

RICARDO APARECIDO NUNES DA SILVA

**SISTEMA PARA GESTÃO DE EMPRESA DE VIGILÂNCIA
RESIDENCIAL**

Assis

2011

RICARDO APARECIDO NUNES DA SILVA

SISTEMA PARA GESTÃO DE EMPRESA DE VIGILÂNCIA RESIDENCIAL

Trabalho de Conclusão
de Curso apresentado ao Instituto
Municipal de Ensino Superior de
Assis como requisito do Curso
de Graduação

Orientador: Dr. Almir Rogério Camolesi.

Área de Concentração: Desenvolvimento de Sistemas

Assis
2011

FICHA CATALOGRÁFICA

SILVA, Ricardo Aparecido Nunes
SISTEMA PARA GESTÃO DE EMPRESA DE VIGILÂNCIA RESIDENCIAL / Ricardo
Aparecido Nunes da Silva. Fundação Educacional do Município de Assis, 2011.
66p.

Orientador: Dr. Almir Rogério Camolesi
Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis –
IMESA.

CDD: 001.61
Biblioteca da Fema

SISTEMA PARA GESTÃO DE EMPRESA DE VIGILÂNCIA RESIDENCIAL

RICARDO APARECIDO NUNES DA SILVA

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: Dr. Almir Rogério Camolesi

Assis

2011

DEDICATÓRIA

Dedico este trabalho a Deus,
A minha mãe Helena,
Minhas irmãs,
Minha Namorada, e
Meu cunhado Claudinei.

AGRADECIMENTOS

Primeiramente a Deus, é a ele que dirijo minha maior gratidão, mais do que me criar, deu propósito à minha vida. Vem dele tudo o que sou o que tenho e o que espero.

A todos os professores da FEMA, pelos constantes esforços em nos fazer aprender, aprofundando nossos conhecimentos para ficarmos aptos ao mercado de trabalho.

Ao meu orientador e amigo Dr. Almir Rogério Camolesi, pelas suas orientações, idéias e opiniões, pelo constante estímulo transmitido durante o trabalho.

A minha mãe Helena, por ter me ajudado a concluir mais essa etapa em minha vida, e por representar meu maior exemplo e experiência de amor, amizade e lealdade e, sobretudo, pelo apoio e dedicação.

A minha Namorada Márcia Cristiane pela compreensão, amor e apoio em todos os momentos dessa jornada.

Ao meu cunhado Claudinei, por ter me ajudado sempre que precisei nas matérias em que estava com dificuldades.

Aos meus amigos da sala: pela amizade adquirida nesses três anos, pelos momentos difíceis que enfrentamos juntos, e também pelos momentos de alegria.

E finalmente, agradeço a todos que me ajudaram diretamente ou indiretamente para o desenvolvimento desse trabalho. Muito Obrigado.

RESUMO

Nesse trabalho será apresentada a análise de um Sistema para Gestão de Empresa de Vigilância Residencial. O sistema proposto tem por objetivo atender e suprir todas as necessidades da empresa, facilitando os atendimentos realizados e proporcionando um melhor calculo financeiro, evitando assim uma possível perda de dados dos clientes. O Trabalho apresenta também um estudo sobre as tecnologias que foram utilizadas para o desenvolvimento do sistema, como a linguagem de programação C#, o Banco de Dados SQL Server 2008 e a ferramenta de desenvolvimento Visual Studio 2010.

Palavras-chave: C#, SQL Server 2008, Visual Studio 2010.

ABSTRACT

This work will be presented the analysis of a Management System for Monitoring Residential Company. The proposed system aims to meet and supply all the needs of the company, facilitating the care provided and providing a better financial calculation, thus avoiding a possible loss of customer data. The work also presents a study on the technologies that were used for the development of the system, programming language like C#, Database SQL Server 2008 and Visual Studio development tool 2010.

Keywords: C#, SQL Server, Visual Studio 2010.

LISTA DE ILUSTRAÇÕES

Figura 1 – Caso de Uso Geral.....	21
Figura 2 - UC1 Diagrama de Use Case Consultar Clientes	22
Figura 3 - UC2 Diagrama de Use Case Consultar Controle de Recebimento..	23
Figura 4 - UC3 Diagrama de Use Case Consultar Diárias	24
Figura 5 - UC4 Diagrama de Use Case Consultar Veículos.....	25
Figura 6 - UC4 Diagrama de Use Case Cadastrar Funcionários.....	26
Figura 7 - UC6 Diagrama de Use Case Emitir Relatório por Clientes.....	27
Figura 8 – UC7 Diagrama de Use Case Relatorios Controle de Recebimento.	28
Figura 9 – UC8 Diagrama de Use Case Emitir Recibos.	29
Figura 10 - Diagrama de Classes.....	30
Figura 11 – Diagrama de Atividades Geral	31
Figura 12 – Diagrama de Atividades de Controle de Recebimento.....	32
Figura 13 – Diagrama de Atividades Cadastros.....	33
Figura 14 – Diagrama de Atividades Emitir Relatórios.	33
Figura 15 – Diagrama Entidade Relacionamento.....	34
Figura 16 - Work Breakdwon Structure.	35
Figura 17 – Sequenciamento de Atividades.....	36
Figura 18 – Organização das Camadas.....	39
Figura 18 – Funcionamento das Camadas.	41
Figura 19 – Interface de Login.....	41
Figura 20 – Interface do Sistema	42
Figura 21 – Menu Consultar.....	42
Figura 22 – Interface Consultar Clientes	43
Figura 23. Interface Cadastrar.....	44
Figura 24 – Interface Cadastrar Recebimento.....	44
Figura 25 – Código para Inserir Dados	46
Figura 26 – Código para Excluir Dados.....	46
Figura 27 – Código para Alterar Dados.....	47
Figura 28 - Cronograma.....	51

LISTA DE TABELAS

Tabela 1 – Manter Cliente.....	22
Tabela 2 – Manter Controle de Recebimento.....	23
Tabela 3 – Consultar Diárias.....	24
Tabela 4 – Manter Veículos.....	25
Tabela 5 – Manter Funcionários.....	26
Tabela 6 – Emitir Relatório por Cliente.....	27
Tabela 7 – Emitir Relatório Controle de Recebimento.....	28
Tabela 8 – Imprimir Recibos.....	29
Tabela 9 – Tabela Orçamento.....	38

SUMÁRIO

1 – INTRODUÇÃO	12
1.1 – OBJETIVOS	13
1.2 – JUSTIFICATIVA	13
1.3 – ESTRUTURA DO TRABALHO	14
2-TECNOLOGIAS DE ANÁLISE E DESENVOLVIMENTO.....	15
2.1- TECNOLOGIAS UTILIZADAS.....	15
2.1.1– C# (C SHARP)	15
2.1.2 – VISUAL STUDIO	15
2.1.3– SQL SERVER 2008.....	16
2.1.4 – CRYSTAL REPORT	16
3 - ANÁLISE E ESPECIFICAÇÃO DO SISTEMA	18
3.1 - METODOLOGIAS DE ANÁLISE	18
3.2 – LEVANTAMENTOS DE REQUISITOS	18
3.3 - LISTA DE EVENTOS	19
3.4 - CASOS DE USO.....	21
3.5 - DIAGRAMA DE CLASSES.....	30
3.6 - DIAGRAMA DE ATIVIDADES	31
3.7 - DIAGRAMA ENTIDADE RELACIONAMENTO	34
4 - PROJETO	35
4.1 - WORK BREAKDOWN STRUCTURE	35
4.2 - SEQUENCIAMENTO DE ATIVIDADES	35
4.3 – ORÇAMENTO.....	36
4.3.1–RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO	36
4.3.2 – CUSTO DE MATERIAIS.....	37
4.3.3 – Orçamento do Projeto	38
5 – IMPLEMENTAÇÃO DA APLICAÇÃO.....	39
5.1 – ORGANIZAÇÃO DAS CAMADAS DE CÓDIGO FONTE.....	39
5.2 – ENTENDENDO AS CAMADAS.....	40
5.3 – INTERFACE DO SISTEMA	41
5.4 – DESCRIÇÕES DAS OPERAÇÕES DE MANUTENÇÃO DE DADOS.....	45
CONCLUSÃO	48

REFERÊNCIAS.....	50
ANEXO – CRONOGRAMA.....	51

1 – INTRODUÇÃO

A Empresa *PROTEGE* foi criada em 25 de junho de 1995, em uma proposta para diminuir o índice de assaltos e furtos nos bairros da cidade de Cândido Mota e conta com a ajuda da Polícia Militar da cidade.

A partir de Junho de 2002, a empresa passou a ter um novo administrador, Alex Doum da Silva Correa, que inovou o setor com novos e diferenciados serviços, nos quais os comércios têm atendimentos diferenciados das residências. A partir daí tem atendido mais de 300 residências nos bairros da cidade.

Desde o seu início o gerenciamento desses atendimentos são feitos manualmente, permitindo que erros humanos ocorram ou até mesmo a perda de informações relevantes, haja vista que, um software traria segurança, agilidade e confiabilidade para a empresa.

Um dos problemas que o administrador tem enfrentado é a falta de um software para o gerenciamento desses dados, sendo que cada estabelecimento efetua o pagamento em dias diferenciados, e não são armazenados os dados do funcionário que trabalhou em um determinado dia, surgindo assim à proposta de desenvolver um software para manipular os atendimentos e serviços realizados no dia-dia da empresa.

O sistema que foi implementado visa suprir todas as necessidades da empresa, para manter um maior controle sobre todos os dados referentes a situações de cada residência, controlar o fluxo de caixa, emitir relatórios mensais, entre outros.

1.1 – OBJETIVOS

O Objetivo deste trabalho foi prover um software que facilite os atendimentos realizados pelos vigilantes, integrando informações a respeito do atendimento de cada residência dos Bairros da Cidade de Cândido Mota.

1.2 – JUSTIFICATIVA

Pesquisas realizadas na rede mundial de computadores mostraram a inexistência de um software que gerencie esses tipos de dados. Desta forma surgiu à oportunidade de desenvolver o mesmo, a fim de informatizar e gerenciar as informações relacionadas ao atendimento das residências, para a Empresa de Vigilância Residencial. O software pode levantar todos e quaisquer dados referentes ao atendimento e aos controles de recebimento das residências.

As necessidades que levaram o surgimento do sistema foram: as informações das residências serem realizados manualmente; a necessidade de agilidade e proteção desses dados; e a emissão de relatórios mensais.

1.3 – ESTRUTURA DO TRABALHO

Para uma melhor compreensão e leitura, o trabalho foi organizado em capítulos.

No capítulo 2 serão apresentadas as tecnologias utilizadas para a análise e o desenvolvimento do sistema, com uma breve descrição de cada uma. A seguir no capítulo 3, as etapas de Análise e Especificações do Sistema contemplando o Levantamento de Requisitos, Lista de Eventos, o Caso de Uso e suas especificações, Diagrama de Classe, Diagrama de Sequência e Diagrama de Entidade Relacionamento. O capítulo 4 mostrará a WBS (Work BreakDown Structure), Sequenciamento das Atividades e Orçamento do Sistema. O Quinto capítulo contém a implementação do sistema, com as divisões das camadas e dos formulários, mostrando a interface do software. Ao final a conclusão do trabalho, referência e cronograma em anexo.

2-TECNOLOGIAS DE ANÁLISE E DESENVOLVIMENTO

Este capítulo contém a descrição das tecnologias que foram utilizadas para o desenvolvimento do Sistema.

2.1- TECNOLOGIAS UTILIZADAS

Para o desenvolvimento do sistema foi utilizada a linguagem de programação C#. O sistema foi desenvolvido utilizando a ferramenta *Visual Studio 2010*, para armazenamento dos dados o banco de dados *Sql Server*, e para a emissão de relatórios foi utilizado o *Crystal Report*.

2.1.1– C# (C SHARP)

C# é uma linguagem de programação orientada a objetos criada pela Microsoft, faz parte da sua plataforma .NET. A companhia baseou C# na linguagem C++ e JAVA.

A linguagem C# foi criada junto com a arquitetura .NET. Embora existam várias outras linguagens que suportam essa tecnologia (como VB.NET, C++, J#) C# é considerada a linguagem símbolo do .NET pois foi criado praticamente do zero para funcionar na nova plataforma, sem reocupações de compatibilidade com o código de legado, o compilador C# foi o primeiro a ser desenvolvido e a maior parte das classes do .NET Framework foram desenvolvidas em C#.

A criação da linguagem, embora tenha sido feita por vários desenvolvedores, é atribuída principalmente a *Anders Hejlsberg*, hoje um *Distinguished Engineer* na *Microsoft*. *Anders Hejlsberg* era desenvolvedor de compiladores na *Borland*, e entre suas criações mais conhecidas estão o Turbo Pascal e o Delphi.

2.1.2 – VISUAL STUDIO

Microsoft Visual Studio é um pacote de programas da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e J# (J Sharp). Também é um grande produto de desenvolvimento na área web, usando a plataforma do

ASP.NET. As linguagens com maior frequência nessa plataforma são: VB.NET (Visual Basic.NET) e o C# (lê-se C Sharp).

Em 12 de Abril de 2010 a Microsoft lançou o Visual Studio 2010, de nome de código Dev10, e a .NET Framework 4.

Foi lançado com o objetivo de ser a IDE mais completa disponível no mercado. Para isso conta com suporte a desenvolvimento de aplicativos Web, aplicativos para Windows Phone, SharePoint além de aprimorar os já conhecidos, Windows Forms, Web Forms e também plataformas como o Microsoft XNA. Oferece IntelliTrace, Gerenciamento de Ciclo de Vida de Aplicativos (ALM), uma nova interface desenvolvida com WPF (Windows Presentation Foundation), para tornar a IDE mais intuitiva, sistema de busca muito mais eficaz, dentre outros.

2.1.3– SQL SERVER 2008

SQL Server 2008 Trata-se uma plataforma de banco de dados que facilita o desenvolvimento de aplicativos ricos em recursos controlados por dados, oferece segurança de armazenamento aprimorada e é rápido de implantar.

SQL Server 2008 é um banco simples, livre e fácil de usar baseado na tecnologia SQL Server 2005. Ele é projetado para fornecer uma plataforma de banco de dados que oferece facilidade de uso, permitindo implementações rápidas para cenários de seu usuário. A concepção e desenvolvimento de aplicações de banco de dados são facilitadas pela integração com os projetos do Visual Studio de maneiras simplificadas e automatizadas.

2.1.4 – CRYSTAL REPORT

SAP Crystal Reports é uma ferramenta que permite a criação de relatórios. Pertence à SAP Business Objects. Pode ser aplicado juntamente em linguagens como o Visual Basic, linguagens .NET ou diretamente em aplicações web ou sua própria sintaxe, o Crystal sintaxe. É um dos mais utilizados no mundo e já está em sua 12ª edição. Não é somente um criador de relatórios: ele pode também criar várias outras coisas como certificados, carnês para loja. Podemos inserir fórmulas nesses relatórios. O Crystal também pode

receber seus dados de um programa, via um Record set ou então se conectar diretamente a um Banco de dados relacional, como Microsoft SQL Server, Sybase ou Oracle para buscar seus dados.

3 - ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

Este capítulo contém a descrição, a análise e a especificação do sistema proposto.

3.1 - METODOLOGIAS DE ANÁLISE

O sistema foi modelado utilizando a metodologia de Análise Orientada a Objetos. Para essa análise foi utilizado os conceitos de UML (*Unified Modeling Language*) que é uma linguagem de modelagem para documentar e visualizar os artefatos que especificamos e construímos na análise de um sistema (LEE, TEPFENHART 2001).

3.2 – LEVANTAMENTOS DE REQUISITOS

Os requisitos foram levantados através de entrevistas com Alex Doum da Silva Correa que atualmente é o administrador do Setor. Durante as entrevistas foram elencados alguns requisitos como:

- Cadastro de Clientes, Fornecedores, Contas a Pagar.
- Realização dos Atendimentos com a maior segurança dos Dados.
- A possibilidade de o sistema listar todos os clientes que efetuaram ou não o pagamento em determinado mês.
- Relatórios mensais de todos os clientes, valor recebido, valor a receber, com nome dos mesmos anexados.
- Realizar controle de recebimento dos Clientes.
- Realizar controle das despesas do Setor (Contas a pagar).
- Impressão dos recibos dos clientes que pagam em determinado dia. (Ex: imprimir só os clientes que pagam no dia 5 ou outro dia estipulado).
- Realizar impressão de recibos

3.3 - LISTA DE EVENTOS

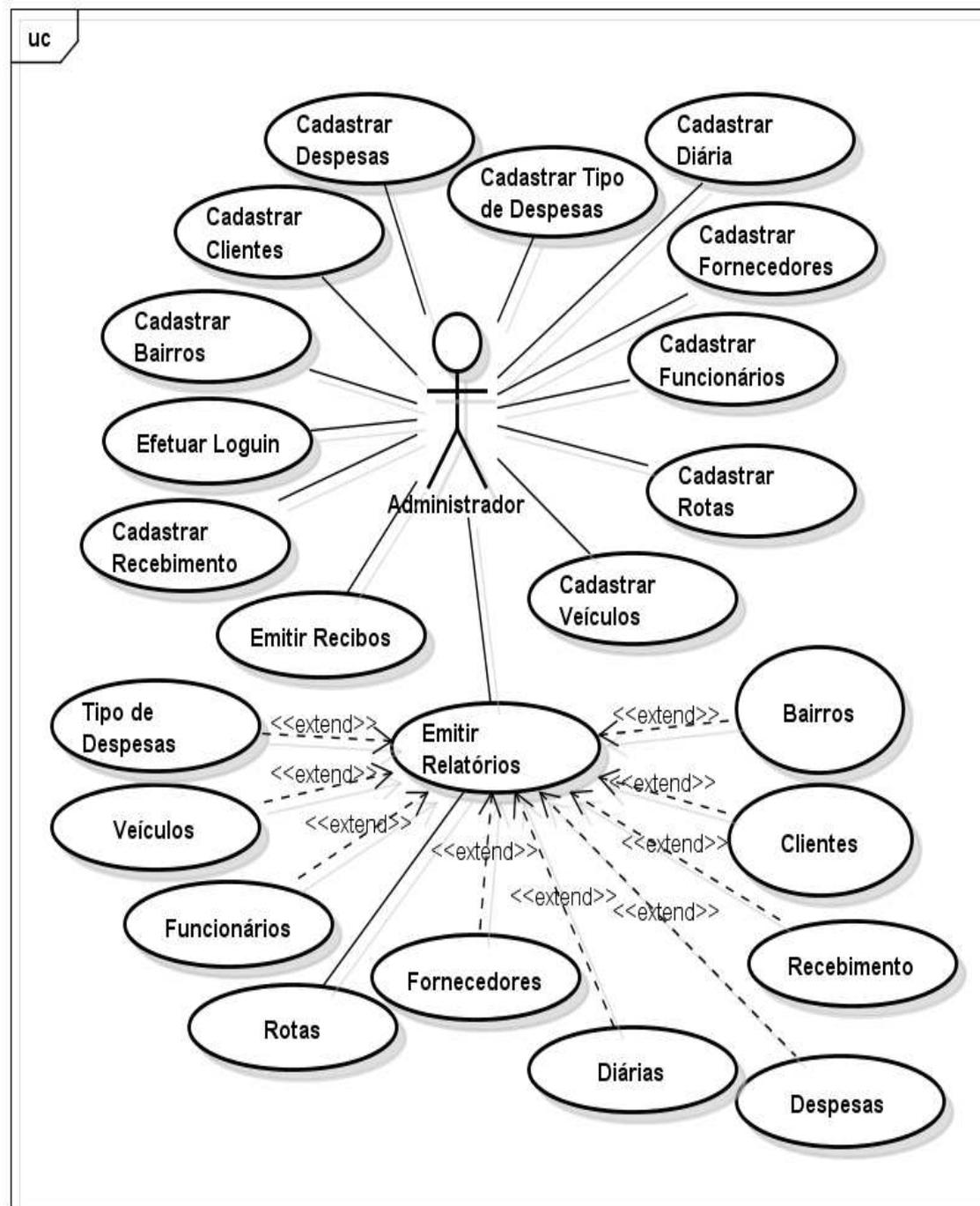
Para modelar o comportamento dos sistemas baseados em objetos, determinam-se quais eventos acontecem. Eventos fazem com que os sistemas tomem várias ações (LEE, TEPFENHART 2001). A seguir são descritos os principais eventos relacionados com as entidades que interagem com o sistema:

1. Efetuar Login
2. Cadastrar Bairros
3. Cadastrar Clientes
4. Cadastrar Despesas
5. Cadastrar Diárias
6. Cadastrar Fornecedores
7. Cadastrar Funcionários
8. Cadastrar Recebimento
9. Cadastrar Rotas
10. Cadastrar Tipo de Despesas
11. Cadastrar Veículos
12. Emitir Relatório de Bairros Cadastrados
13. Emitir Relatório de Todos os Clientes Cadastrados
14. Emitir Relatório de Clientes que Efetuam Pagamento em Determinado Mês
15. Emitir Relatório do Controle de Recebimento
16. Emitir Recibos em Determinado Dia
17. Emitir Relatório de Clientes por Grupo de Bairros

18. Emitir Relatório de Clientes por Grupo de Setores
19. Emitir Relatório das Despesas
20. Emitir Relatório de Fornecedores
21. Emitir Relatório de Funcionários
22. Emitir Relatório de Todos do Controle de Recebimento
23. Emitir Relatório de Todos os Recebimentos por grupo de Clientes
24. Emitir Relatório de Rotas
25. Emitir Relatório de Tipo de Despesas
26. Emitir Relatório de Veículos

3.4 - CASOS DE USO

Um caso de uso especifica uma sequência de ações, inclusive variantes, que um sistema realiza e que produz um observável resultado de valor para um particular ator (LEE, TEPFENHART 2001).



powered by astah

Figura 1 - Caso de Uso Geral

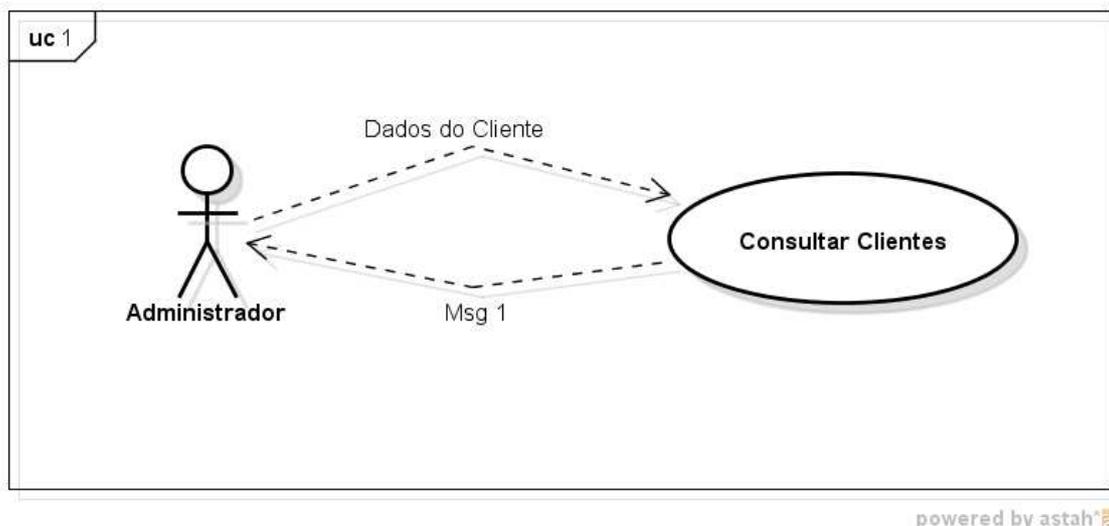


Figura 2 - UC1 Diagrama de Use Case Consultar Clientes

Nome da Use Case	Emitir Cadastrar Clientes
Ator (es)	Administrador
Pré-Condição	O Administrador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o Cadastro dos clientes.</p> <p>2- O Administrador informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O Administrador seleciona o botão <i>Cadastrar</i>.</p> <p>4- O sistema emite uma mensagem "<i>Cliente Cadastrado com sucesso</i>".</p> <p>5- O sistema cadastra o cliente.</p>
Cenário Alternativo	O Administrador poderá cancelar o cadastro durante o processo.
Casos de Teste	<p>2.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>2.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>2.3- O sistema cancela a operação.</p>

Tabela 1 - Manter Cadastro de Clientes

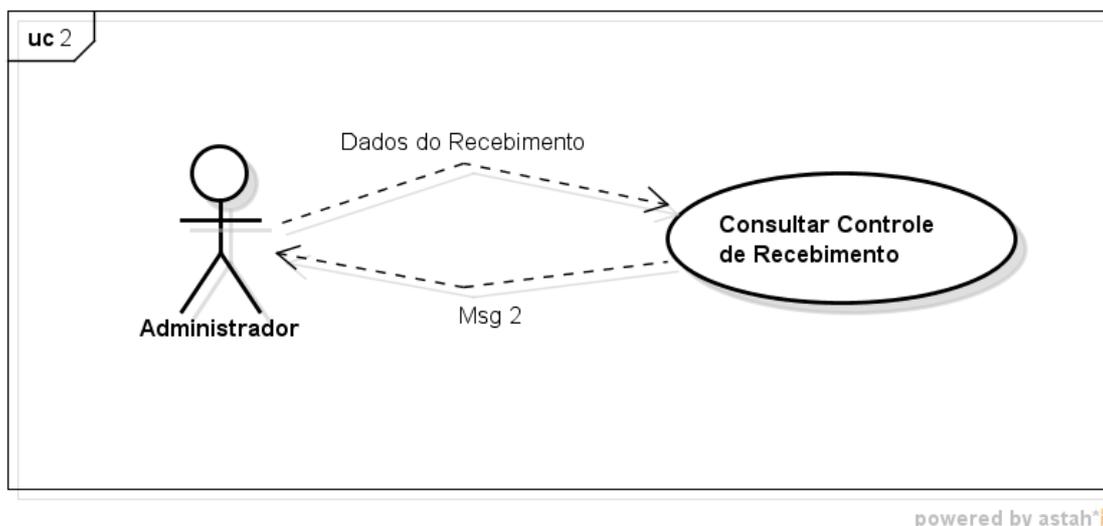


Figura 3 - UC2 Diagrama de Use Case Consultar Controle de Recebimento

Nome da Use Case	Efetuar Recebimento
Ator (es)	Administrador
Pré-Condição	O Administrador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O Administrador Seleciona a opção <i>Consultar Recebimento dos Clientes</i>.</p> <p>2- O Sistema abre uma interface informando todos os clientes que não efetuaram o pagamento da mensalidade.</p> <p>3- O Administrador escolhe o cliente e a seguir seleciona o botão <i>Efetuar Recebimento</i>.</p> <p>4- O sistema emite a mensagem "<i>Mensalidade paga com sucesso</i>".</p> <p>5- O sistema da baixa na mensalidade.</p>
Cenário Alternativo	O Administrador poderá cancelar o processo durante o cadastro.
Casos de Teste	<p>3.1- O sistema não confirma o recebimento e emite uma mensagem de erro.</p> <p>3.2- O sistema cancela a operação.</p>

Tabela 1 - Manter Controle de Recebimento

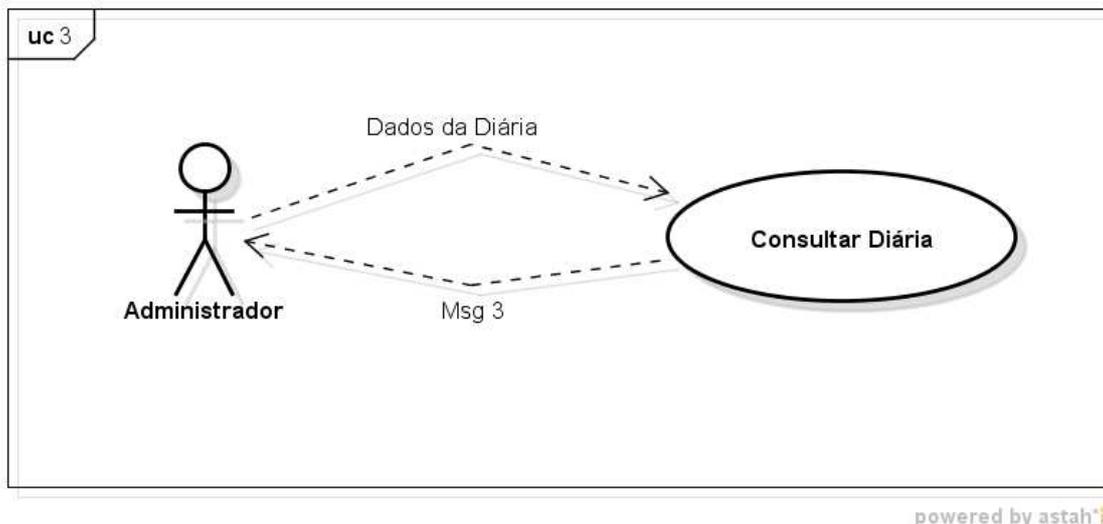


Figura 4 - UC3 Diagrama de Use Case Consultar Diárias

Nome da Use Case	Consultar Diária
Ator (es)	Administrador
Pré-Condição	Administrador deverá estar autenticado no sistema.
Cenário Principal	<p>1- Administrador seleciona a opção <i>Consultar Diária</i>.</p> <p>2- O Sistema abre uma interface informando todos os dados das Diárias.</p> <p>3- O Administrador preenche os dados da diária e seleciona o botão <i>Cadastrar</i>.</p> <p>4- O sistema emite a mensagem "<i>Diária Cadastrada com Sucesso</i>".</p>
Cenário Alternativo	Administrador poderá cancelar o processo durante a consulta.
Casos de Teste	<p>1.1- O Sistema informa que não foram cadastrados diários no mes.</p> <p>1.2- O Sistema volta para o inicio.</p>

Tabela 2 – Consultar Diárias

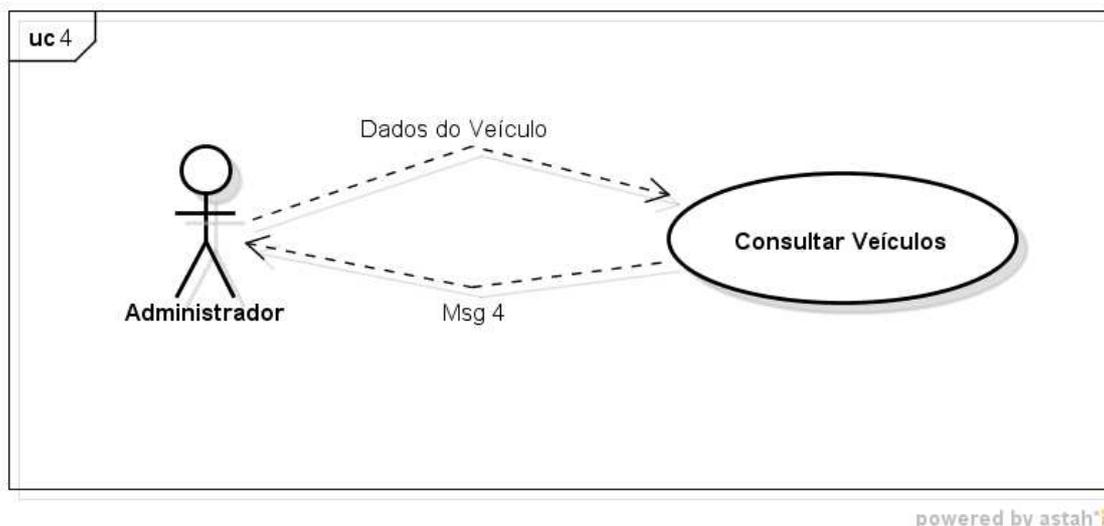


Figura 5 - UC4 Diagrama de Use Case Consultar Veículos

Nome da Use Case	Cadastrar Veículos
Ator (es)	Administrador
Pré-Condição	O Administrador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o Cadastro do Veículo.</p> <p>2- O Administrador informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O Administrador seleciona o botão <i>Cadastrar</i>.</p> <p>4- O sistema emite uma mensagem "<i>Veículo Cadastrado com Sucesso</i>".</p> <p>5- O sistema cadastra o veículo.</p>
Cenário Alternativo	O Administrador poderá cancelar o cadastro durante o processo.
Casos de Teste	<p>2.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>2.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>2.3- O sistema cancela a operação.</p>

Tabela 3 - Manter Veículos

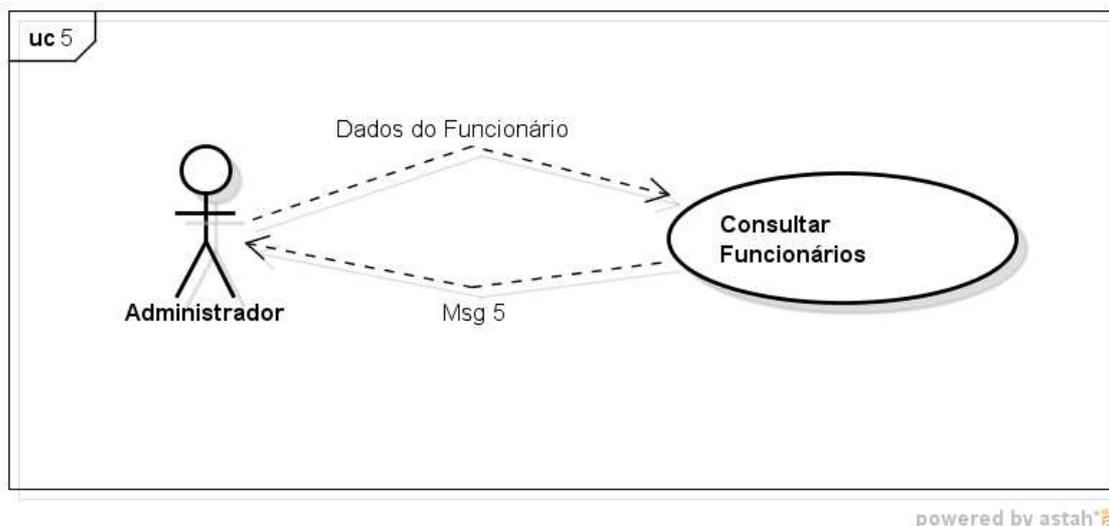


Figura 6 - UC5 Diagrama de Use Case Cadastrar Funcionários

Nome da Use Case	Cadastrar Funcionários
Ator (es)	Administrador
Pré-Condição	O Administrador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o Cadastro do funcionário.</p> <p>2- O Administrador informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O Administrador seleciona o botão <i>Cadastrar</i>.</p> <p>4- O sistema emite uma mensagem "<i>Funcionário Cadastrado com Sucesso</i>".</p> <p>5- O sistema cadastra o funcionário.</p>
Cenário Alternativo	O Administrador poderá cancelar o cadastro durante o processo.
Casos de Teste	<p>2.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>2.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>2.3- O sistema cancela a operação.</p>

Tabela 4 – Manter Funcionário

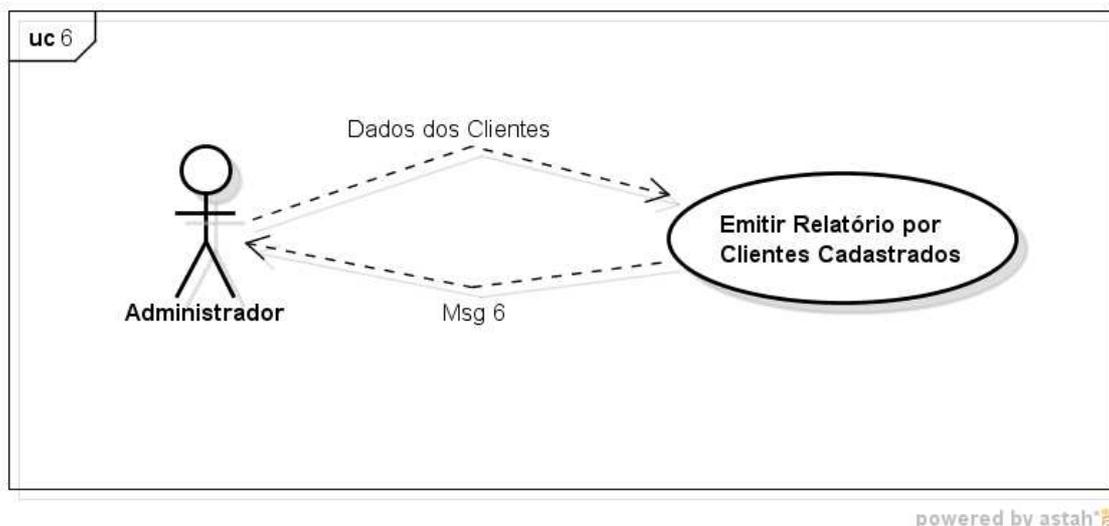


Figura 7 – UC6 Diagrama de Use Case Emitir Relatório por Clientes

Nome da Use Case	Emitir Relatório por Clientes
Ator (es)	Administrador
Pré-Condição	O Administrador deverá estar autenticado no sistema.
Descrição	Os relatórios serão impressos conforme solicitados.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema disponibiliza os dados necessários para o relatório. 2- O Administrador seleciona o botão <i>Visualizar Relatório</i>. 3- O Administrador seleciona o botão <i>Imprimir</i>. 4- O sistema imprime o relatório com sucesso.
Cenário Alternativo	O Administrador poderá visualizar o relatório e não imprimir.
Casos de Teste	4.1- O coordenador cancela a operação.

Tabela 6 - Emitir Relatório por Cliente

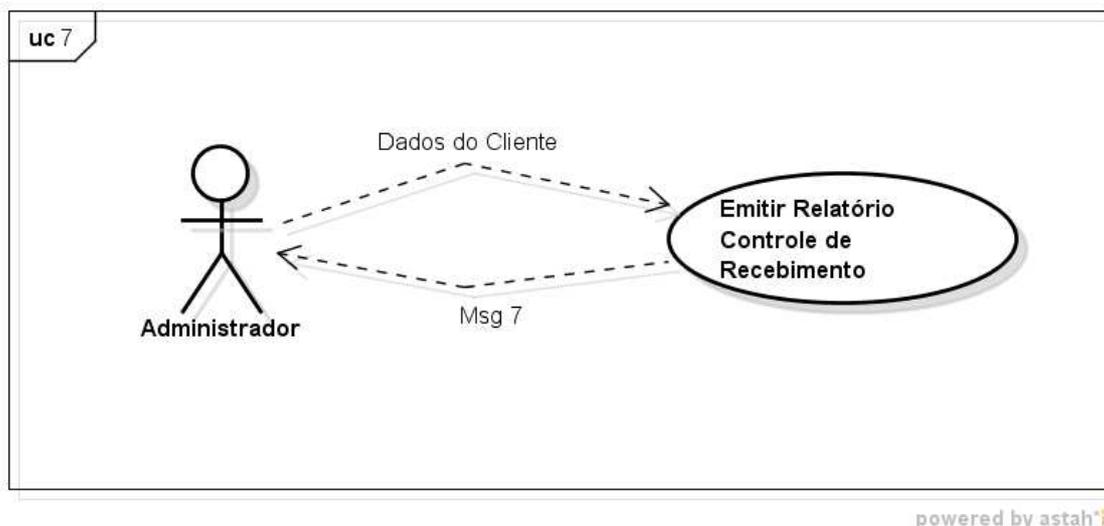


Figura 8 – UC7 Diagrama de Use Case Emitir Relatório Controle de Recebimento

Nome da Use Case	Emitir Relatório Controle de Recebimento
Ator (es)	Administrador
Pré-Condição	O Administrador deverá estar autenticado no sistema.
Descrição	Os relatórios serão impressos conforme solicitados.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema disponibiliza os dados necessários para o relatório. 2- O Administrador seleciona o botão <i>Visualizar Relatório</i>. 3- O Administrador seleciona o botão <i>Imprimir</i>. 4- O sistema imprime o relatório com sucesso.
Cenário Alternativo	O Administrador poderá visualizar o relatório e não imprimir.
Casos de Teste	4.1- O Administrador cancela a operação.

Tabela 7 - Emitir Relatório Controle de Recebimento

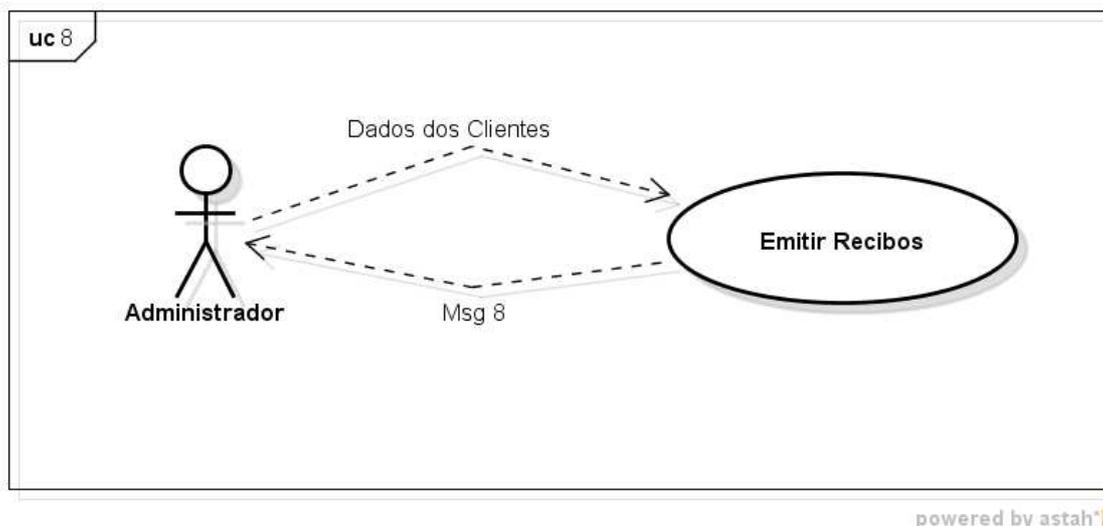


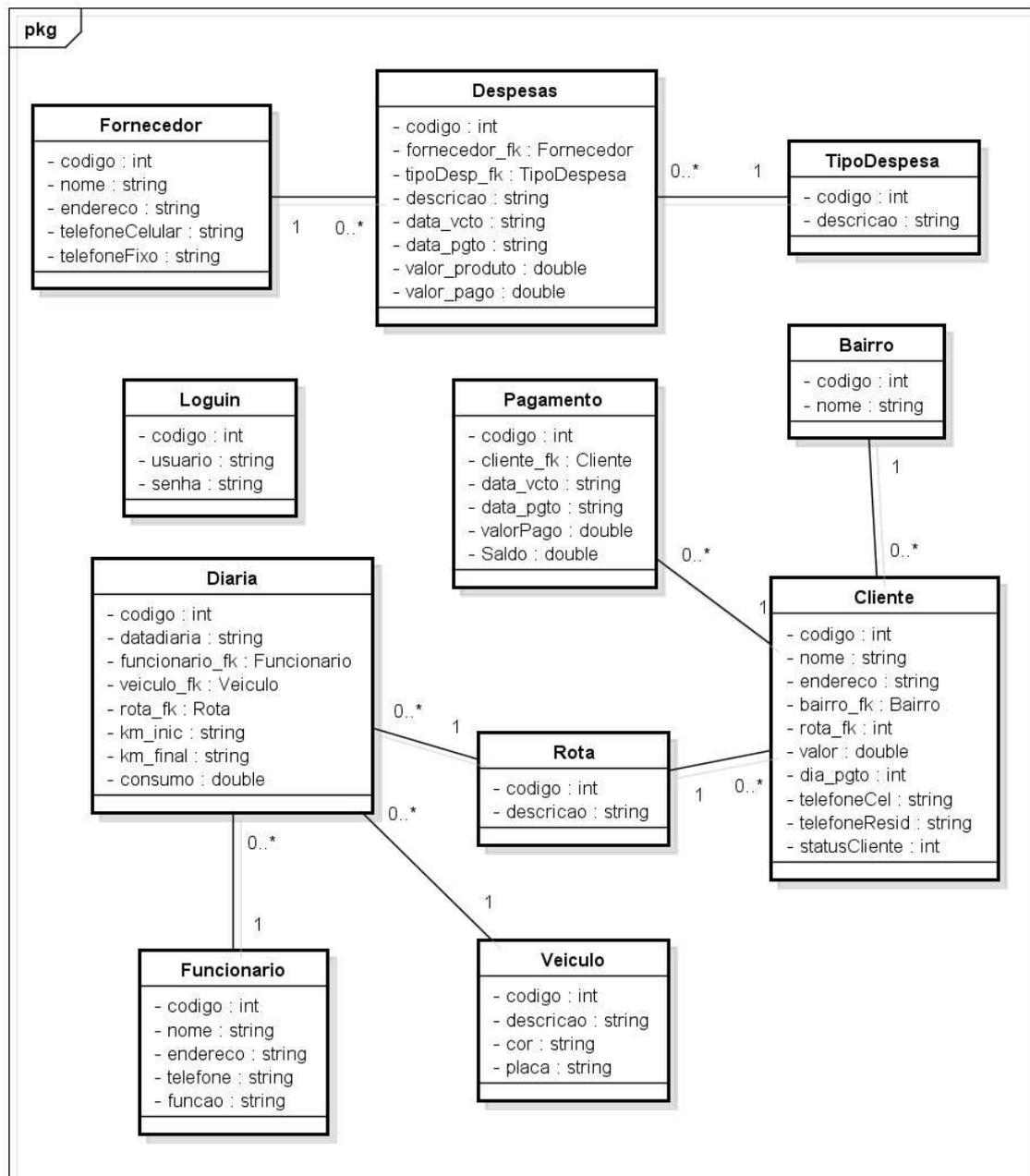
Figura 9 – UC8 Diagrama de Use Case Imprimir Recibos

Nome da Use Case 9	Emitir Recibos
Ator (es)	Administrador
Pré-Condição	O Administrador deverá estar autenticado no sistema.
Descrição	Os recibos serão impressos conforme solicitados.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema disponibiliza os dados necessários para a impressão dos recibos. 2- O Administrador seleciona o botão <i>imprimir recibos</i> do dia escolhido. 3- O Sistema faz a busca e verifica todos que estão cadastrados naquele dia. 4- O Sistema informa quais clientes estão cadastrados naquele dia. 5- O Sistema emite uma mensagem “<i>Deseja Imprimir os Dados?</i>”. 6- O Administrador Seleciona o Botão <i>OK</i>. 7- O Sistema Imprime os dados correspondentes.
Cenário Alternativo	O Administrador poderá visualizar os recibos e não imprimir.
Casos de Teste	4.1- O Administrador cancela a operação.

Tabela 8 – Imprimir Recibos

3.5 - DIAGRAMA DE CLASSES

Um Diagrama de Classes descreve os tipos de objetos no sistema e os vários tipos de relacionamentos estático que existem entre eles. O Diagrama de classe mostra também atributos e operações de uma classe e as restrições à maneira com que os objetos são conectados (BOOCH, JACOBSON, RUMBAUGH,2000).



powered by astah®

Figura 10 - Diagrama de Classes

3.6 - DIAGRAMA DE ATIVIDADES

Um diagrama de atividade descreve como é feita a coordenação de atividades. Alguns autores consideram que os diagramas de atividade são um melhor mecanismo para mostrar dependências essenciais entre atividades realizadas por várias entidades (LEE, TEPFENHART 2001).

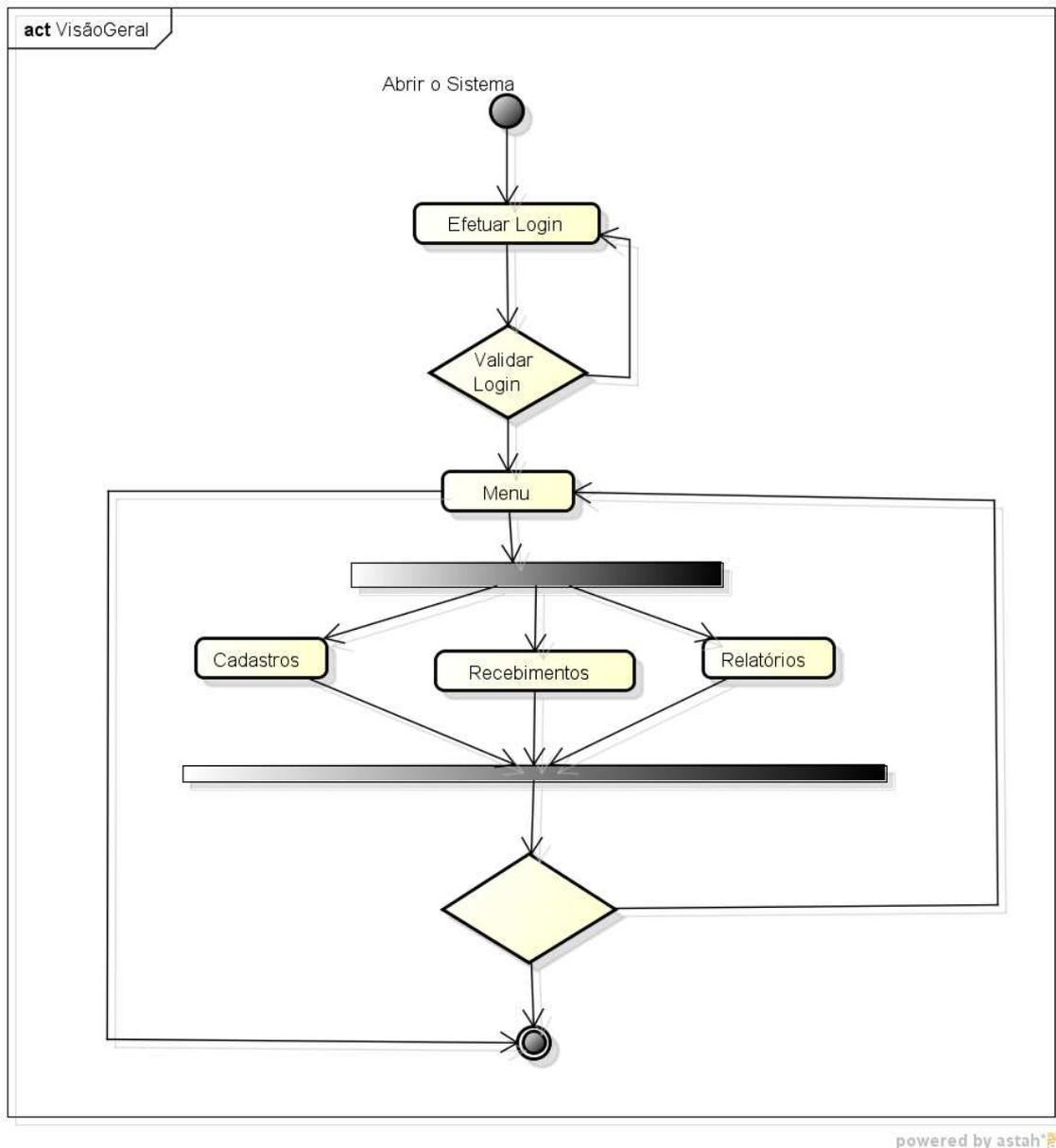


Figura 11 - Diagrama de Atividades Geral

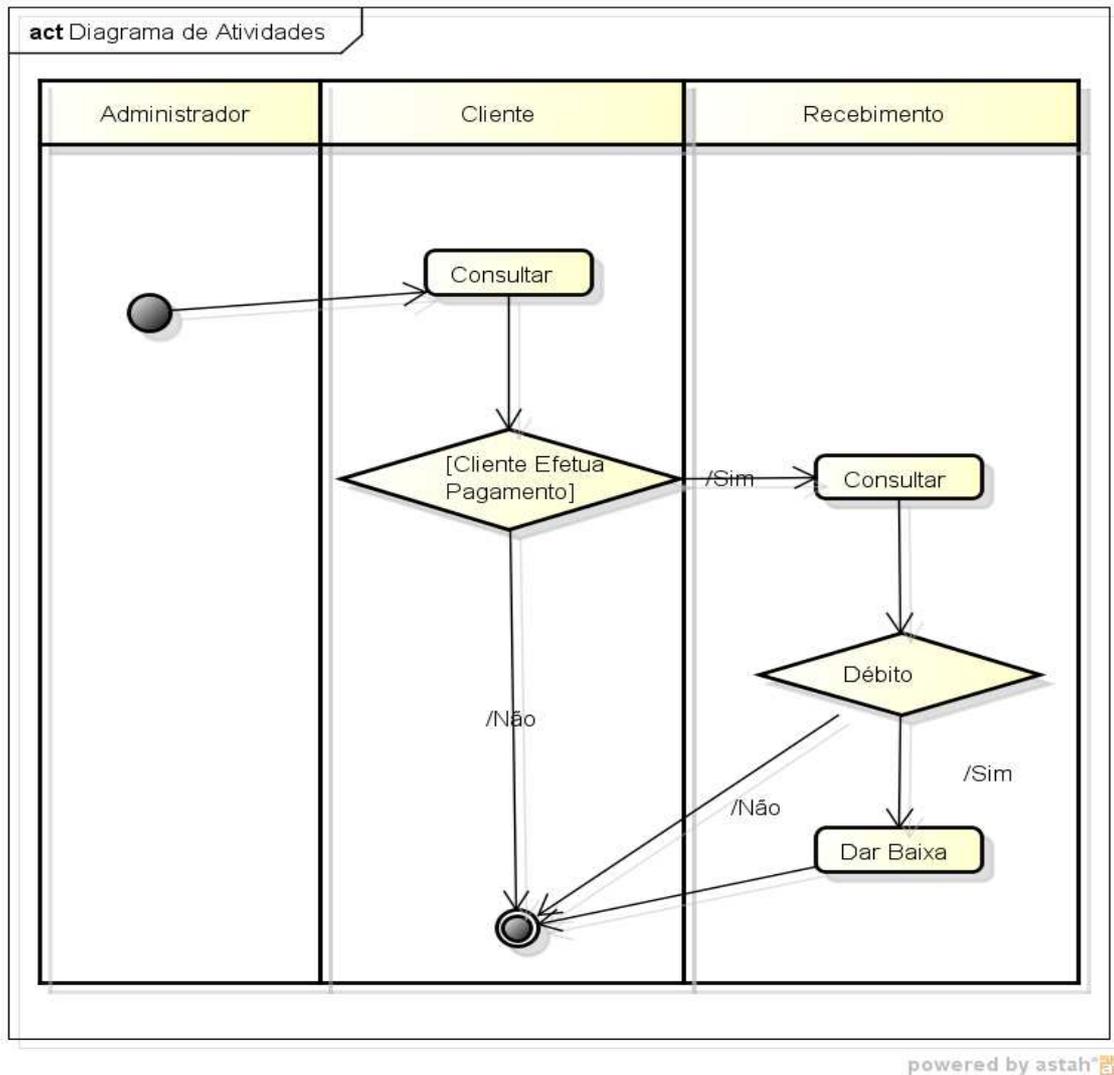


Figura 12 - Diagrama de Atividades Controle de Recebimento

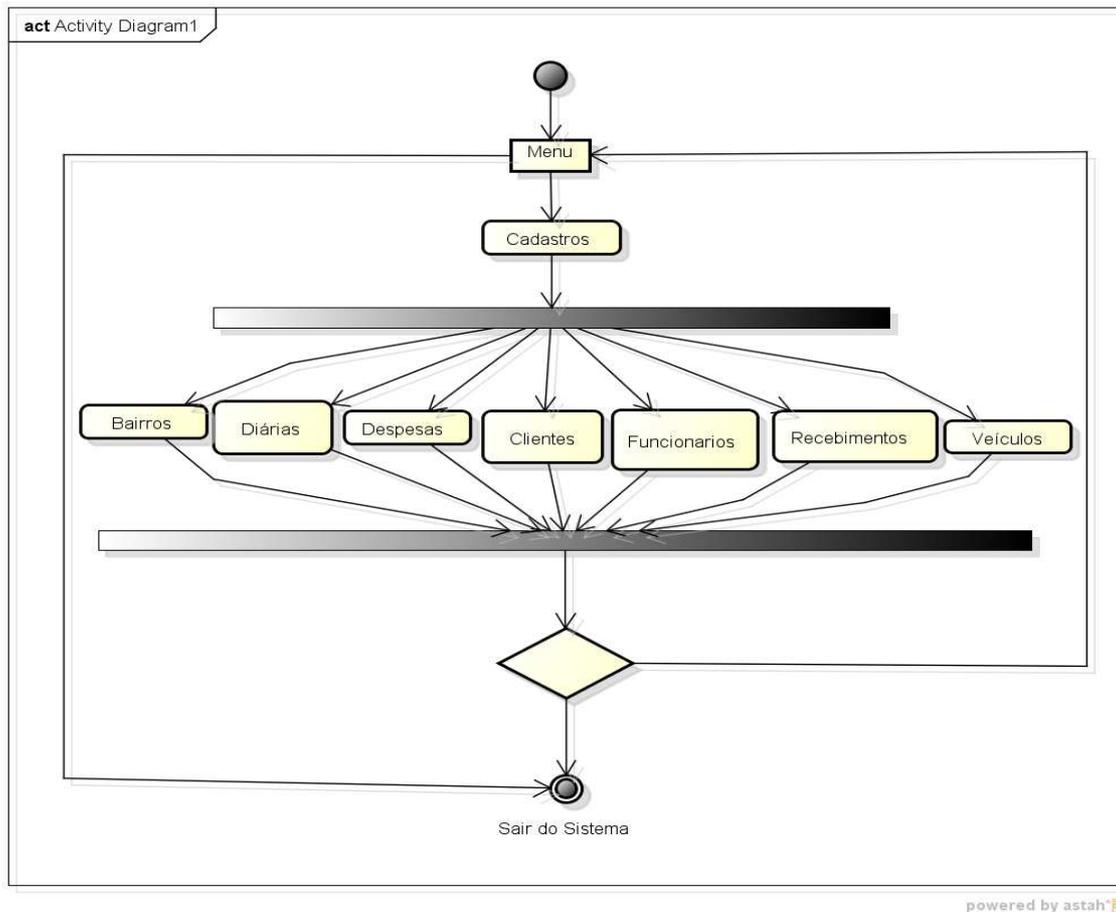


Figura 13 - Diagrama de Atividades Cadastros

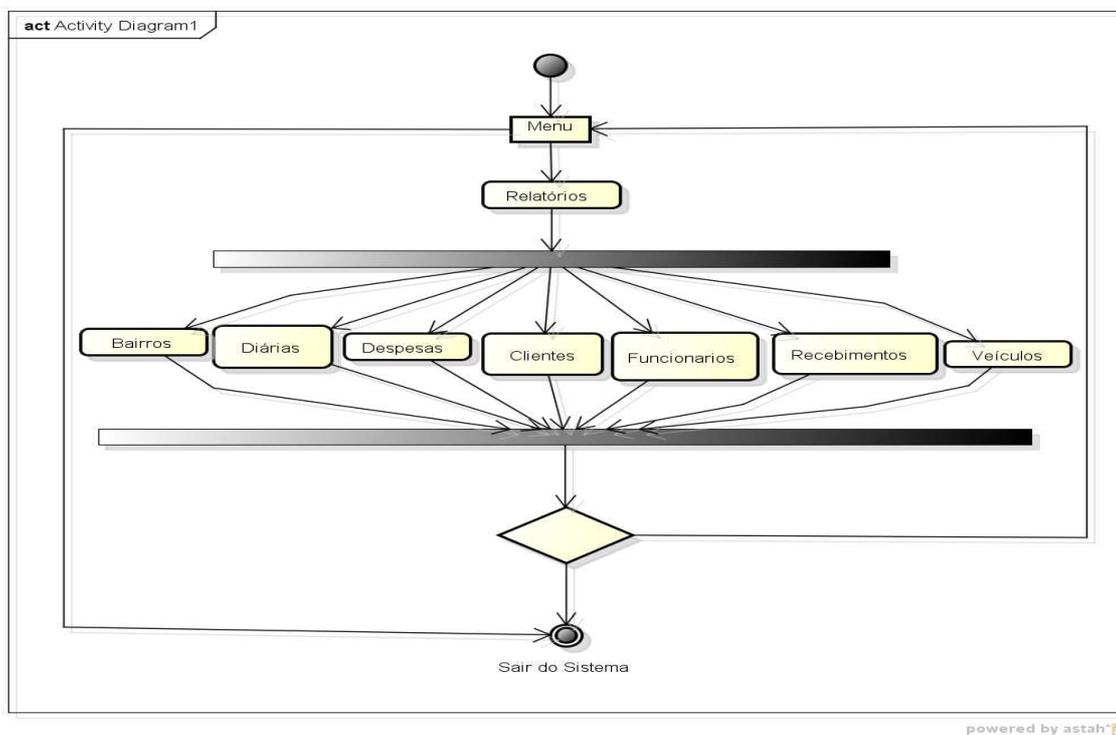


Figura 14 - Diagrama de Atividades Emitir Relatórios

3.7 - DIAGRAMA ENTIDADE RELACIONAMENTO

O Diagrama Entidade Relacionamento é baseado na percepção do mundo real que consiste em um conjunto de objetos básicos chamados entidades e nos relacionamentos entre estes objetos.

Ele foi desenvolvido para facilitar o projeto de banco de dados, permitindo a especificação de um esquema de negócio, onde tal esquema representa a estrutura lógica geral do banco de dados (REZENDE, 2005).

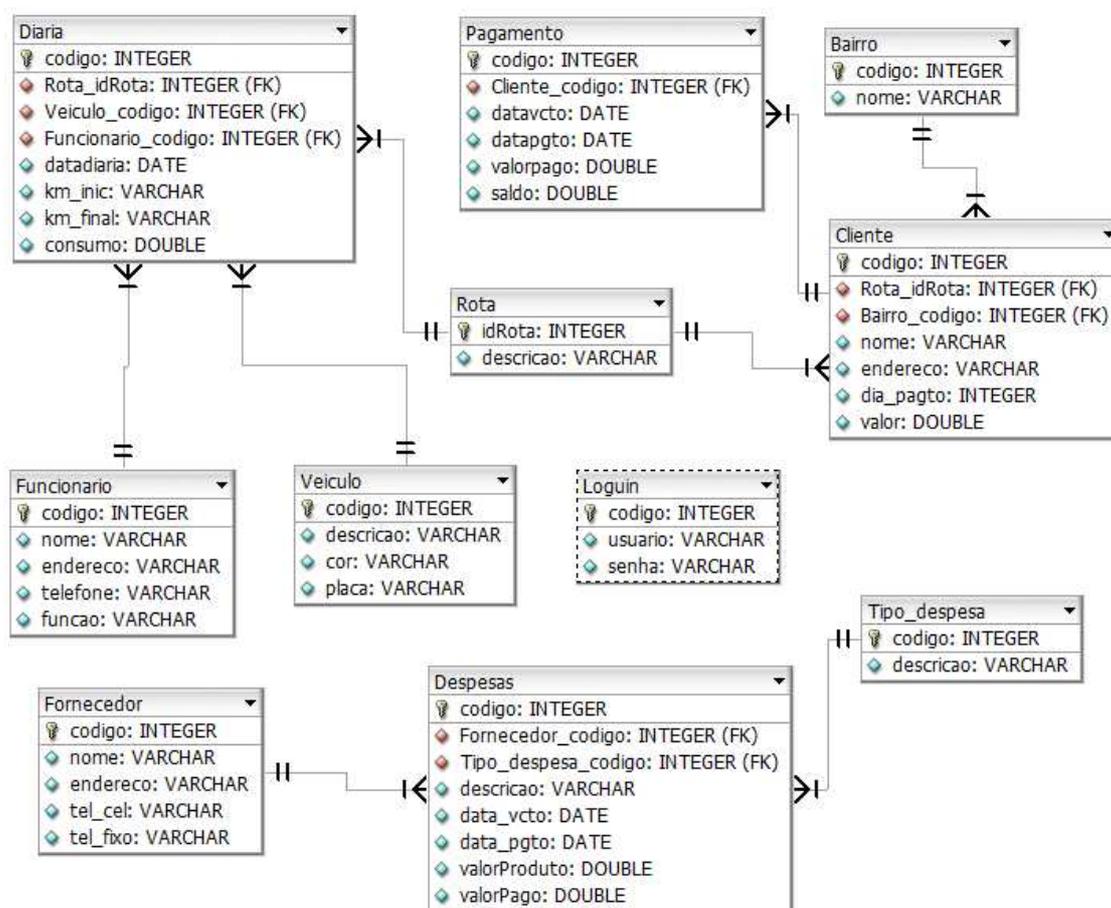


Figura 15 - Diagrama Entidade Relacionamento

4 - PROJETO

4.1 - WORK BREAKDOWN STRUCTURE

A *Work Breakdown Structure* (WBS) também conhecida como Estrutura Analítica de Trabalho é uma ferramenta de decomposição do trabalho do projeto em partes manejáveis. A WBS tem por objetivo indentificar elementos terminais (os produtos, serviços e resultados a serem feitos em um projeto). Assim, a WBS serve como base para a maior parte do planejamento de projeto. A ferramenta primária para descrever o escopo do projeto (trabalho) é a estrutura analítica do projeto WBS (<http://www.scribd.com/doc/22483180/Artigo-WBS>).

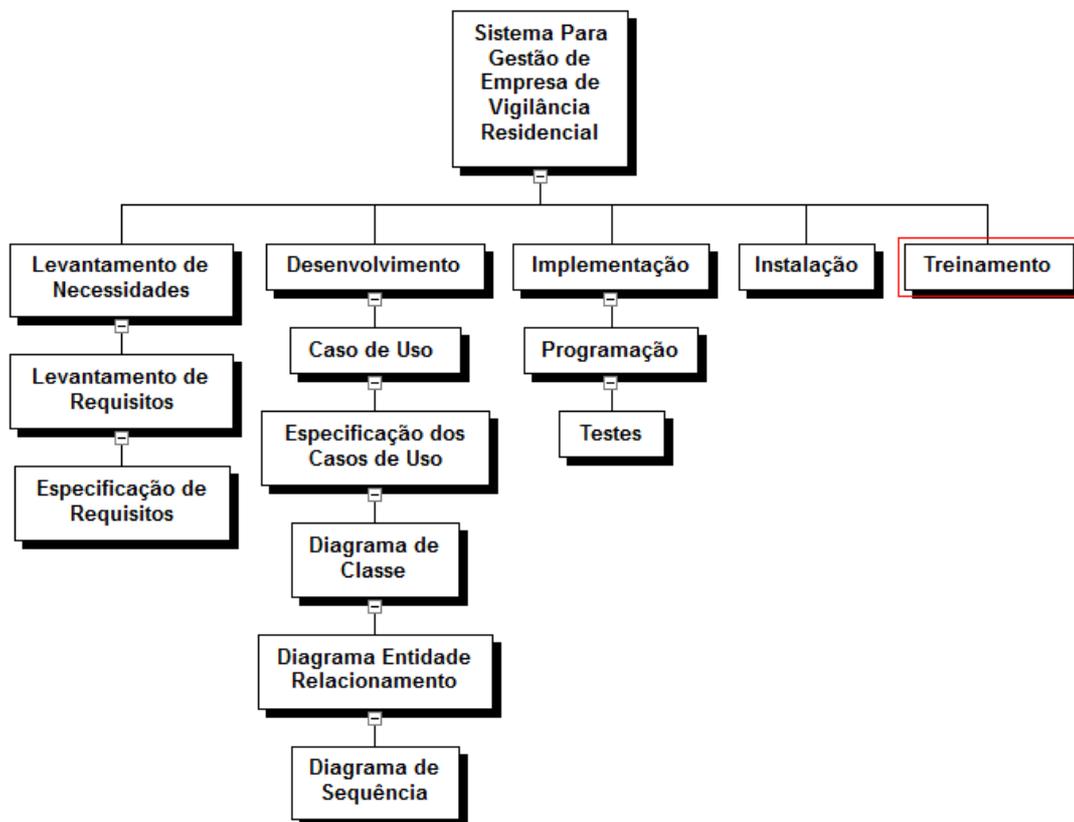


Figura 16 - Work Breakdown Structure

4.2 - SEQUENCIAMENTO DE ATIVIDADES

O diagrama ilustrado na Figura 1 descreve a seqüência das atividades e o tempo de duração para a realização de cada uma delas, a serem desenvolvidas no decorrer do Projeto.

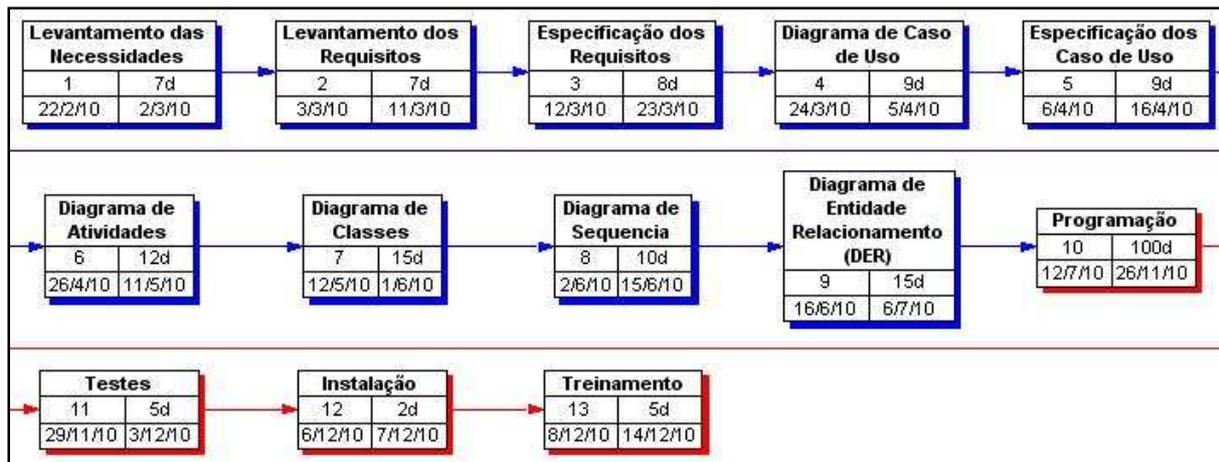


Figura 17 - Sequenciamento de Atividades

4.3 – ORÇAMENTO

4.3.1–RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO

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

- 01 Analista de Sistemas;
- 01 Programador;
- 01 Notebook Dell Core i3, 4 Giga de memória, HD de 500 Giga;
- 01 Multifuncional HP;
- Visual Studio 2010;
- SQL Server 2010;
- Crystal Report

Programador	Quantidade Hora	Custo/hora(R\$)	Total(R\$)
Ricardo Nunes	100	20,00	2.000,00
Custo Programador			2.000,00

Analista	Quantidade Horas	Custo/hora(R\$)	Total(R\$)
Ricardo Nunes	55	30,00	1.650,00
Custo Analista			1.650,00
Total			3.650,00

4.3.2 – CUSTO DE MATERIAIS

Notebook: R\$ 1.700,00 (mil e setecentos reais);

Depreciação 3 anos: $R\$ 1.700,00 / 36$ (meses) = R\$ 47,22 (quarenta e sete reais e vinte e dois centavos) por mês;

Custo de um dia: $R\$ 47,22 / 30$ (dias) = R\$ 1,57 (um real e cinquenta e sete centavos) por dia;

Custo de 167 dias: $R\$ 1,57 * 167 = R\$ 262,19$ (duzentos e sessenta e dois reais e dezenove centavos).

Multifuncional: R\$ 265,00 (duzentos e sessenta e cinco reais);

Depreciação 3 anos: $R\$ 265,00 / 36$ (meses) = R\$ 7,36 (sete reais e trinta e seis centavos) por mês;

Custo de um dia: $R\$ 7,36 / 30$ (dias) = R\$ 0,24 (vinte e quatro centavos) por dia;

Custo de 167 dias: $R\$ 0,24 * 167 = R\$ 40,97$ (quarenta reais e noventa e sete centavos).

Sistema Operacional Microsoft Windows Vista Basic: R\$ 329,00 (trezentos e vinte e nove reais);

Depreciação 2 anos: $R\$ 329,00 / 24$ (meses) = R\$ 13,78 (treze reais e setenta e oito centavos) por mês;

Custo de um dia: $R\$ 13,78 / 30$ (dias) = R\$ 0,45 (quarenta e cinco centavos) por dia;

Custo de 167 dias: $R\$ 0,45 * 167 = R\$ 76,31$ (setenta e seis reais e trinta e um centavos).

Visual Studio 2010: R\$ 890,00 (oitocentos e noventa reais);

Depreciação 3 anos: $R\$ 890,00 / 36$ (meses) = R\$ 24,72 (vinte e quatro reais e setenta e dois centavos) por mês;

Custo de um dia: R\$ 24,72 / 30 (dias) = R\$ 0,82 (oitenta e dois centavos) por dia;

Custo de 167 dias: R\$ 0,82 * 167 = R\$ 137,62 (cento e trinta e sete reais e sessenta e dois centavos).

Crystal Report: R\$ 877,04 (oitocentos e setenta e sete reais);

Depreciação 3 anos: R\$ 877,04 / 36 (meses) = 24,36 (vinte e quatro reais e trinta e seis centavos) por mês;

Custo de um dia: R\$ 24,36 / 30 (dias) = R\$ 0,81 (oitenta e um centavos) por dia;

Custo de 167 dias: R\$ 0,81 * 167 = R\$ 135,62 (cento e trinta e cinco e sessenta e dois centavos).

SQL Server 2008: R\$ 350,00 (trezentos e cinquenta reais);

Depreciação 3 anos: R\$ 350,00 / 36 (meses) = R\$ 9,72 (nove reais e setenta e dois centavos por mês);

Custo de um dia: R\$ 9,72 / 30 (dias) = R\$ 0,32 (trinta e dois centavos) por dia;

Custo de 142 dias: R\$ 0,32 * 142 = R\$ 45,44 (quarenta e cinco reais e quarenta e quatro centavos).

4.3.3 – Orçamento do Projeto

Recursos Utilizados	Valor
Analista – Programador	R\$ 3.650,00
Notebook	R\$ 262,19
Multifuncional	R\$ 40,97
Crystal Report	R\$ 135,62
Visual Studio 2010	R\$ 137,62
SQL Server 2008	R\$ 45,44
Windows Vista Basic	R\$ 76,31
TOTAL	R\$ 4.348,15

Tabela 9 – Orçamento

5 – IMPLEMENTAÇÃO DA APLICAÇÃO

Para a implementação da aplicação foi utilizado o ambiente de desenvolvimento MICROSOFT VISUAL STUDIO 2010 com a linguagem de programação C#.

<http://www.microsoft.com/visualstudio/pt-br/products/2010-editions/ultimate>

5.1 – ORGANIZAÇÃO DAS CAMADAS DE CÓDIGO FONTE.

Para uma melhor organização, o sistema foi implementado em camadas. A figura 18 apresenta as camadas: Models, Camada Dal, CamadaBLL e Camada de Visão.

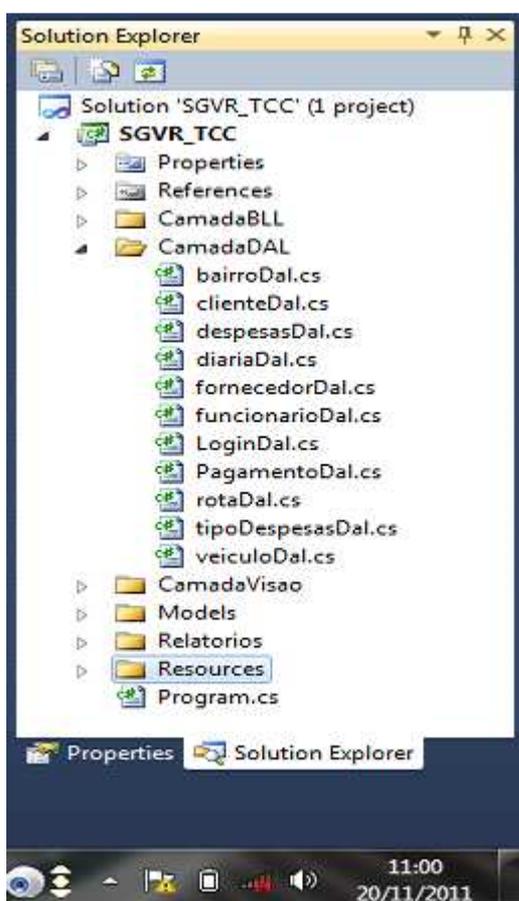


Figura 18 – Organização das Camadas

Models (Modelos): Contém os modelos do banco de dados onde as informações são obtidas ou gravadas por meio dos métodos gets e

sets(acessores e modificadores). É nesta camada que ficam as classes de modelagem do projeto, e as classes que fazem a persistência dos dados com o banco de dados.

Camada DAL (*Data Access Layer*): É nela que fica as classes e métodos responsáveis por realizar o acesso ao banco de dados, gravar, alterar, excluir e pesquisar as informações do banco de dados.

Camada BLL (*Business Logic Layer*): Esta camada contém as classes com as regras de serviços.

Camada UIL (*User Interface Layer*): Contém todo o desenho do software que o usuário irá utilizar, esta seria a ultima camada a ser implementada, pois é nela que ocorrerão as ligações, por exemplo, nela serão invocados os métodos da camada de serviço(BLL) que por sua vez vai instanciar classes da camada de acesso aos dados(DAL) que poderá ou não instanciar objetos pertencentes a classe dos modelos do banco(Model).

Relatórios: É nela que ficam guardados todos os desenhos, classes e os *crystal reports* responsáveis por gerenciar relatórios e recibos do sistema.

Resources (*Recursos*): É nela que ficam todas as imagens e ícones utilizados na aplicação.

5.2 – ENTENDENDO AS CAMADAS.

Na descrição acima foi citado o que é cada camada, agora vamos entender como funciona:

Assim como na figura abaixo, a camada de visão é a que o usuário vai utilizar(software), esta camada não poderá de forma nenhuma fazer contato direto com a camada de acesso a dados(DAL), para isso terá que chamar a camada de Negócio(BLL) para que o mesmo faça a chamada da DAL, pois somente esta camada poderá fazer acesso direto com o Banco. As camadas de modelos esta vinculada com todas as camadas, pois somente ela poderá pegar informações para estar passando para qual outra camada desejar.

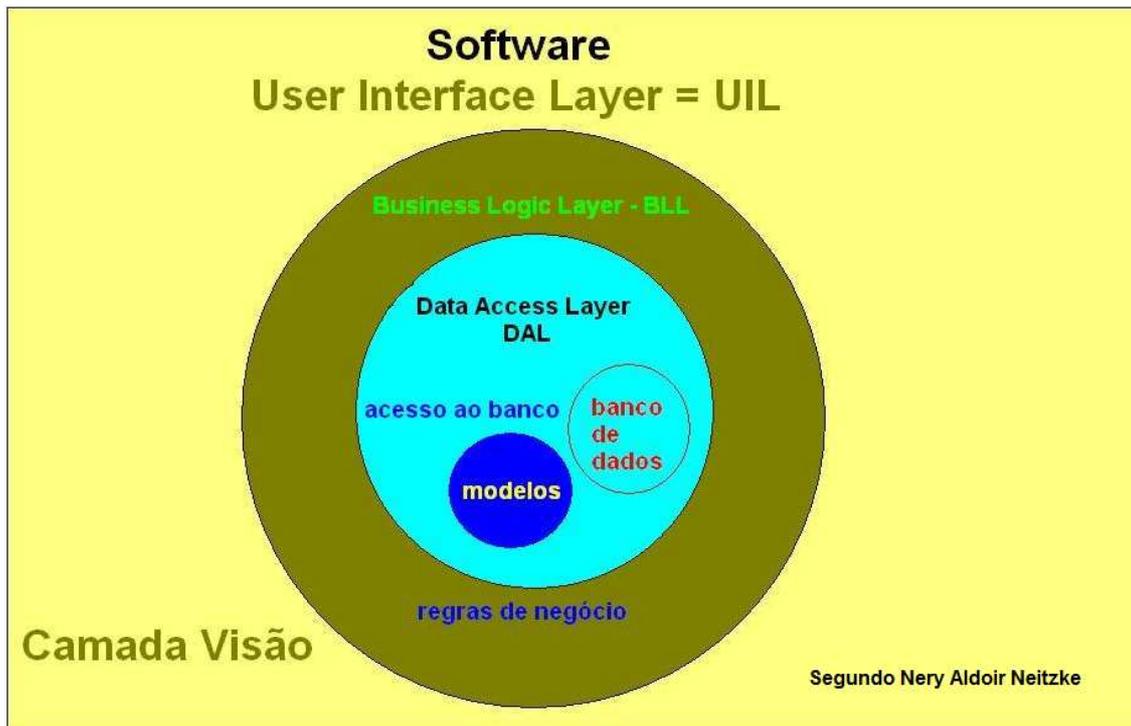


Figura 19 – Funcionamento das Camadas

5.3 – INTERFACE DO SISTEMA

Ao iniciar o sistema, o usuário deverá informar Login e Senha, conforme mostra a figura 20:



Figura 20 – Interface de Login

Após o usuário informar Login e Senha, o sistema então abrirá o menu principal contendo as seguintes opções: Consultar, Cadastrar, Relatórios, Emitir Recibos, Acesso, Sair. Conforme a Figura 21.



Figura 21 – Interface do Sistema

No menu Consultas abrirá as seguintes opções como mostra a Figura 22.



Figura 22 - Menu Consultar

O Usuário poderá escolher uma das opções mostradas, vamos supor que a opção *Cliente* foi escolhida. O Sistema automaticamente abrirá a janela para consulta dos Clientes, como mostra a figura 23.

ID	Nome	Endereço	ID Bairro	ID Rota	Valor	Dia Pgto	Telef. Celular	Telef. Fixo	Status
55	Antônio Pereira Mendezzz	Antonio de Araujo Marques, 170	65	1	12,00	5	() -	() -	1
56	Salvador Bomzacheiro	João Favareto	65	1	30,00	5	(018) 3341-3065	(018) 9785-1101	1
57	Evany	Santa Rita	1	1	10,00	5	() -	() -	1
60	Adir Teixeira	Santa Izabel, 81	1	1	10,00	5	() -	() -	1
61	Silvano dos Santos Alcantara	Santa Izabel, 91	1	1	10,00	5	() -	() -	1
64	José Everton de Souza	Misael Simões Gamido, 219	1	1	12,00	5	() -	() -	0
65	Igreja São Francisco	João Favareto, 446	1	1	15,00	5	() -	() -	1
66	Alvaro Manzini	Felísberto Simões Gamido, 197	46	6	12,00	5	() -	() -	1
67	Claudinei Camilo	Felísberto Simões Gamido, 259	46	6	15,00	5	() -	() -	1
68	William da Silva Moreira	Felísberto Simões Gamido, 121	54	6	15,00	5	() -	() -	0
69	Francisco de Almeida	Henrique José Pires, 399	54	6	15,00	5	(018) 9724-3276	() -	0
70	Antônio Germano	Henrique José Pires, 305	46	6	7,00	5	() -	() -	1

Figura 23 – Interface Consultar Clientes

O Usuário terá a opção de Ativar ou desativar o cliente ao invés de excluir, evitando assim de perder os dados do cliente.

Ao selecionar em um nome da lista os dados automaticamente serão preenchidos nos campos para edição. Estes campos são completamente desabilitados a não ser que o usuário clique em *habilitar campos*, que somente poderá ser habilitado se decidir editar algum dado do cliente.

As Pesquisas serão feitas por três tipos: por nome, endereço ou dia de pagamento, facilitando assim a busca realizada pelo usuário. Além desta pesquisa, o usuário poderá também selecionar tipos diferentes de clientes tais como: *Ativos*, *Inativos* ou *Todos*.

No menu *Cadastrar* abra as seguintes opções como mostra a figura 24.



Figura 24 – Interface Cadastrar

O Usuário poderá escolher uma das opções mostradas, vamos supor que a opção *Recebimentos* foi escolhida. O Sistema automaticamente abrirá a janela para controle de recebimento, como mostra a figura 25.

ID	ID Cliente	Data Vcto	Data Pgto	Valor Pago	Saldo
1446	83	05/10/2011		0,00	-10,00
1447	84	05/10/2011	22/11/2011	15,00	0,00
1448	85	05/10/2011		0,00	-12,00
1449	86	05/10/2011	22/11/2011	13,00	0,00
1450	87	05/10/2011	22/11/2011	13,00	0,00
1451	88	05/10/2011		0,00	-12,00
1452	89	05/10/2011	19/11/2011	8,00	0,00
1453	90	05/10/2011	22/11/2011	10,00	0,00
1454	91	05/10/2011		0,00	-15,00
1455	92	10/10/2011		0,00	-10,00
1456	93	10/10/2011	22/11/2011	12,00	12,00

Figura 25 – Interface Cadastrar Recebimentos

Nesta interface o usuário terá a opção de *Gerar Recebimento*, o sistema fará uma busca nos recebimentos cadastrados e se não houver recebimentos cadastrados para o devido mês ele listará todos os clientes na tabela *Recebimentos* como devedores do respectivo mês.

O usuário poderá efetuar o recebimento do cliente clicando em *Efetuar Pgto*, se esta for a opção o sistema se encarregará de zerar o saldo deste cliente.

As pesquisas são realizadas por quatro tipos: *ID Cliente*, *ID Pgto*, *Data de Pgto* e *Data de Vcto* assim como mostra a figura 25.

5.4 – DESCRIÇÕES DAS OPERAÇÕES DE MANUTENÇÃO DE DADOS

Neste capítulo, serão apresentadas as partes do código fonte para a elaboração das operações de manutenção de dados. Estas operações são também conhecidas pelo acrônimo CRUD, ou seja, rotinas de *Inserção*, *Alteração* e *Exclusão* que compõem os cadastros da aplicação, por exemplo: Clientes, Funcionários, Recebimentos etc.

As figuras 26, 27 e 28 mostra o código fonte de uma *Inserção*, *Exclusão* e *Alteração* da camada de acesso a dados *DAL*. Podemos observar que tal rotina inicialmente declara a operação SQL responsável pela inserção, exclusão e alteração dos dados no banco de dados. Na sequência, configura os comandos e as estruturas necessárias para a manipulação do banco de dados. Depois de iniciados os comandos, realiza a conexão com o banco de dados e executa a operação de inserção exclusão ou alteração desejada. Ao final, independentemente de ter ou não conseguido realizar a operação, a conexão com o banco de dados é fechada.

```
//-----
public void gravaCliente(ClienteModel cliente)
{
    try
    {
        conexao = new SqlConnection(conexao_sqlserver);
        SqlCommand sql = new SqlCommand("insert into Cliente values (@nome, @endereco,"
            + "@bairro_fk, @rota_fk, @valor, @dia_pagamento, "
            + "@telef_celular, @telef_residencial, "
            + "@statusCli)", conexao);

        sql.Parameters.AddWithValue("@nome", cliente.Nome);
        sql.Parameters.AddWithValue("@endereco", cliente.Endereco);
        sql.Parameters.AddWithValue("@bairro_fk", cliente.Bairro_FK);
        sql.Parameters.AddWithValue("@rota_fk", cliente.Rota_fk);
        sql.Parameters.AddWithValue("@valor", cliente.Valor);
        sql.Parameters.AddWithValue("@dia_pagamento", cliente.Dia_pagamento);
        sql.Parameters.AddWithValue("@telef_celular", cliente.Telef_celular);
        sql.Parameters.AddWithValue("@telef_residencial", cliente.Telef_residencial);
        sql.Parameters.AddWithValue("@statusCli", cliente.StatusCli);
        conexao.Open();
        sql.ExecuteNonQuery();
    }
    catch (Exception erro)
    {
        throw erro;
    }
    finally
    {
        conexao.Close();
    }
}
//-----
```

Figura 26 – Código para Inserir Dados

```
//-----
public void excluiCliente(ClienteModel cliente)
{
    try
    {
        conexao = new SqlConnection(conexao_sqlserver);
        SqlCommand sql = new SqlCommand("delete from Cliente where codigo = @codigo", conexao);
        sql.Parameters.AddWithValue("@codigo", cliente.Codigo);
        conexao.Open();
        sql.ExecuteNonQuery();
    }
    catch (Exception erro)
    {
        throw erro;
    }
    finally
    {
        conexao.Close();
    }
}
//-----
```

Figura 27 – Código para Excluir Dados

```
//-----
public void alteraCliente(ClienteModel cliente)
{
    try
    {
        conexao = new SqlConnection(conexao_sqlserver);
        SqlCommand sql = new SqlCommand("update Cliente set nome = @nome," +
            "endereco = @endereco, bairro_fk = @bairro_fk," +
            "rota_fk = @rota_fk," +
            "valor =@valor, dia_pagamento = @dia_pagamento," +
            "telef_celular = @telef_celular," +
            "telef_residencial = @telef_residencial," +
            "statusCli = @statusCli where codigo = @codigo", conexao);

        sql.Parameters.AddWithValue("@codigo", cliente.Codigo);
        sql.Parameters.AddWithValue("@nome", cliente.Nome);
        sql.Parameters.AddWithValue("@endereco", cliente.Endereco);
        sql.Parameters.AddWithValue("@bairro_fk", cliente.Bairro_FK);
        sql.Parameters.AddWithValue("@rota_fk", cliente.Rota_fk);
        sql.Parameters.AddWithValue("@valor", cliente.Valor);
        sql.Parameters.AddWithValue("@dia_pagamento", cliente.Dia_pagamento);
        sql.Parameters.AddWithValue("@telef_celular", cliente.Telef_celular);
        sql.Parameters.AddWithValue("@telef_residencial", cliente.Telef_residencial);
        sql.Parameters.AddWithValue("@statusCli", cliente.StatusCli);
        conexao.Open();
        sql.ExecuteNonQuery();
    }
    catch (Exception erro)
    {
        throw erro;
    }
    finally
    {
        conexao.Close();
    }
}

```

class System.String
Represents text as a series of Unicode characters.

Figura 28 – Código para Alterar Dados

CONCLUSÃO

Com a utilização do sistema pela Empresa, a base de dados que compõe o mesmo pode ajudar na manutenção diária dos serviços utilizados, sendo que o gerenciamento do controle de recebimentos será feito automaticamente, listando todos os clientes que já efetuaram ou não o pagamento do mês, podendo ser feita uma consulta em meses em atrasos entre outros.

A informatização da empresa ajudará a melhorar os serviços realizados pelos vigilantes, sem contar que o administrador vai estar mais atualizado sobre os serviços dos vigilantes, os dados terão mais segurança e agilidade nas informações.

Por meio de entrevistas realizou-se todo o levantamento de requisitos e funcionalidades para o sistema e depois de realizada essa etapas foram feitas as modelagens do sistema, como: caso de uso, diagrama de classe, e diagrama de sequência. Tais diagramas foram muito importantes para o desenvolvimento do sistema, pois permitiu ter uma visão completa sobre o sistema e o que deveria ser feito para dar continuidade ao mesmo, sem a modelagem seria difícil efetuar o desenvolvimento do sistema.

O conceito MVC (Modelo-Visão-Controle) também foi de suma importância para o desenvolvimento da aplicação, pois este padrão permitiu separar em camadas cada uma de suas responsabilidades.

Durante o desenvolvimento do sistema, existiram dificuldades, tais dificuldades foram sanadas por meio de conversas com o orientador.

As tecnologias empregadas no trabalho mostraram - ser eficientes no desenvolvimento de aplicações. A aplicação será muito útil, pois poderá oferecer todas as informações sobre os funcionários, diárias, controle de recebimentos, emissão de relatórios e recibos, já que o mesmo era feito de forma manual.

O Microsoft SQL Server 2008 é um gerenciador de banco de dados com um diferencial, totalmente integrado a ferramenta Visual Studio, permitindo maior produtividade no desenvolvimento de aplicações atendendo as necessidades desejadas.

Para desenvolvimentos futuros, pretende-se implantar um sistema voltado para web, já que mantém uma maior facilidade de acesso, pois as informações poderão ser consultadas a longa distância e os clientes poderão registrar reclamações de onde estiver, e ainda será possível registrar ocorrências e fazer um mapeamento de todo território onde os serviços estão sendo realizados.

REFERÊNCIAS

BOOCH, Grady; RUMBAUCGH James; JACOBSON Ivar; Uml Guia do Usuário. Tradução de Fábio Freitas da Silva. Rio de Janeiro: Editora Campus, 2000.

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.

MILANI, André; Sql Server guia do programador. São Paulo: Editora Novatec, 2007.

REZENDE, Denis Alcides; Engenharia de Software e Sistema de informação. Rio de Janeiro: Brasport, 2005.

VARGAS, Ricardo; Gerenciamento de Projetos Estabelecendo Diferenciais Competitivos. 6º Edição. São Paulo: Editora BRASPORT Livros e Multimídias, 2005.

Video Aula: NEITZKE, Neri Aldoir. **Aprenda a Desenvolver Sistemas Completos**. Carazinho, RS. 2007.

<http://social.msdn.microsoft.com/Forums/pt-br/categories/> > Acesso em 11, 12, 13 de Agosto e 22, 23,24, 28, e 29 de Outubro de 2011

ANEXO – CRONOGRAMA

O cronograma abaixo se refere ao fluxo de tarefas cumpridas.

Tarefas	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV
Levantamento de necessidades										
Levantamento de Requisitos										
Especificação dos Requisitos										
Diagrama de Casos de Uso										
Especificação de Casos de Uso										
Diagrama de Classes										
Diagrama de Sequenciamento de Atividades										
Diagrama de Atividades										
Diagrama de Implementação										
Diagrama de Entidade Relacionamento										
Elaboração da Qualificação										
Qualificação										
Programação										
Testes										
Escrita da Versão final do Projeto										
Apresentação do Projeto										

Figura 29 - Cronograma