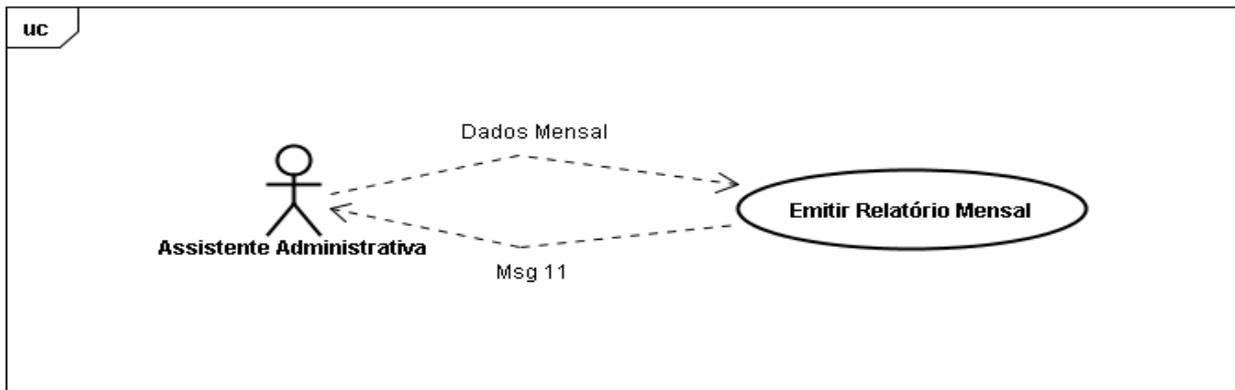
**Tabela 10 – Emitir Relatório de Contas a Receber**



**Figura 14- Diagrama Use Case Emitir Relatório do Projeto**

<b>Nome da Use Case (10)</b>	Emitir Relatório do Projeto
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Descrição</b>	Os relatórios deverão ser impressos conforme forem necessitados.
<b>Cenário Principal</b>	<p>1- O sistema disponibiliza informações necessárias a assistente administrativa.</p> <p>2- A assistente administrativa clica em visualizar relatório.</p> <p>3- A assistente administrativa clica em Imprimir.</p> <p>4- O sistema imprime o relatório com sucesso.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante a visualização.
<b>Casos de Teste</b>	4.1- O sistema cancela a operação.

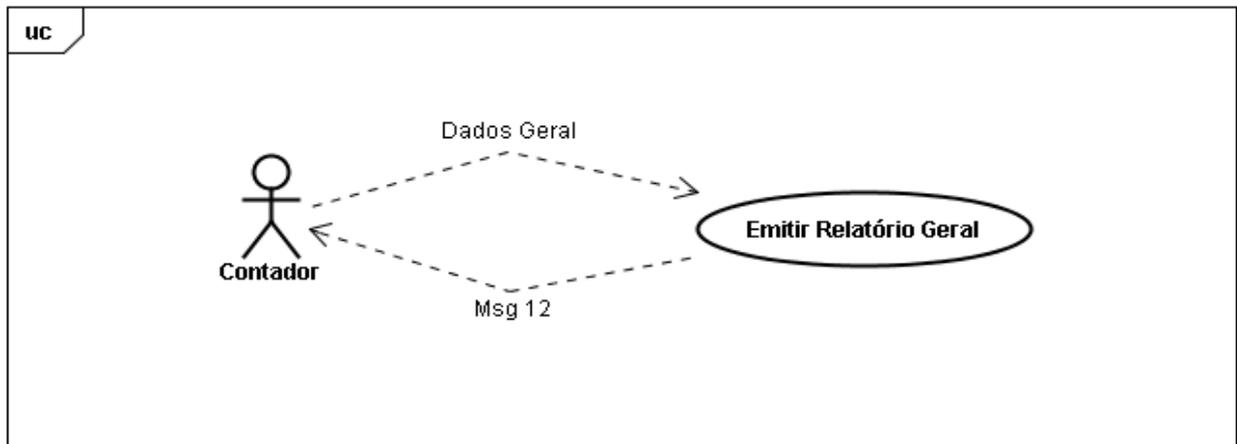
**Tabela 11 – Emitir Relatório do Projeto**



**Figura 15- Diagrama Use Case Emitir Relatório Mensal**

<b>Nome da Use Case (11)</b>	Emitir Relatório Mensal
<b>Ator (es)</b>	Assistente Administrativa
<b>Pré-Condição</b>	A assistente administrativa deverá estar logado no sistema.
<b>Descrição</b>	Os relatórios deverão ser impressos conforme forem necessitados.
<b>Cenário Principal</b>	<p>1- O sistema disponibiliza informações necessárias a assistente administrativa.</p> <p>2- A assistente administrativa clica em visualizar relatório.</p> <p>3- A assistente administrativa clica em Imprimir.</p> <p>4- O sistema imprime o relatório com sucesso.</p>
<b>Cenário Alternativo</b>	A assistente administrativa poderá cancelar o processo durante a visualização.
<b>Casos de Teste</b>	4.1- O sistema cancela a operação.

**Tabela 12 – Diagrama Use Case Emitir Relatório Mensal**



**Figura 16- Diagrama Use Case Emitir Relatório Geral**

<b>Nome da Use Case (12)</b>	Emitir Relatório Geral
<b>Ator(es)</b>	Contador
<b>Pré-Condição</b>	O contador deverá estar logado no sistema.
<b>Descrição</b>	Os relatórios deverão ser impressos conforme forem necessitados.
<b>Cenário Principal</b>	<p>1- O sistema disponibiliza informações necessárias ao contador.</p> <p>2- O contador clica em visualizar relatório.</p> <p>3- O contador clica em Imprimir.</p> <p>4- O sistema imprime o relatório com sucesso.</p>
<b>Cenário Alternativo</b>	O contador poderá cancelar o processo durante a visualização.
<b>Casos de Teste</b>	4.1- O sistema cancela a operação.

**Tabela 13 – Emitir Relatório Geral**

## 7.8 DIAGRAMA DE CLASSE

Um modelo de classe descreve a estrutura estática dos objetos de um sistema, suas relações com outros objetos, seus atributos e suas operações. A ilustração abaixo da (Figura 17) mostra o diagrama de classe do sistema.

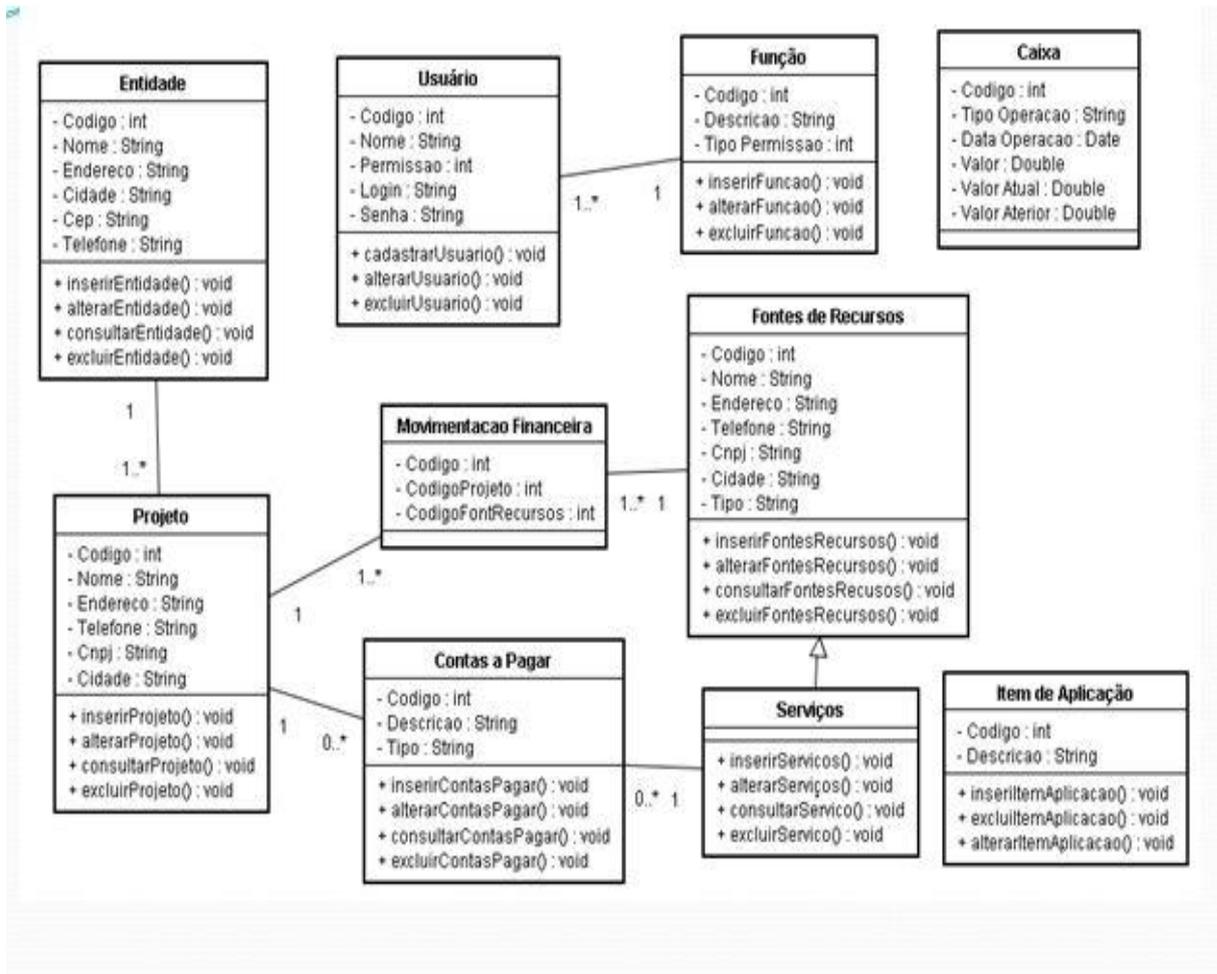
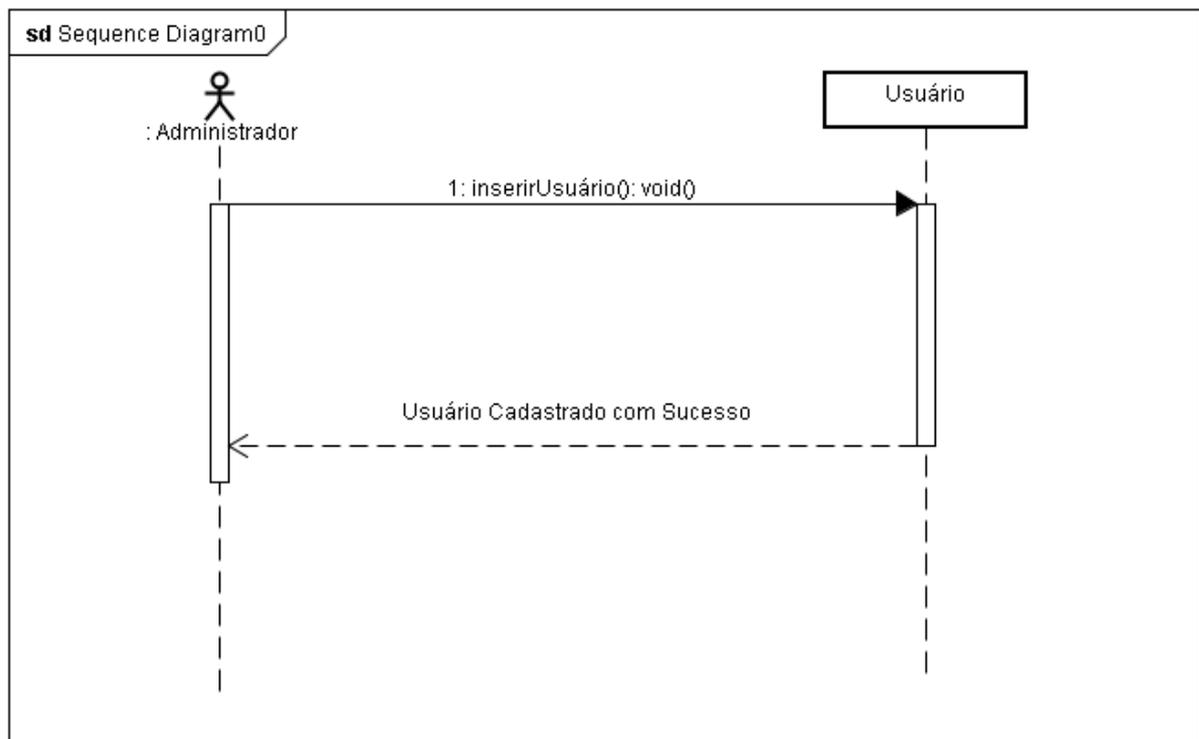


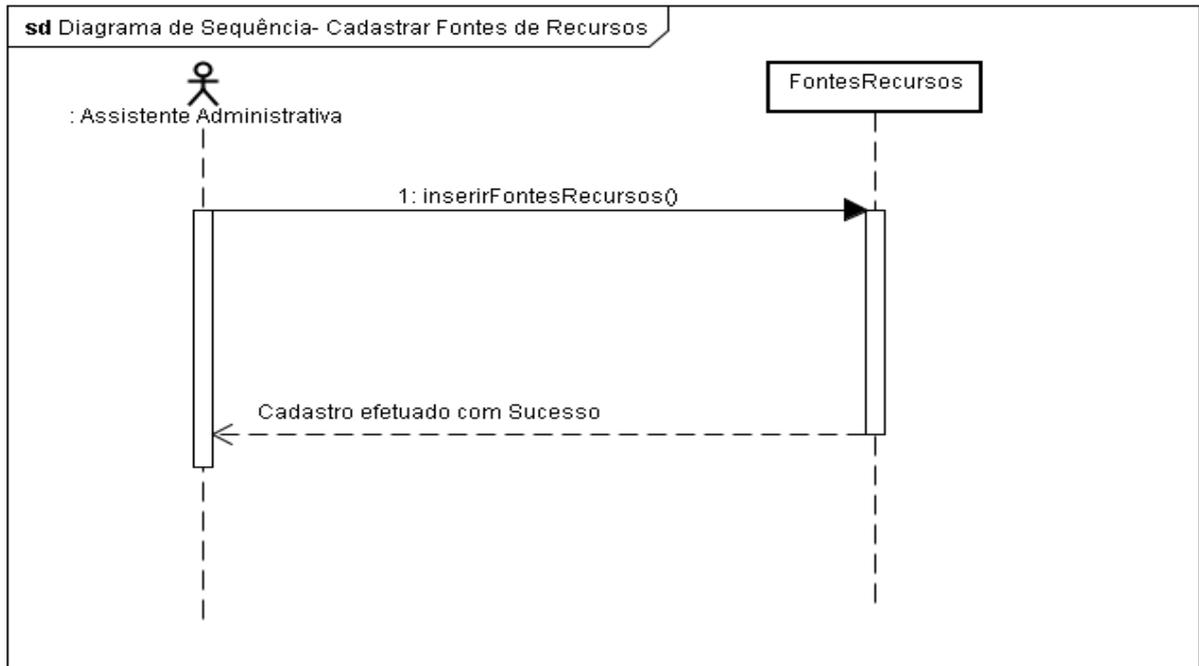
Figura 17- Diagrama de Classe

## 7.9 Diagrama de Sequência

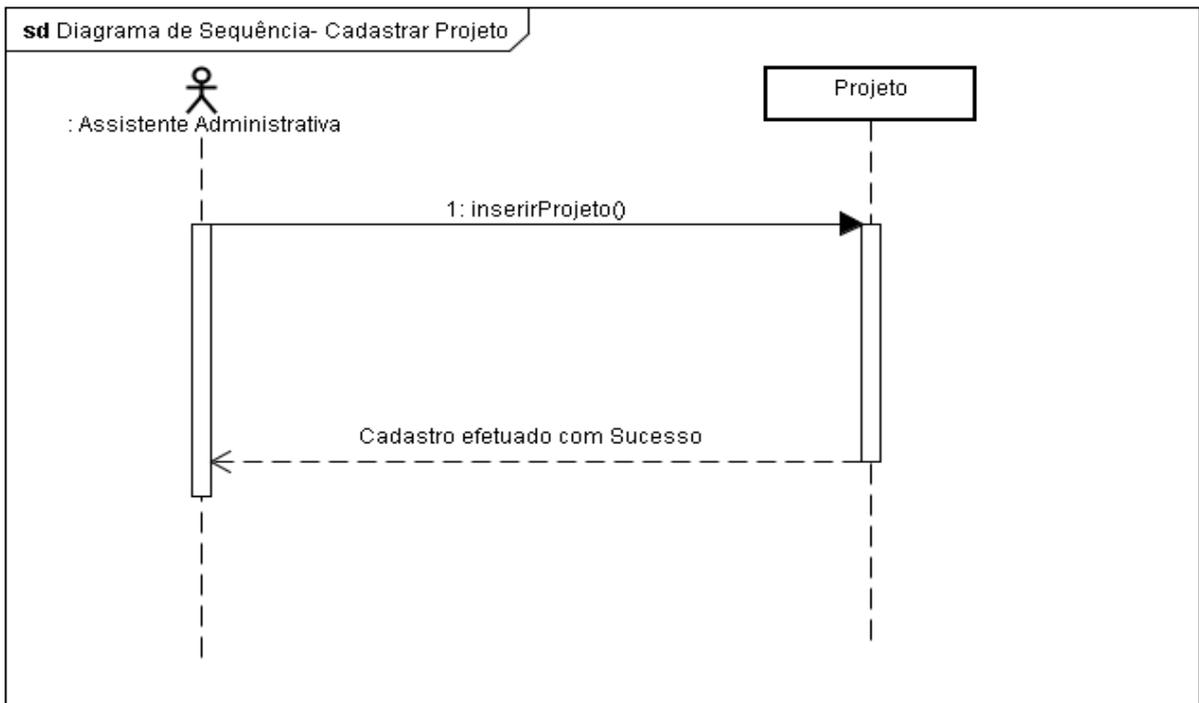
Um diagrama de sequência mostra objetos reais e interações entre objetos no sentido horizontal, e sequência no sentido vertical de um sistema, com seus atores para realizar todo um caso de uso ou uma parte dele. Essas interações são modeladas como intercâmbio de mensagens. Esses intercâmbios resultarão em um comportamento desejado. (Lee, Tepefenhart, 2001).



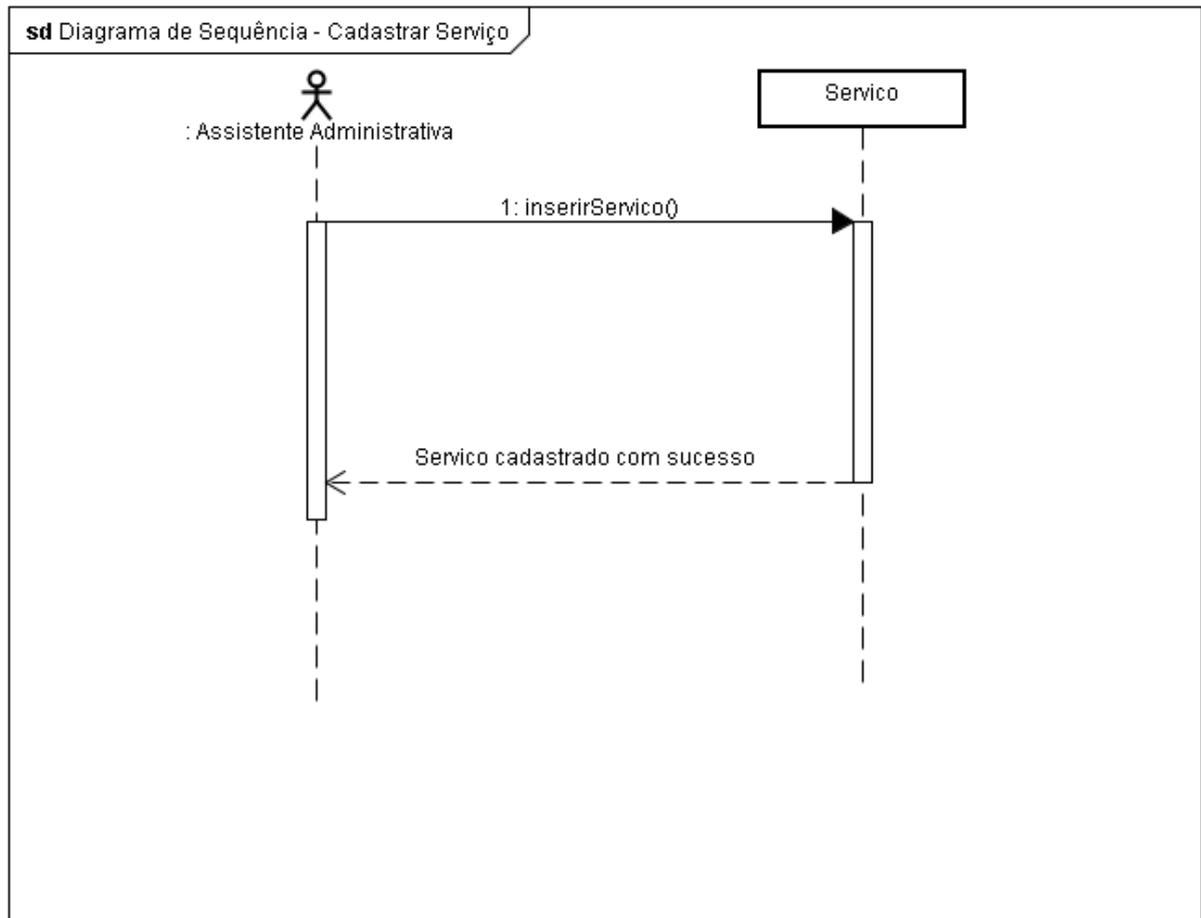
**Figura 18- Diagrama de Sequência Cadastrar Usuário**



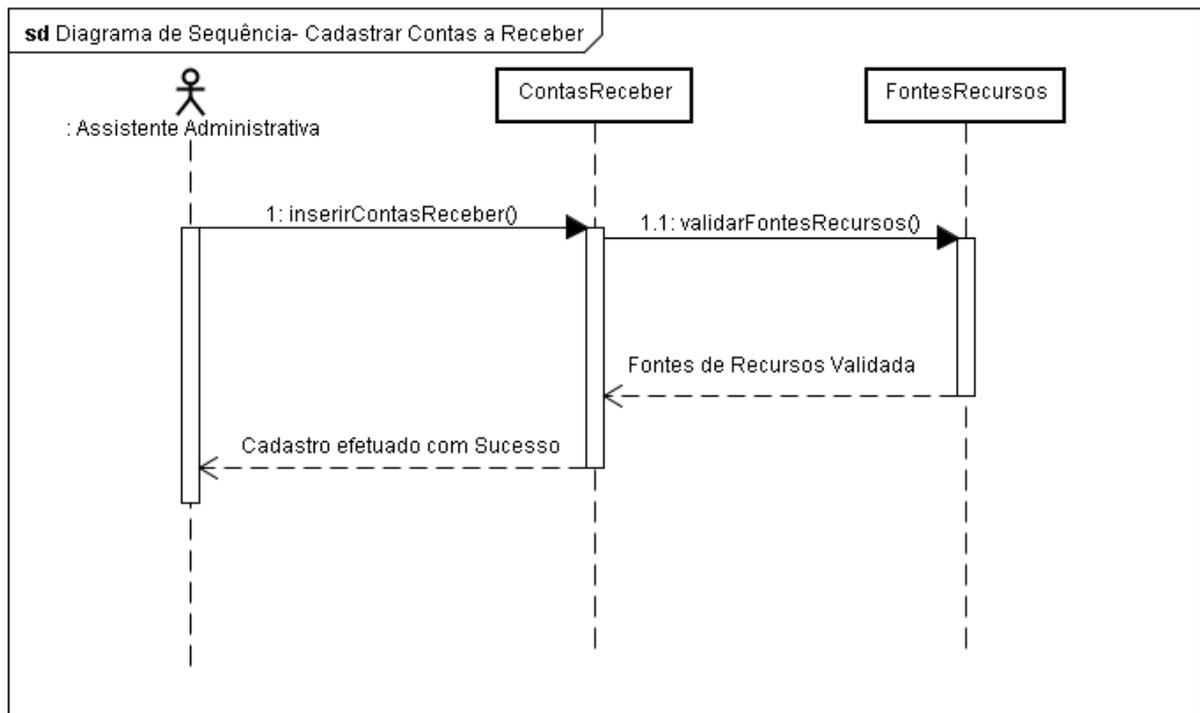
**Figura 19 – Diagrama de Sequência Cadastrar Fontes de Recursos**



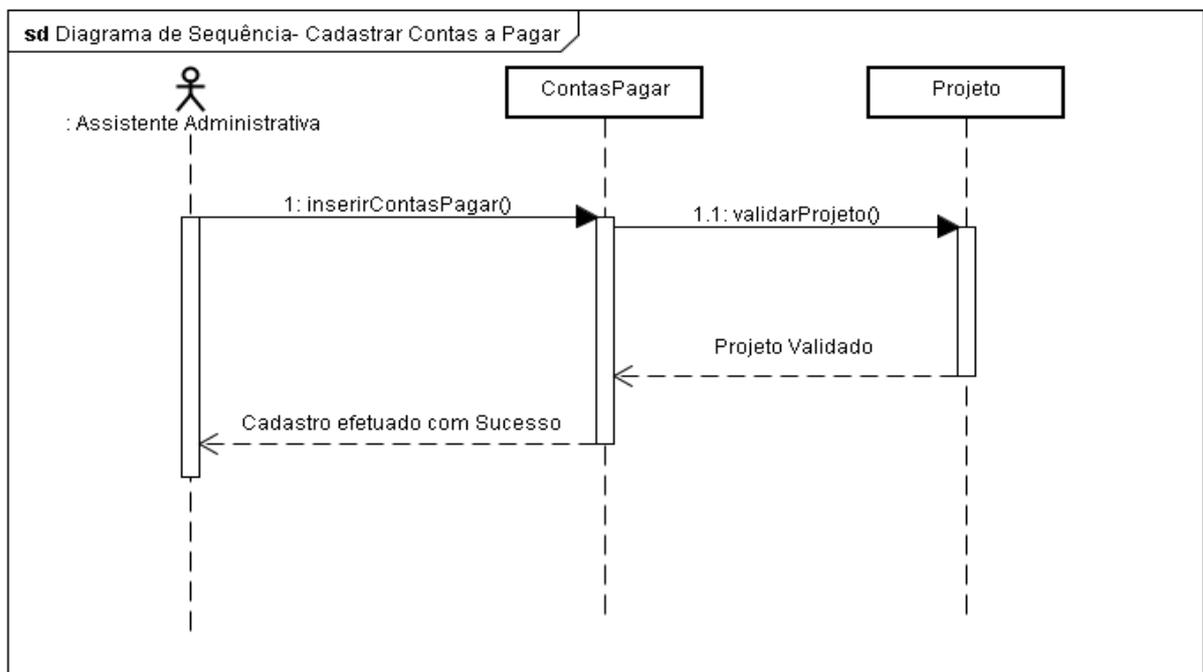
**Figura 20 – Diagrama de Sequência Cadastrar Projeto**



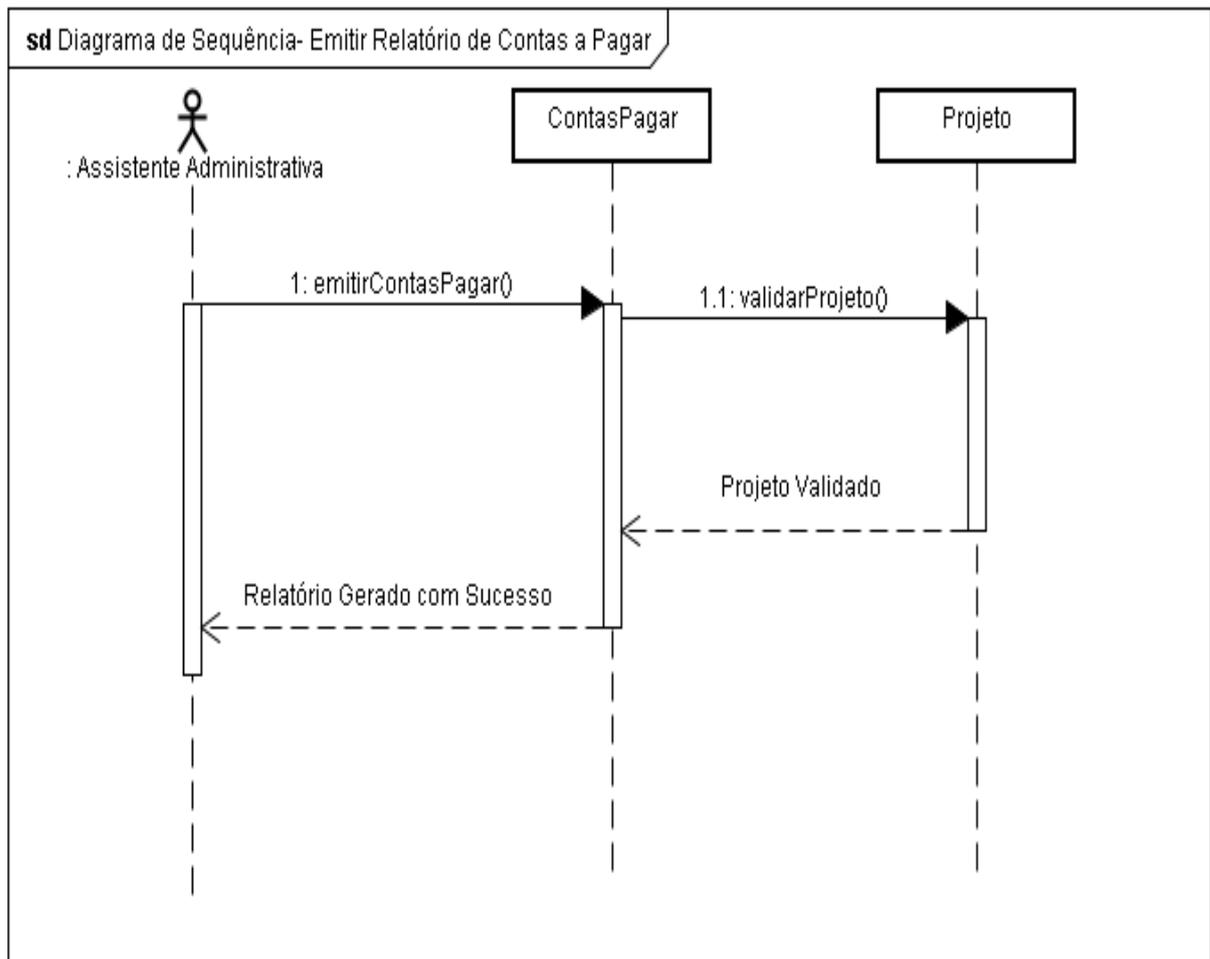
**Figura 21 – Diagrama de Sequência Cadastrar Serviço**



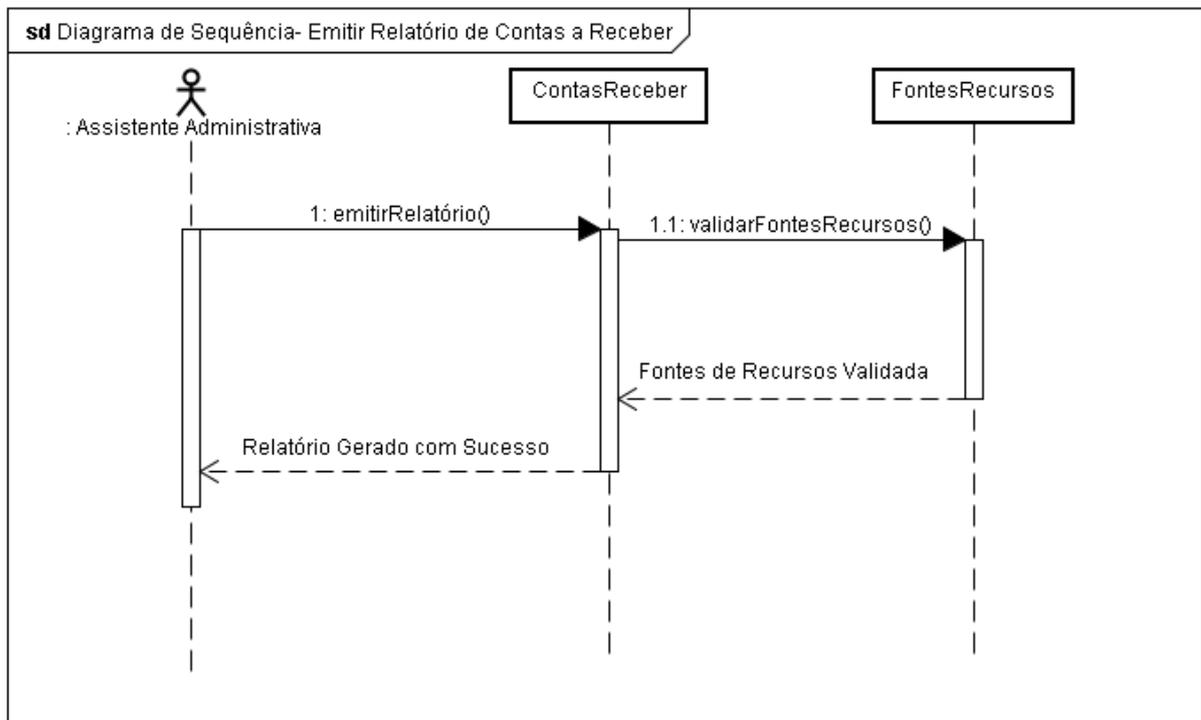
**Figura 22 – Diagrama de Sequência Cadastrar Contas a Receber**



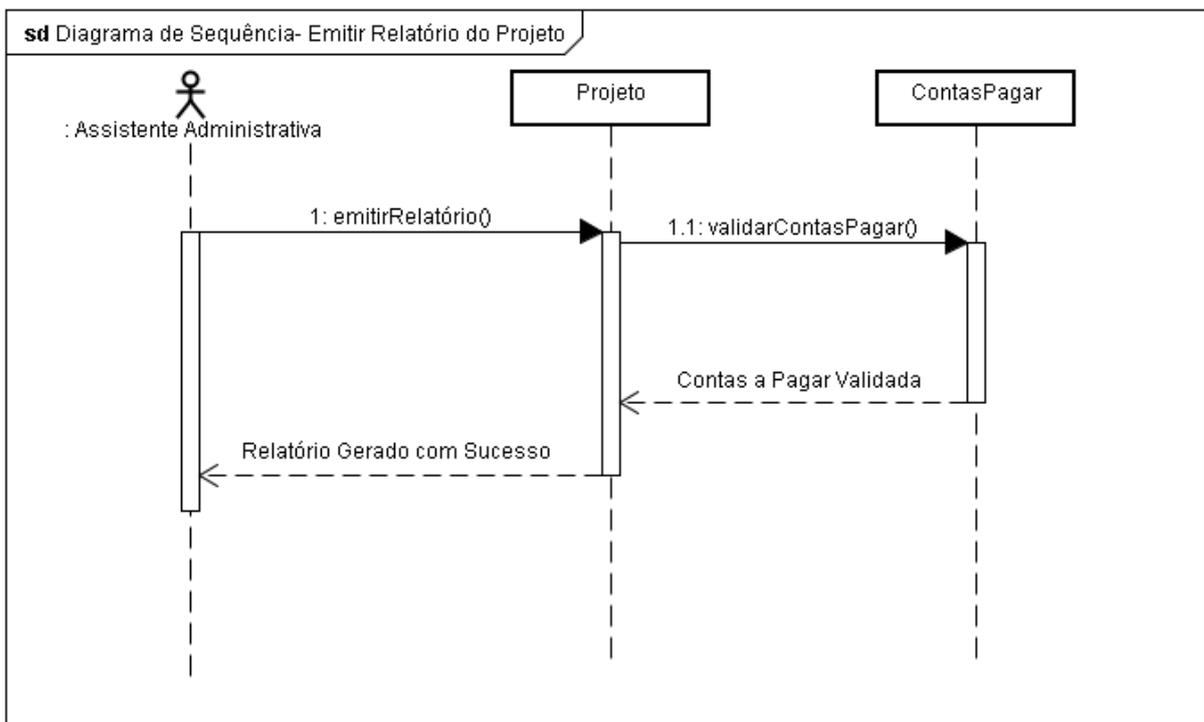
**Figura 23 – Diagrama de Sequência Cadastrar Contas a Pagar**



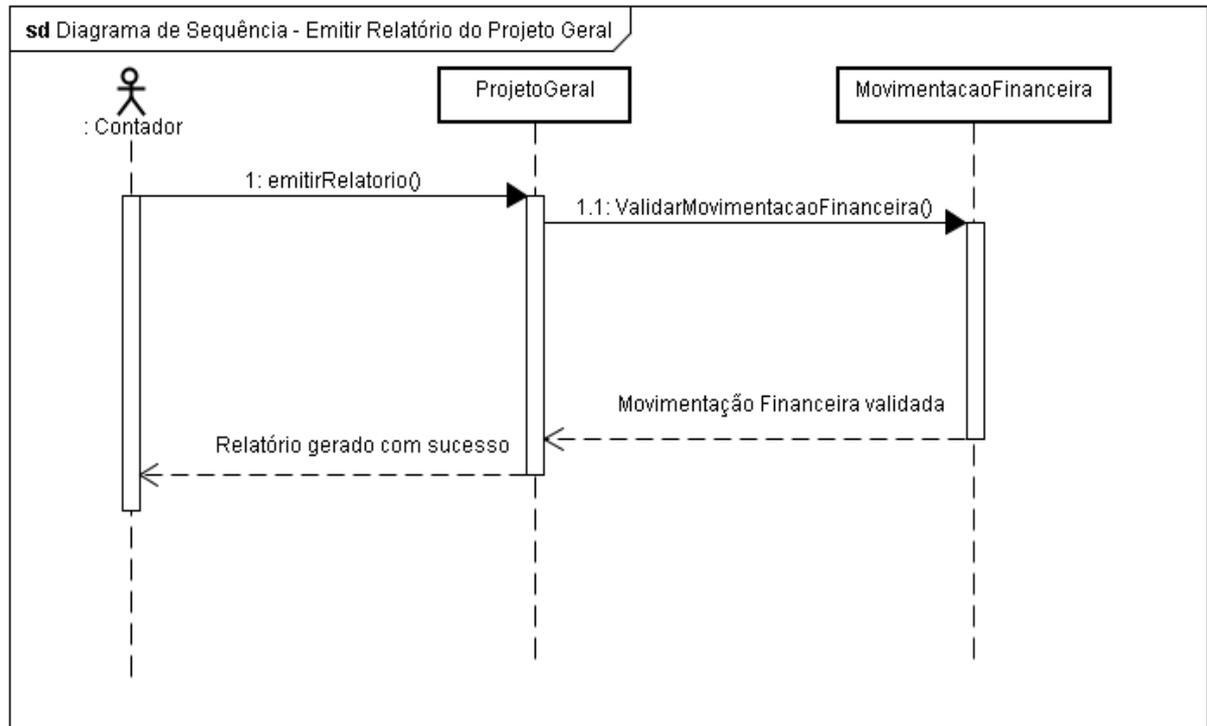
**Figura 24 – Diagrama de Sequência Emitir Relatório Contas a Pagar**



**Figura 25 – Diagrama de Sequência Emitir Relatório Contas a Receber**



**Figura 26 – Diagrama de Sequência Emitir Relatório do Projeto**



**Figura 27 – Diagrama de Sequência Emitir Relatório do Projeto Geral**

## 7.10 Diagrama de Entidade e Relacionamento

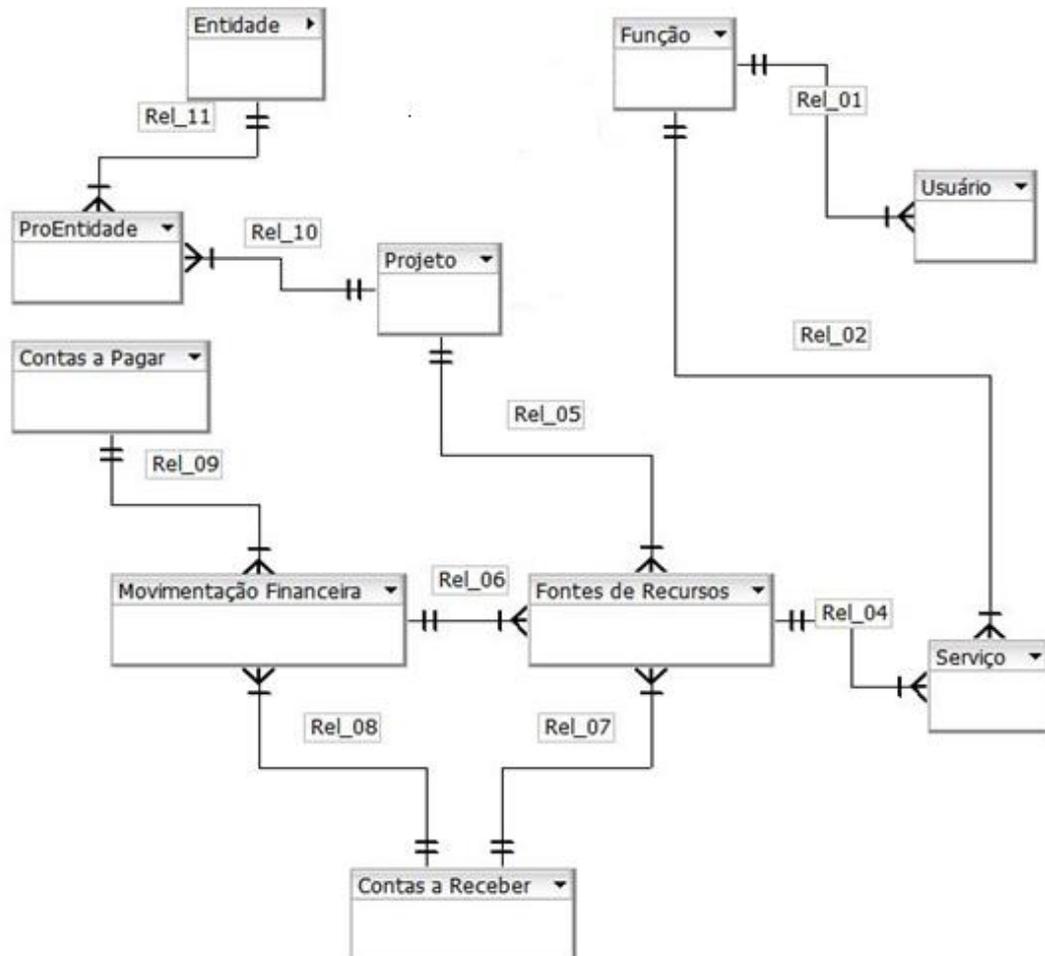


Figura 28 – Diagrama de Entidade e Relacionamento



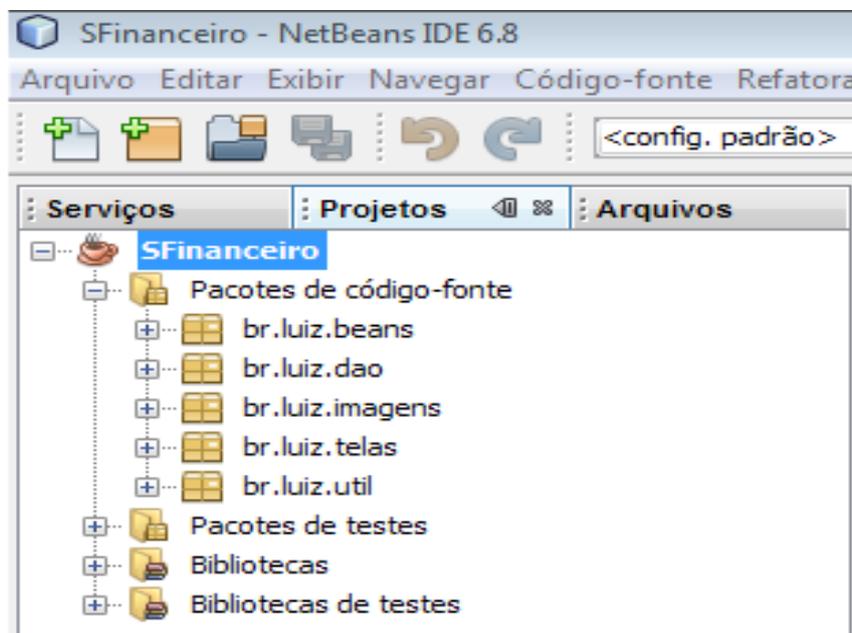
## 9- IMPLEMENTAÇÃO DA APLICAÇÃO

Para a implementação do sistema foi utilizado o ambiente de desenvolvimento NetBeans com a linguagem de programação Java.

### 9.1- ORGANIZAÇÃO DOS PACOTES

O sistema foi organizado em pacotes:

- br.luiz.beans
- br.luiz.dao
- br.luiz.imagens
- br.luiz.telas
- br.luiz.util



**Pacote:** br.luiz.beans: É nele contém as classes de modelagem do projeto, as classes que fazem a persistência dos dados com o banco de dados.

**Pacote:** br.l Luiz.dao: É nele contém as classes que realizam o acesso ao banco de dados, gravar, alterar, excluem informações do banco de dados.

**Pacote:** br.l Luiz.imagens: É nele contém todas imagens dos botões e da tela principal do sistema.

**Pacote:** br.l Luiz.telas: É nele contém a modelagem do projeto, e o código-fonte da interface.

**Pacote:** br.l Luiz.util: Neste pacote estão adicionadas todas as classes que faz conexão com o banco de dados.

## 9.2- INTERFACE DO SISTEMA

### LOGIN

Para acessar sistema abrirá tela de login, contendo usuário cadastrado.



A captura de tela mostra uma janela de login com o título "Cadastro do Login" e um ícone de fechamento. O conteúdo principal da janela é "Acesso ao Sistema". À esquerda, há um ícone de uma fechadura prateada. À direita, há dois campos de entrada de texto: "Usuário" e "Senha". Abaixo dos campos, há dois botões: "Entrar" e "Cancelar". Na base da janela, há o texto "Bem Vindo ao Sistema Financeiro!".

Ao acessar o sistema, após login temos a tela principal, contendo cadastro, movimentações, relatórios, utilitários e sair.



Usuário Luiz

Nessa tela usuário realiza cadastro de entidade e faz pesquisa das entidades cadastradas.

**Cadastro de Entidade**

Nome  Telefone ( ) -

Endereço  Cep

Cidade  U F

Pesquisar por

Código	Nome	Endereço	Cidade

Nessa tela o usuário faz a movimentação relacionada o projeto relacionada com as fontes de recursos.

**Movimentação**

Projeto: Rede Ciranda      Fontes Recursos: Telefônica

Descrição: Capacitação

Data: 22/03/2010

Valor: 5500

Pesquisar por:

Código	Descrição	Valor	Data
2	Capacitação	450.0	2010-11-24 00:00:00.0
3	Despesas Operacionais	4000.0	2010-11-24 00:00:00.0
13	Despesas Operacionais	839.0	2010-11-26 00:00:00.0
6	Entrada de Dinheiro	154.0	2010-01-01 00:00:00.0
4	Equipamentos	3850.0	2010-11-25 00:00:00.0

Nessa tela o usuário faz a movimentação das contas à pagar

**Contas à Pagar**

Fontes Recursos: Estado      Projeto: SEADS

Descrição: Prestação de Serviços

Valor: 1500      Conta Quitada: Sim

Data: 26/11/2010

Pesquisar por:

Código	Projeto	Valor	Data	Quitada
23	SEADS	2500	2010-11-02 00:00:00.0	Não
29	Integraassis	65	2010-01-04 00:00:00.0	Sim
25	Rede Ciranda	10	2010-12-05 00:00:00.0	Sim
33	LA	25	2010-11-26 00:00:00.0	Sim
26	teste	10	2010-12-26 00:00:00.0	Sim

Relatório principal do caixa (iReports).

JasperViewer

100%

# Relatório de Caixa

Data de Operação	Tipo de Operação	Valor Anterior	Valor	Valor Atual
01/01/2010	Crédito	R\$ 0,00	R\$ 123,00	R\$ 123,00
02/01/2010	Crédito	R\$ 123,00	R\$ 17,00	R\$ 140,00
03/01/2010	Crédito	R\$ 140,00	R\$ 65,00	R\$ 205,00
04/01/2010	Débito	R\$ 205,00	R\$ 65,00	R\$ 140,00
05/01/2010	Débito	R\$ 140,00	R\$ 35,00	R\$ 105,00
10/01/2010	Crédito	R\$ 105,00	R\$ 150,00	R\$ 255,00
15/01/2010	Débito	R\$ 255,00	R\$ 200,00	R\$ 55,00
26/11/2010	Crédito	R\$ 55,00	R\$ 150,00	R\$ 205,00
26/11/2010	Crédito	R\$ 205,00	R\$ 120,00	R\$ 325,00
26/11/2010	Débito	R\$ 325,00	R\$ 25,00	R\$ 300,00
05/12/2010	Débito	R\$ 300,00	R\$ 10,00	R\$ 290,00
26/11/2010	Débito	R\$ 290,00	R\$ 25,00	R\$ 265,00
27/12/2010	Débito	R\$ 265,00	R\$ 114,00	R\$ 151,00
26/12/2010	Débito	R\$ 151,00	R\$ 10,00	R\$ 141,00
26/11/2010	Crédito	R\$ 141,00	R\$ 839,00	R\$ 980,00
27/12/2010	Débito	R\$ 980,00	R\$ 35,00	R\$ 945,00
26/11/2010	Crédito	R\$ 945,00	R\$ 10,00	R\$ 955,00
26/11/2010	Débito	R\$ 955,00	R\$ 20,00	R\$ 935,00

## CONCLUSÃO

Com o sistema financeiro desenvolvido, o proprietário terá controle total das informações da entidade: contas a pagar, contas a receber, projetos e fontes de recursos.

O sistema da entidade Associação Filantrópica Nosso Lar, busca melhorar e facilitar as consultas e as movimentações financeiras.

Por meio de entrevistas realizou-se todo levantamento de requisitos e funcionalidades para o sistema, foi percebido uma falta de experiência de ambas as partes, mas que foi bem suprida pelo orientador. O diagrama de classes foi muito importante para o desenvolvimento da implementação.

No desenvolvimento do software foram encontradas muitas dificuldades, pela falta de conhecimento na linguagem Java, tais dificuldades foram supridas com pesquisas na internet, livros e troca de informações com professores e colegas.

Também uma das dificuldades encontradas foi em relação ao framework Hibernate utilizado para manter a persistência com o banco de dados. Inicialmente não havia domínio de conhecimento sobre esta tecnologia, porém com pesquisas em livros, sites fóruns, tutorias e na própria documentação do Hibernate, foram sanados as dificuldades encontradas.

## REFERÊNCIAS

CAMPOS, Augusto; IDE Jude; <http://br-linux.org/linux/node/3335>. Acesso em: 20 de jun. 2010.

GONÇALVES, Edson; Desenvolvendo Aplicações Web com NetBeans IDE 6. Rio de Janeiro: Editora Ciência Moderna, 2008.

GONÇALVES, Edson, Dominando Java Server Faces e Facelets Utilizando Spring 2.5, Hibernate e JPA. Rio de Janeiro: Editora Ciência Moderna Ltda, 2008.

Gonzaga, Jorge Luiz; Dominando o PostgreSQL, Rio de Janeiro: Editora Ciência Moderna Ltda, 2007.

GONÇALVES, Edson, Dominando Relatórios JasperReport com iReport. Rio de Janeiro: Editora Ciência Moderna Ltda.,2008.

LEE, Richard C; TEPFENHART, William M; Uml e C++ Guia Prático de Desenvolvimento Orientado a Objeto. Tradução de Celso Roberto Paschoa. São Paulo: Editora MAKRON Books, 2001.

MARTIN, James; Princípios de Análise e Projeto Baseados em Objetos. 5º Tiragem. Rio de Janeiro: Editora Campus, 1994.

MARTIN, Fowler, SCOTT, Kendall; UML Essencial 2º Edição. Porto Alegre : Bookman, 2000. Apresentação de Ivar Jacobson, 1994.

SHALLOWAY, Allan, Explicando padrões de projeto: Uma nova perspectiva em projeto orientado a objeto. Ana M. de Alencar Price. Porto Alegre Bookman,2004.

