



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

LUIS PAULO BATISTA

**SISTEMA PARA CONTROLE NA PRESTAÇÃO DE SERVIÇOS DE
MANUTENÇÃO EM OFICINA MECÂNICA**

**Assis
2013**

LUIS PAULO BATISTA

SISTEMA PARA CONTROLE NA PRESTAÇÃO DE SERVIÇOS DE
MANUTENÇÃO EM OFICINA MECÂNICA

Trabalho de Conclusão de Curso apresentado ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA como requisito parcial à obtenção do Certificado de Conclusão.

Orientador: Prof. Ms. Felipe A. C. Pazinato

Área de Concentração: Análise e Desenvolvimento de Sistemas

Assis
2013

FICHA CATALOGRÁFICA

Batista, Luis Paulo.

Sistema para Controle na Prestação de Serviços de Manutenção em Oficina Mecânica, Luis Paulo Batista. Fundação Educacional do Município de Assis - FEMA - Assis -2013.

Número de Páginas 63p.

Orientador: Professor Ms. Felipe A. C. Pazinato.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis

1.Sistema de Oficina Mecânica 2. Cadastros 3.Ordem de Serviços

CDD: 001.61

Biblioteca Fema

SISTEMA PARA CONTROLE NA PRESTAÇÃO DE SERVIÇOS DE MANUTENÇÃO EM OFICINA MECÂNICA

LUIS PAULO BATISTA

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: _____

Analisador(1): _____

AGRADECIMENTOS

Primeiramente de tudo agradeço a DEUS por ter me dado forças, me sustentado e vigiado me dando sabedoria e equilíbrio emocional, por tudo que passei.

Segundo a minha família, meus pais Amarildo Batista Filho e Solange Cristina de Almeida, ao meu irmão Gustavo Augusto Batista, a todos meus tios que sempre admiraram meus esforços, e também aos meus amigos que estiveram em todas as horas de dificuldades e alegrias Silvia, Mayara, e todos que torceram em especial Jéssica, Willian, Franciele, Mariana e Maysa.

Aos meus amigos de graduação do curso, pelo convívio, compreensão e estudos.

A minha amiga Edlaine Santos que sempre me ajudou nos momentos de dúvidas.

Em terceiro ao meu orientador Felipe Pazinato, pela paciência, confiança e por acreditar em meu potencial. E também o constante estímulo transmitido durante o trabalho.

A todos os professores que fizeram parte do curso e contribuíram para o nosso futuro profissional e pessoal.

"Podemos escolher recuar em direção à segurança ou avançar em direção ao crescimento. A opção pelo crescimento tem que ser feita repetidas vezes. E o medo tem que ser superado a cada momento."

Abraham Maslow

(1908-1970)

RESUMO

O sistema tem como objetivo, gerar cadastro, movimentar entrada e saída de veículos, movimentar orçamentos e gerar relatórios. Software busca facilitar o usuário realizar suas tarefas rotineiras, como cadastrar clientes, veículos, mão de obra mecânica, abrir ordens de serviços, criar orçamentos e gerar relatórios. Ajudar o dono a ter maior agilidade no atendimento dos clientes, facilitando o serviço do usuário.

Palavra chave: C#, Visual Studio 2010, SqlServer 2008.

ABSTRACT

The system aims to generate entries, control the transactions informations about vehicles, creating business demonstratives graphics and generate reports. Software aims to facilitate the user to perform their routine tasks, such as registering customers, vehicles, mechanical work force, to open service orders, to create estimates and to generate reports. Helping the owner to have greater flexibility in serving customers, facilitating service user.

Keyword: C #, Visual Studio 2010, SqlServer 2008.

SUMÁRIO

1. INTRODUÇÃO	13
1.1 OBJETIVO	14
1.2 PÚBLICOS ALVO.....	14
1.3 JUSTIFICATIVA	14
1.4 MOTIVAÇÃO	15
1.5 PERSPECTIVAS DE CONTRIBUIÇÃO	15
1.6 RECURSOS NECESSÁRIOS	15
2. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA	16
2.1 METODOLOGIAS DE ANÁLISE E DESENVOLVIMENTO.....	17
2.1.1. DBDESIGNER 4	17
2.1.2. MICROSOFT VISUAL C# 2010	17
2.1.3. BANCO DE DADOS SQL Server 2008	18
2.1.4. REPORTS VIEW.....	18
2.1.5. ASTAH COMMUNITY	18
3. LEVANTAMENTO DOS REQUISITOS	20
3.1 ENTREVISTAS COM O PROPRIETÁRIO	20
4. ANÁLISES DOS REQUISITOS.....	22
4.1 PROBLEMAS POTENCIAIS	22
4.2 PRIORIZAÇÃO DA IMPLEMENTAÇÃO DOS REQUISITOS.....	22
4.3 REQUISITOS FUNCIONAIS	22
4.4 VALIDAÇÕES DOS REQUISITOS	23
5. ESTIMATIVA DE CUSTOS	24
5.1 ESPECIFICAÇÕES DE CUSTOS	24
6. CRONOGRAMA.....	25
7. ESTRUTURA ANALITICA DO PROJETO	26
8. SEQUENCIAMENTO DE ATIVIDADES	27
9. MODELAGEM BANCO DE DADOS	28

9.1 DICIONÁRIOS DE DADOS	29
10. ANÁLISE ORIENTADA A OBJETO.....	35
10.1 USE CASE – ESCOPO DO SISTEMA.....	35
10.1.1. MANTER CLIENTE.....	36
10.1.2. MANTER MECÂNICO	37
10.1.3. MANTER VEÍCULO	38
10.1.4. MANTER ORDEM DE SERVIÇO	39
10.1.5. MANTER ORÇAMENTO	40
10.2 EMISSÃO DE RELATÓRIOS	41
10.2.1. EMITIR RELATÓRIO CLIENTE	41
10.2.2. EMITIR RELATÓRIO VEÍCULO POR CLIENTE.....	42
10.2.3. EMITIR RELATÓRIO CLIENTE DEVEDOR	43
10.2.4. EMITIR RELATÓRIO CONTA RECEBER POR DATA	44
10.2.5. EMITIR RELATÓRIO DE MECÂNICO	45
10.2.6. MANTER ORDEM DE SERVIÇO POR MECÂNICO	46
10.2.7. MANTER ORDEM DE SERVIÇO POR DATA	47
10.2.8. MANTER RELATÓRIO ORDEM DE SERVIÇO POR CÓDIGO.....	48
10.2.9. MANTER RELATÓRIO DE SERVIÇO EM ABERTO	49
11. DIAGRAMA DE CLASSES	50
12. DIAGRAMA DE SEQUÊNCIA.....	51
13. DIAGRAMA DE ATIVIDADE.....	52
14. CONCLUSÃO	53
15. ANEXOS	54
15.1 CONEXÕES DO BANCO DE DADOS	54
15.1.2. MENU	55
15.1.3. CADASTRO DE CLIENTE	55
15.1.4. ORÇAMENTO.....	56
15.1.5. ORDEM DE SERVIÇO	56
15.2 COMANDOS.....	57
15.2.1. COMANDO NOVO.....	57

15.2.2. COMANDO SALVAR	57
15.2.3. COMANDO ALTERAR.....	58
15.2.4. COMANDO APAGAR	58
15.2.5. COMANDO SAIR.....	58
15.3 DAL. CLIENTE.....	59
REFERÊNCIAS.....	61

LISTA DE ILUSTRAÇÕES

Figura 1: ESTRUTURA ANALITICA DO PROJETO (WBS).....	25
Figura 2: SEQUENCIAMENTO DE ATIVIDADES.....	26
Figura 3: MODELAGEM BANCO DE DADOS.....	27
Figura 4: USE CASE – ESCOPO DO SISTEMA.....	34
Figura 5: DIAGRAMA USE CASE – MANTER CLIENTE.....	35
Figura 6: DIAGRAMA USE CASE – MANTER MECÂNICO.....	36
Figura 7: DIAGRAMA USE CASE – MANTER VEÍCULO.....	37
Figura 8: DIAGRAMA USE CASE – MANTER ORDEM DE SERVIÇO.....	38
Figura 9: DIAGRAMA USE CASE – MANTER ORÇAMENTO.....	39
Figura 10: DIAGRAMA USE CASE – EMITIR RELATÓRIO CLIENTE.....	40
Figura 11: DIAGRAMA USE CASE – EMITIR RELATÓRIO VEÍCULO POR CLIENTE.....	41
Figura 12: DIAGRAMA USE CASE – EMITIR RELATÓRIO CLIENTE DEVEDOR.....	42
Figura 13: DIAGRAMA USE CASE – EMITIR RELATÓRIO CONTA RECEBER POR DATA.....	43
Figura 14: DIAGRAMA USE CASE – EMITIR RELATÓRIO MECÂNICO.....	44
Figura 15: DIAGRAMA USE CASE – EMITIR RELATÓRIO ORDEM DE SERVIÇO POR MECÂNICO.....	45
Figura 16: DIAGRAMA USE CASE – EMITIR RELATÓRIO ORDEM DE SERVIÇO POR DATA.....	46
Figura 17: DIAGRAMA USE CASE – EMITIR RELATÓRIO ORDEM DE SERVIÇO POR CÓDIGO.....	47
Figura 18: DIAGRAMA USE CASE – EMITIR RELATÓRIO ORDEM DE SERVIÇO EM ABERTO.....	48

Figura 19: DIAGRAMA DE CLASSES.....	49
Figura 20: DIAGRAMA SEQUÊNCIA.....	50
Figura 21: DIAGRAMA DE ATIVIDADE.....	51
Figura 22: Tela – MENU.....	53
Figura 23: Tela – CADASTRO DE CLIENTE.....	53
Figura 24: Tela – ORÇAMENTO.....	54
Figura 25: Tela – ORDEM DE SERVIÇO.....	54
Figura 26: Tela – COMANDO.....	55

1. INTRODUÇÃO

Atualmente, a Tecnologia da Informação está presente no dia a dia e nas organizações. Ela é usada como uma ferramenta eficiente de gestão da informação e de apoio às decisões, gerando um diferencial competitivo no mercado, sendo uma ferramenta que afeta interesses, valores e rotinas das pessoas. A competitividade e a sobrevivência das organizações dependem cada vez mais de sua capacidade de perceber as mudanças e antecipar-se às novas demandas, realinhando os investimentos em competências, tecnologias, produtos, serviços e mercados.

A nova economia mundial, baseada na Tecnologia de Informação, e que oferece recursos tecnológicos e computacionais para a geração de informações requer das organizações um conhecimento para coletar, trabalhar, interpretar e gerenciar este recurso. Os sistemas de informações cada vez mais sofisticados propõem mudanças no processo da estrutura do projeto.

O desenvolvimento e o domínio destas habilidades serão fundamentais para as organizações buscarem uma posição melhor no mercado em relação aos seus concorrentes, sendo necessário um engajamento na constante busca por inovação.

Com o crescimento das microempresas e dos profissionais autônomos, surgiu a ideia de desenvolver um projeto que ajude esses usuários a se organizarem em sua área de atuação, tendo um controle das movimentações financeiras.

1.1 OBJETIVO

O principal objetivo do sistema é o auxílio no controle de clientes e serviços aos profissionais do setor automotivo, fazendo cadastro de clientes, cadastro de automóveis, imprimindo orçamentos, abrindo ordens de serviço, e emitindo relatório com a descrição do serviço feito no carro.

1.2 PÚBLICO ALVO

Este sistema visa ajudar oficinas pequenas, microempresas e profissionais que prestam serviços de manutenção para automóveis.

1.3 JUSTIFICATIVA

Atualmente muitas microempresas e empresas de pequeno porte começam um pequeno negócio, sem se atentar às necessidades do mesmo. A intenção é desenvolver um sistema que facilitará o empreendedor a controlar sua empresa, onde o usuário possa gerenciar o fluxo de informações da sua oficina, e consiga assim atingir seus objetivos.

Com esse projeto o usuário terá facilidade em saber como está a situação de seus negócios, se há clientes devedores, a movimentação do caixa, imprimir orçamentos, atender as necessidades dos clientes, entre outras coisas. Portanto com esse software os usuários poderão administrar melhor seus custos e visualizar a taxa de crescimento da empresa.

1.4 MOTIVAÇÃO

Após acompanhar por muito tempo a necessidade de um usuário, e vendo a suas dificuldades em não conseguir manter a ordem nos seus estabelecimentos surgiu à proposta então de desenvolver um projeto que atenda as necessidades dos usuários e o ajude a se organizar na sua empresa.

1.5 PERSPECTIVAS DE CONTRIBUIÇÃO

Esse projeto visa auxiliar os usuário microempresa, podendo, trazer melhorias e organizando para que no futuro venha conquistar a mais clientes.

1.6 RECURSOS NECESSÁRIOS

Livros, pesquisas na internet e orientação docente. Para o desenvolvimento do sistema será necessário um computador com a ferramenta de desenvolvimento Visual Studio C# 2010 e um bando de dados (SQL Server 2008).

2. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA

Para a realização do projeto será utilizada as seguintes tarefas, visando à organização do trabalho e obtenção dos resultados desejados.

Estrutura:

- a) Levantamento dos Requisitos
- b) Análise dos Requisitos
- c) Validação dos Requisitos
- d) Especificação dos Requisitos
- e) Definição dos Objetivos
- f) Planejamento do projeto
- g) Diagrama de Caso de Uso
- h) Diagrama Entidade relacionamento
- i) Diagrama de Classes
- j) Diagrama de atividades
- k) Diagrama de sequencia
- l) Modelo E-R
- m) Programação
- n) Testes

2.1 METODOLOGIAS DE ANÁLISE E DESENVOLVIMENTO

Primeiramente foi feita a análise, verificando os requisitos do sistema. Os requisitos foram modelados na forma de Use Cases, utilizando a metodologia chamada UML. Logo após, um modelo entidade relacionamento foi construído para o banco de dados. Do modelo ER, partiu-se para o diagrama de classes, onde a estrutura de classes foi definida respeitando a programação em camadas, MVC (Model, View, Control). Do diagrama de classes, criou-se o diagrama de atividades, onde modela-se o comportamento do sistema.

2.1.1. DBDESIGNER 4

DBDesigner é uma ferramenta para projeto de banco de dados que integra a modelagem, projeto, implementação e manutenção em um mesmo ambiente.

Permite que se faça conexão com o banco de dados e desta ter acesso aos dados através das tabelas do modelo. Também é permitido o sincronismo do modelo com o banco de dados, isto é, toda alteração feita no modelo é aplicada no banco de dados sem necessidade de geração de Script.

2.1.2. MICROSOFT VISUAL C# 2010

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# é, de certa forma, a linguagem de programação que mais diretamente reflete a plataforma .NET sobre a qual todos os programas desta plataforma executam. C# está de tal forma ligada a esta plataforma que não existe o conceito de código não gerenciado em C#. Suas estruturas de dados primitivas são objetos que correspondem a tipos em .NET. Apesar de C# ser tido como similar a Java, existem uma série de diferenças importantes, mas a maioria é implementada de forma diferenciada em ambas as linguagens.

2.1.3. BANCO DE DADOS SQL Server 2008

Microsoft SQL Server é um servidor de banco de dados relacional , desenvolvido pela Microsoft : é um produto de software cuja principal função é armazenar e recuperar dados, conforme solicitado por outras aplicações de software, seja aqueles no mesmo computador ou aqueles em execução em outro computador através de uma rede (incluindo a Internet). Há pelo menos uma dúzia de diferentes edições do Microsoft SQL Server destinadas a públicos diferentes e para diferentes cargas de trabalho (que variam de pequenas aplicações que armazenam e recuperam dados no mesmo computador, para milhões de usuários e computadores que acessam grandes quantidades de dados a partir da Internet ao mesmo tempo).

2.1.4. REPORTS VIEW

Reports View é uma inteligência de negócios do aplicativo usado para criar e gerar relatórios a partir de uma ampla gama de fontes de dados . Vários outros aplicativos, incluindo Microsoft Visual Studio , ao mesmo tempo um pacote OEM versão do Reports View como uma ferramenta de propósito geral de relatórios. Reports View é um popular escritor de relatório , especialmente quando Microsoft pacote com Visual Studio versões 2003 a 2008. Reports View para Visual Studio 2010 ainda está disponível como add-on software.

2.1.5. ASTAH COMMUNITY

É uma ferramenta de modelagem UML gratuita, por ser uma versão Community possui algumas limitações. Há uma versão Professional disponível, mas os recursos presentes na versão Community podem suprir a necessidade de grande parte dos artefatos necessários no dia-a-dia.

Possui uma funcionalidade de gerar código a partir das classes definidas para o projeto, além de criar o DER.

Outra funcionalidade é a criação de comandos SQL para geração das tabelas existentes no diagrama.

A ferramenta Astah facilita muito a vida do programador, principalmente na documentação do projeto. Então para fazer a modelagem do sistema, foi utilizado a ferramenta Astah.

3. LEVANTAMENTO DOS REQUISITOS

O levantamento de requisitos será feito através de entrevista pelo analista ao cliente.

3.1 ENTREVISTAS COM O PROPRIETÁRIO

1) Qual o segmento da empresa?

R: A empresa atua no segmento de oficina mecânica especializada em manutenção de veículos.

2) O que o sistema devera controlar?

R: Cadastro de cliente, cadastro de veiculo, ordem de serviço, fluxo de caixa.

3) Quais dados serão necessários para manter:

a) Cliente

R: Código, nome, RG, CPF, idade, endereço, cidade.

b) Mecânico

R: Código, nome, RG, CPF, idade, endereço,

c) Ordem de serviço

R: Cod_funcionário, placa_ carro, quilometragem, chassis, nº ordem de serviço, data, dados do cliente.

d) Veículo

R: Cod_cliente, placa, nome, chassis, modelo, cor, motor.

4) Qual o objetivo da informatização da empresa?

R: Devido o crescimento da oficina, e a procura pelo serviço oferecido, foi preciso a criação de um sistema, onde venha controlar o faturamento e organizar a empresa.

5) Quais os problemas que deveram ser resolvidos após a informatização?

R: Controle de cliente, faturamento, o que foi feito no carro

6) Há algum controle atualmente? Se sim, como e qual?

R: Não

7) Quais as pessoas que utilizará o sistema?

R: O proprietário e funcionário

4. ANÁLISES DOS REQUISITOS

4.1 PROBLEMAS POTENCIAIS

- Controlar vendas por clientes
- Conhecer inadimplência de clientes
- Armazenar serviços por funcionários
- Gerar faturas de cliente com os devidos serviços
- Emitir relatórios de vendas por clientes.

4.2 PRIORIZAÇÃO DA IMPLEMENTAÇÃO DOS REQUISITOS

- Cadastro de clientes
- Cadastro de mecânico
- Cadastro de serviços
- Controlar serviços prestados
- Controle sobre pagamentos
- Controle de recebimento
- Gerar relatórios de serviço por cliente
- Gerar relatório com todas as faturas no mês
- Os relatórios deverão ser visualizados antes da impressão.

4.3 REQUISITOS FUNCIONAIS

- Manter cliente
- Manter ordem de serviço
- Manter funcionários
- Manter veículos
- Manter orçamento
- Movimentar contas a pagar
- Movimentar contas a receber

- Movimentar ordem de serviço
- Movimentar orçamento
- Emitir relatórios clientes
- Emitir relatórios veículo por cliente
- Emitir relatórios conta a receber por data
- Emitir relatórios conta a receber atrasado
- Emitir relatórios de ordem de serviço por mecânico
- Emitir relatórios de ordem de serviço por data
- Emitir relatórios de ordem de serviço por código
- Emitir relatório funcionários
- Emitir relatórios de orçamento em aberto

4.4 VALIDAÇÕES DOS REQUISITOS

1) Falta alguma informação?

R: Não

2) As informações levantadas atenderam o cliente?

R: Sim

3) Os requisitos são reais?

R: Sim

4) Há alguma duplicidade nas informações?

R: Não

5) O cliente quer acrescentar alguma informação que julgar importante?

R: No momento não

5. ESTIMATIVA DE CUSTOS

Pessoal: $100 * R\$ 25,00 = 2.500$

Computador: R\$ 1.600,00

Depreciação de 2 anos: $R\$1.600,00/24 = R\$66,67$ Mensal ou R\$2,22 Diário

Custo de utilização de $100 * 2,22 = R\$ 222,00$

Total de Custos do Sistema: R\$ 6.050,00

5.1 ESPECIFICAÇÕES DE CUSTOS

Recursos Necessários ao Desenvolvimento

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

Recursos Físicos

Numero de Pessoas: 01 Analista/Programador;

Equipamentos:

- 01 Microcomputador;
- 01 Impressora Jato de Tinta;

Softwares:

- Plataforma de Desenvolvimento: Microsoft Visual Studio 2010
- Linguagem: C#
- Modelagem: Astah Community
- Banco de Dados: Microsoft SQL Server.

7. ESTRUTURA ANALÍTICA DO PROJETO

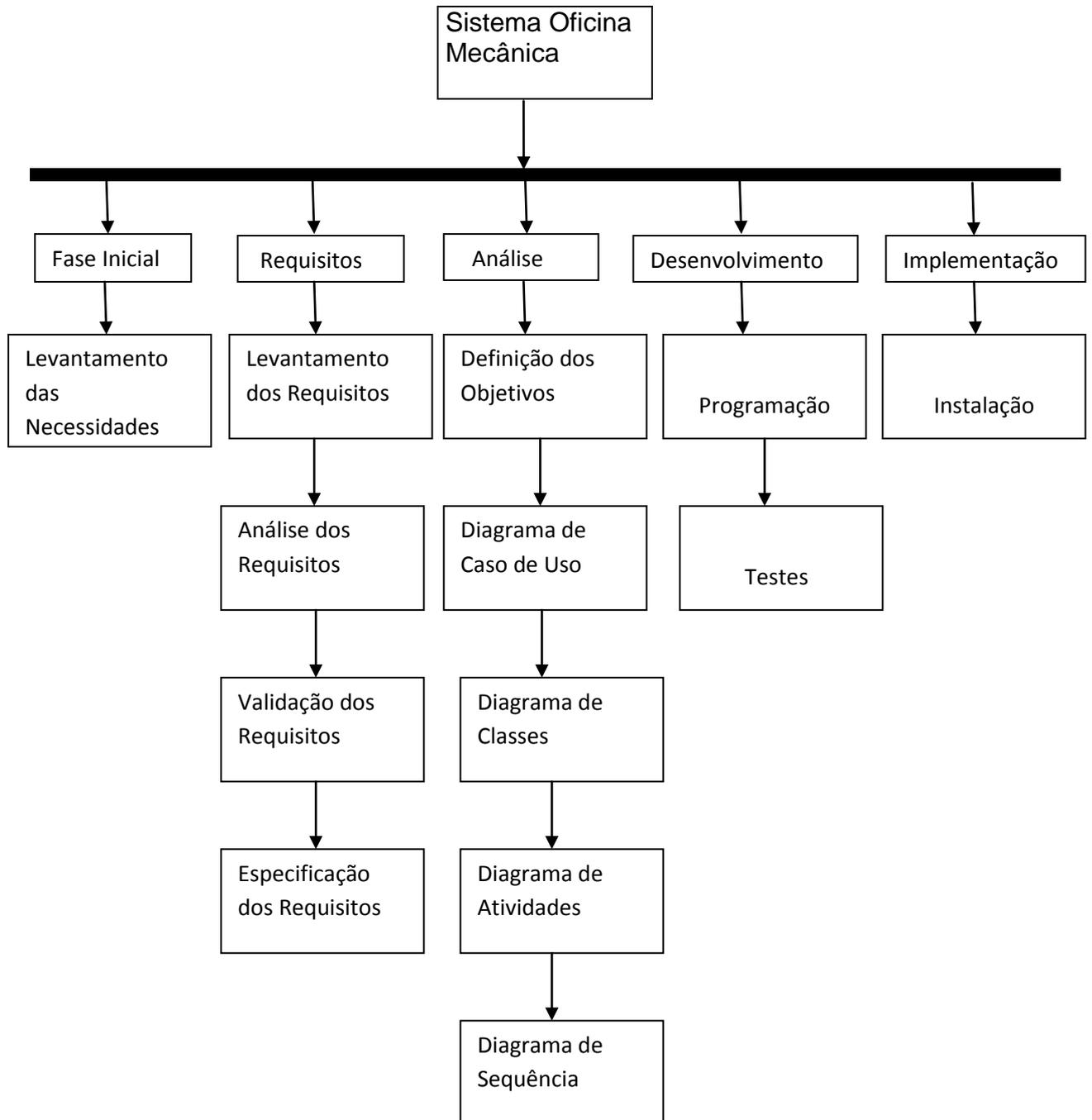


Figura 1- Estrutura Analítica (WBS)

8. SEQUENCIAMENTO DE ATIVIDADES

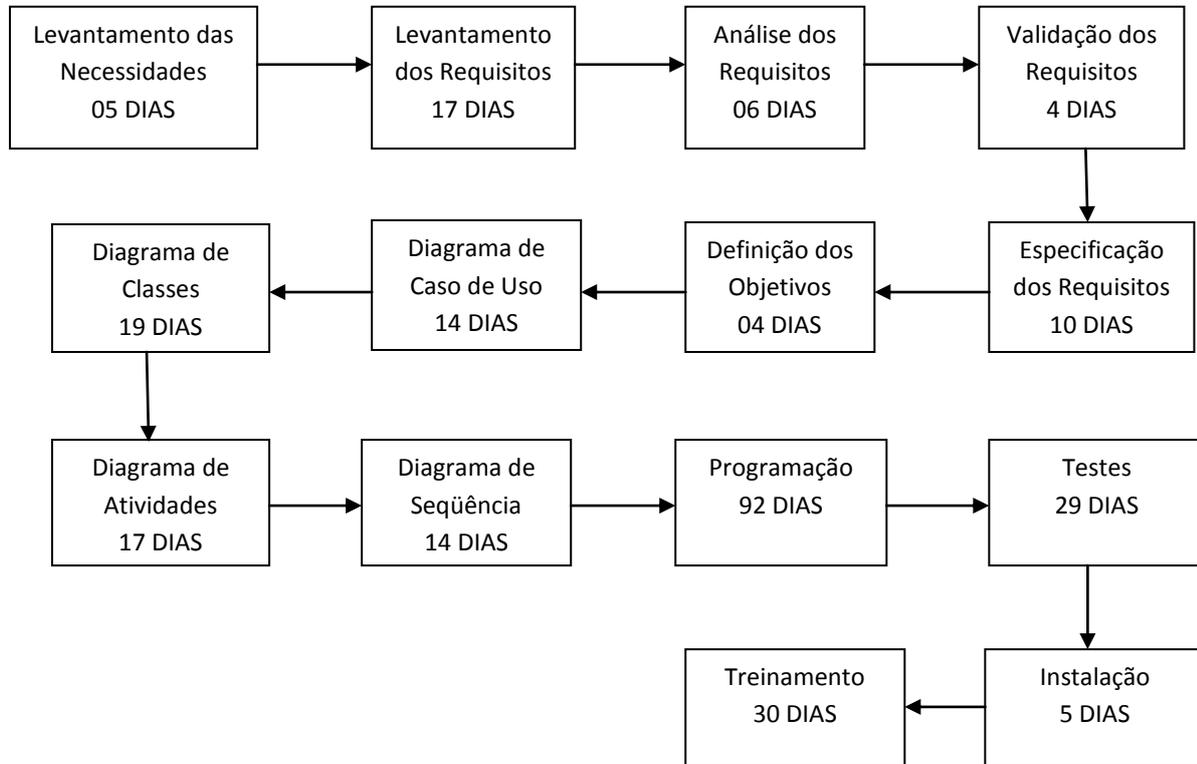


Figura 2- Senquêncimento de Atividades

9. MODELAGEM BANCO DE DADOS

9.1.1 MODELOS ENTIDADE RELACIONAMENTO

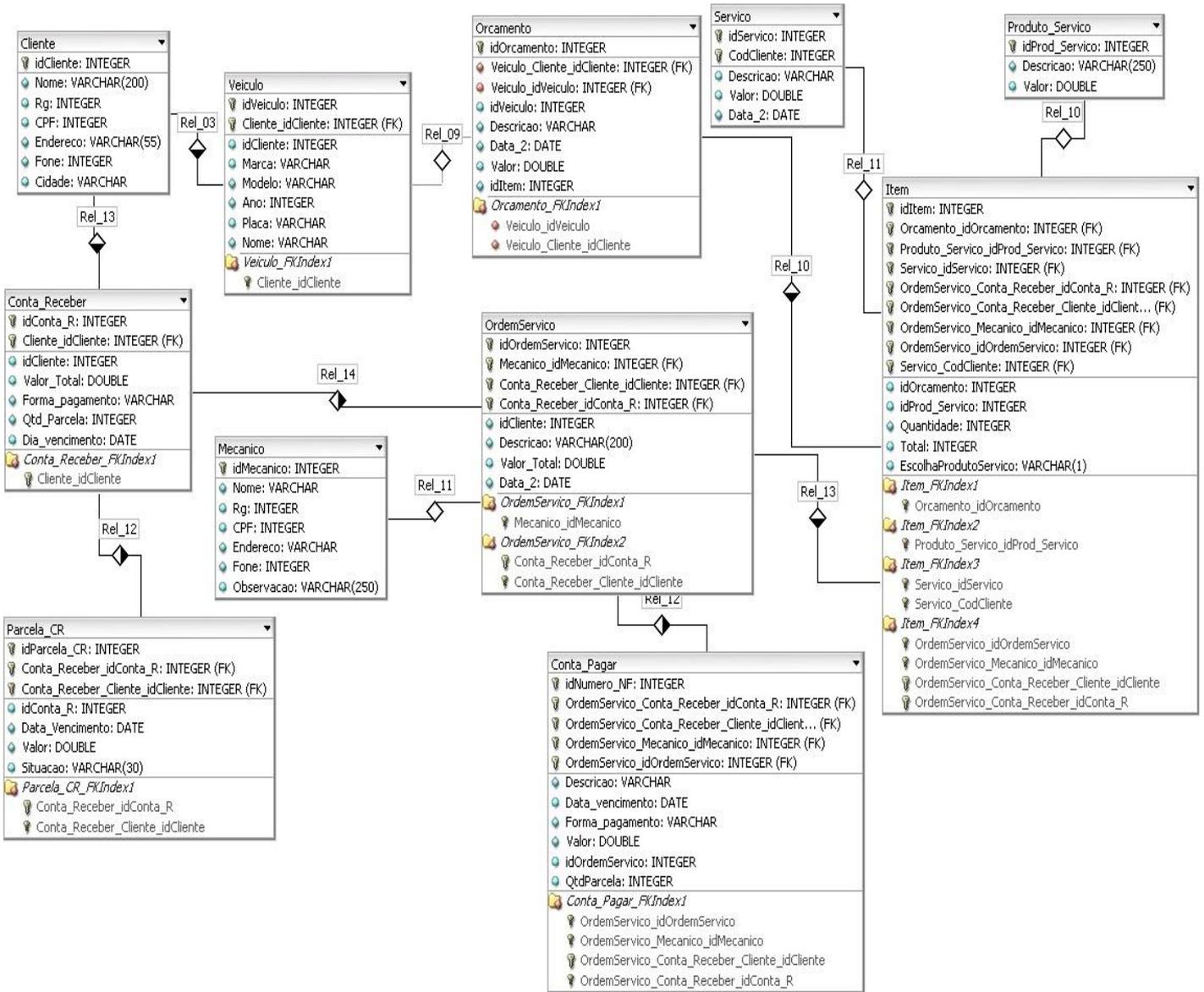


Figura 3 – Diagrama Entidade Relacionamento

9.1 DICIONÁRIOS DE DADOS

Tabela Cliente:

Campo	Descrição	Tamanho	Tipo
idCliente	Código do Cliente Chave Primária da Tabela Cliente	04	Numérico
Nome	Nome do Cliente	50	Texto
RG	RG do Cliente	10	Numérico
CPF	CPF do Cliente	14	Numérico
Endereço	Endereço do Cliente	50	Texto
Telefone	Telefone do Cliente	10	Numérico
Cidade	Cidade do Cliente	30	Texto

Tabela 1- Dicionário de Dados – Cliente

Tabela Veiculo:

Campo	Descrição	Tamanho	Tipo
idVeiculo	Código do Veiculo Chave Primária da Tabela Veiculo	04	Numérico
idCliente	Código do Cliente Chave Estrangeira da Tabela Cliente	04	Numérico
Marca	Marca do Veiculo	10	Texto
Modelo	Modelo Veiculo	15	Texto
Ano	Ano do Veiculo	04	Numérico
Placa	Placa do Veiculo	08	Texto/Numérico

Tabela 2- Dicionário de Dados - Veiculo

Tabela Orçamento

Campo	Descrição	Tamanho	Tipo
idOrçamento	Código do Orçamento Chave Primária da Tabela Orçamento	05	Numérico
idVeiculo	Código do Veiculo Chave Estrangeira da Tabela Veiculo	04	Numérico
Descrição	Descrição do Orçamento	250	Texto
Data	Data do Orçamento	08	Data
Valor	Valor do Orçamento	08	Numérico
idItem	Código do Item Chave Estrangeira da Tabela Item	04	Numérico

Tabela 3- Dicionário de Dados – Orçamento

Tabela Serviço

Campo	Descrição	Tamanho	Tipo
idServiço	Código do Serviço Chave Primária da Tabela Serviço	04	Numérico
Descrição	Descrição do Serviço	250	Texto
Valor	Valor do Serviço	08	Numérico

Tabela 4- Dicionário de Dados Serviço

Tabela Produto Serviço

Campo	Descrição	Tamanho	Tipo
idServiço	Código do Produto_Serviço Chave Primária da Tabela Produto_Serviço	04	Numérico
Descrição	Descrição do Produto_Serviço	250	Texto
Valor	Valor Produto_Serviço	08	Texto

Tabela 5- Dicionário de Dados - Produto Serviço

Tabela Item

Campo	Descrição	Tamanho	Tipo
idItem	Código do Item Chave Primária da Tabela Item	04	Numérico
idOrçamento	Código de Orçamento Chave Estrangeira da Tabela Orçamento	04	Numérico
idProd_Orçamento	Código do Produto_Serviço Chave Estrangeira da Tabela Item	04	Numérico
Quantidade	Quantidade do Item	04	Numérico
Total	Total do Item	04	Numérico
ESP	ESP do Item	50	Texto

Tabela 6- Dicionário de Dados - Item

Ordem Serviço

Campo	Descrição	Tamanho	Tipo
idOrdemServiço	Código da Ordem Serviço Chave Primária da Tabela OrdemServiço	04	Numérico
idCliente	Código do Cliente Chave Estrangeira da Tabela Cliente	04	Numérico
Descrição	Descrição da OrdemServiço	250	Texto
Valor	Valor da OrdemServiço	08	Numérico

Tabela 7- Dicionário de Dados – Ordem Serviço

Tabela Mecânico

Campo	Descrição	Tamanho	Tipo
idMecânico	Código do Mecânico Chave Primária da Tabela Mecânico	04	Numérico
Nome	Nome do Mecânico	50	Texto
RG	RG do Mecânico	10	Numérico
CPF	CPF do Mecânico	14	Numérico
Endereço	Endereço do Mecânico	50	Texto
Telefone	Telefone do Mecânico	10	Numérico
Observação	Observação do Mecânico	50	Texto

Tabela 8- Dicionário de Dados – Mecânico

Tabela Conta Receber

Campo	Descrição	Tamanho	Tipo
idConta_R	Código da Conta Chave Primária da Tabela Conta Receber	04	Numérico
idCliente	Código do Cliente Chave Estrangeira da Tabela Cliente	04	Numérico
Valor	Valor da Conta Receber	08	Numérico
Forma_Pagamento	Forma de Pagamento da Conta Receber	10	Texto
Qtd_Parcela	Qtd Parcela da Conta Receber	02	Numérico
Dia_Vencimento	Dia Vencimento da Conta Receber	08	Data

Tabela 9- Dicionário de Dados – Conta Receber

Tabela Parcela Conta Receber

Campo	Descrição	Tamanho	Tipo
idParcela	Código da Parcela Chave Primaria da Tabela Parcela_CR	04	Numérico
Data_Vencimento	Data do Vencimento da Parcela	08	Data
Valor	Valor da Parcela	08	Numérico
Situação	Situação da Parcela	50	Texto

Tabela 10- Dicionário de Dados Parcela Conta Receber

Tabela Conta Pagar

Campo	Descrição	Tamanho	Tipo
idNúmero_NF	Código da Conta Chave Primária da Tabela Conta_Pagar	04	Numérico
Descrição	Descrição da Conta	25	Texto
Data_Vencimento	Vencimento da Conta	08	Numérico
Forma_Pagamento	Forma Pagamento da Conta	10	Texto
Valor	Valor da Conta	08	Numérico
idOrdemServiço	Código da Ordem Serviço Chave Estrangeira da Tabela OrdemServiço	04	Numérico

Tabela 11- Dicionário de Dados Conta Pagar

Tabela Parcela Conta Pagar

Campo	Descrição	Tamanho	Tipo
idParcela	Código da Parcela Chave Primária da Tabela Parcela_CP	04	Numérico
idNúmero_NF	Código Nota Fiscal Chave Estrangeira da Tabela Conta Pagar	04	Numérico
Qtd_Parcela	Quantidade de Parcela	02	Numérico

Tabela 12- Dicionário de Dados Parcela Conta Pagar

10. ANALISE ORIENTADA A OBJETO

10.1 USE CASE – ESCOPO DO SISTEMA

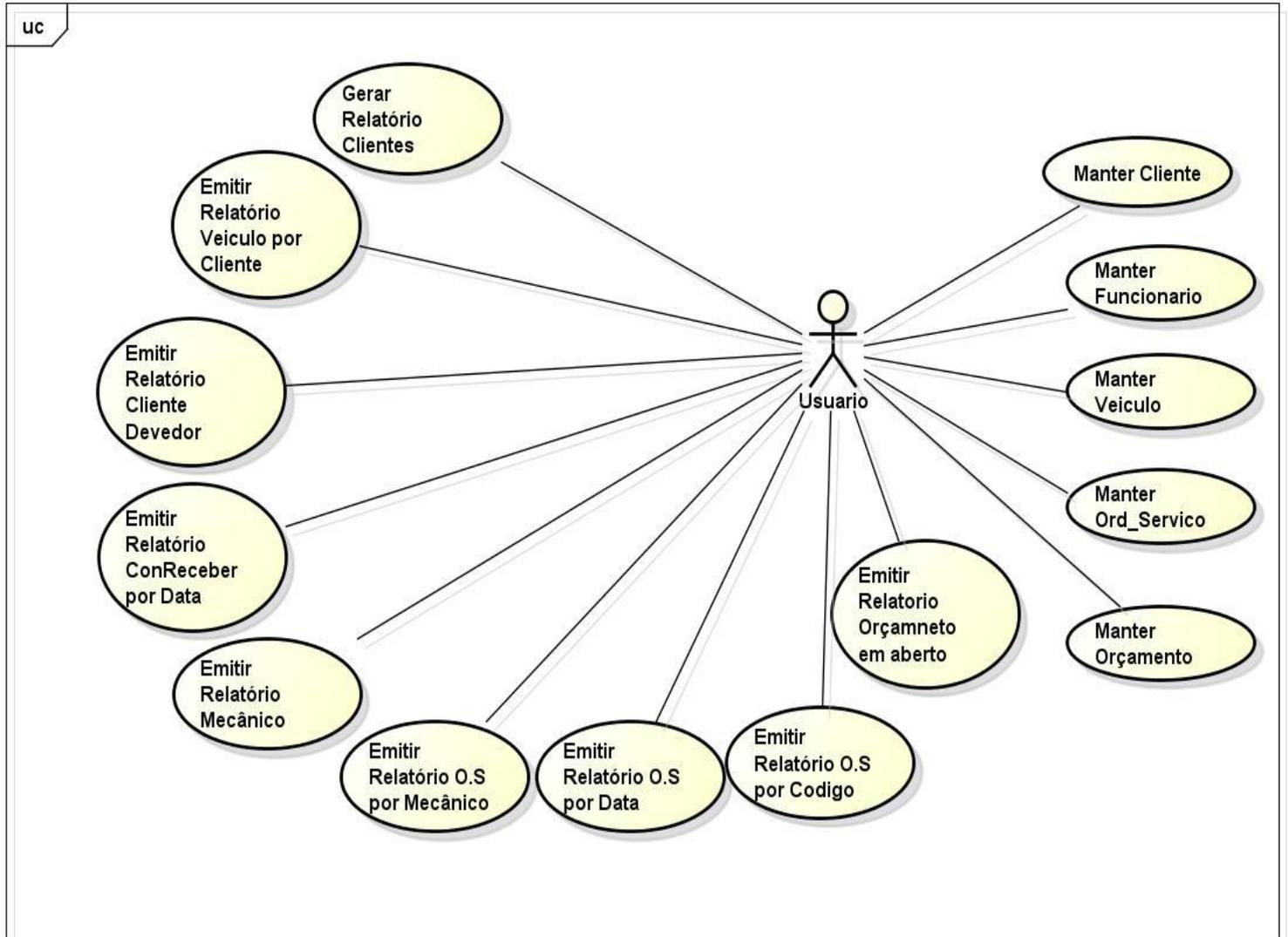


Figura 4- Use Case Escopo Do Sistema

10.1.1. MANTER CLIENTE



Figura 5- Use Case Manter Cliente

Nome Use Case	Manter Cliente
Ator	Usuário
Descrição	A função deste caso de uso é inserir, alterar, excluir e listar cliente.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema mostrara os campos a serem preenchidos 2- O usuário devera preencher os campos necessários 3- Após preencher corretamente o cadastro será finalizado
Cenário Alternativo	<ol style="list-style-type: none"> 1- Sistema retorna mensagem "Cliente não pode ser inserido" 2- Sistema retorna mensagem "Cliente não pode ser alterado" 3- Sistema retorna mensagem "Cliente não pode ser excluído" 4- Sistema retorna mensagem " É necessário preencher campos obrigatórios".

Tabela 13- Use Case Manter Cliente

10.1.2. MANTER MECÂNICO



Figura 6- Use Case Manter Mecânico

Nome Use Case	Manter Mecânico
Ator	Usuário
Descrição	A função deste caso de uso é inserir, alterar, excluir e listar mecânico.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema mostrara os campos a serem preenchidos 2- O usuário devera preencher os campos necessários 3- Após preencher corretamente o cadastro será finalizado
Cenário Alternativo	<ol style="list-style-type: none"> 1- Sistema retorna mensagem "Mecânico não pode ser inserido" 2- Sistema retorna mensagem "Mecânico não pode ser alterado" 3- Sistema retorna mensagem "Mecânico não pode ser excluído" 4- Sistema retorna mensagem "É necessário preencher campos obrigatórios".

Tabela 14- Use Case Manter Mecânico

10.1.3. MANTER VEÍCULO



Figura 7- Use Case Manter Veículo

Nome Use Case	Manter Veículo
Ator	Usuário
Descrição	A função deste caso de uso é inserir, alterar, excluir e listar veículo.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema mostrara os campos a serem preenchidos 2- O usuário devera preencher os campos necessários 3- Após preencher corretamente o cadastro será finalizado
Cenário Alternativo	<ol style="list-style-type: none"> 1- Sistema retorna mensagem " Veiculo não pode ser inserido" 2- Sistema retorna mensagem "Veiculo não pode ser alterado" 3- Sistema retorna mensagem "Veiculo não pode ser excluído" 4- Sistema retorna mensagem " É necessário preencher campos obrigatórios

Tabela 15- Use Case Manter Veículo

10.1.4. MANTER ORDEM DE SERVIÇO



Figura 8- Use Case Manter Ordem Serviço

Nome Use Case	Manter Ordem serviço
Ator	Usuário
Descrição	A função deste caso de uso é inserir, alterar, excluir e listar ordem de serviço.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema mostrara os campos a serem preenchidos 2- O usuário devera abrir ordem de serviço para o veiculo 3- Após abrir ordem serviço será locado um mecânico para a realização do serviço.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Sistema retorna mensagem "Ordem de serviço em aberto" 2- Sistema retorna mensagem "Ordem do serviço finalizada" 3- Sistema retorna mensagem "Ordem de serviço não pode ser excluída"

Tabela 16- Use Case Manter Ordem Serviço

10.1.5. MANTER ORÇAMENTO



Figura 9- Use Case Manter Orçamento

Nome Use Case	Manter Orçamento
Ator	Usuário
Descrição	A função deste caso de uso é inserir, alterar, excluir e listar orçamento.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema mostrara os campos a serem preenchidos 2- O usuário devera criar um orçamento para o cliente 3- Após criar o orçamento será emitido uma copia para o cliente.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Sistema retorna mensagem "Orçamento finalizado com sucesso" 2- Sistema retorna mensagem "Acrescentar item" 3- Sistema retorna mensagem "Excluir item"

Tabela 17- Use Case Manter Orçamento

10.2 EMISSÕES DE RELATÓRIOS

10.2.1. EMITIR RELATÓRIO CLIENTE



Figura 10- Use Case Emitir Relatório Cliente

Nome Use Case	Emitir Relatório de Clientes
Ator	Usuário
Descrição	A função deste caso de uso é emitir relatórios de todos os clientes cadastrados.
Cenário Principal	O usuário solicita emissão de relatórios de Clientes.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios de clientes. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 18- Use Case Emitir Relatório Cliente

10.2.2. EMITIR RELATÓRIO VEÍCULO POR CLIENTE



Figura 11- Use Case Emitir Relatório Veículo por Cliente

Nome Use Case	Emitir Relatório Veículo por Cliente
Ator	Usuário
Descrição	A função deste caso de uso é emitir relatórios de todos os veículos por cliente cadastrados.
Cenário Principal	O usuário solicita emissão de relatórios do veículo por cliente.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios do veículo por cliente. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 19- Use Case Emitir Relatório Veículo por Cliente

10.2.3. EMITIR RELATÓRIO CLIENTE DEVEDOR



Figura 12- Use Case Emitir Relatório Cliente Devedor

Nome Use Case	Emitir Relatório Cliente Devedor
Ator	Usuário
Descrição	A função deste caso de uso é emitir relatórios de todos os clientes devedores.
Cenário Principal	O usuário solicita emissão de relatórios de clientes devedores.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios do cliente devedor. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 20- Use Case Emitir Relatório Cliente Devedor

10.2.4. EMITIR RELATÓRIO CONTA RECEBER POR DATA



Figura 13- Use Case Emitir Relatório Conta Receber por Data

Nome Use Case	Emitir Relatório Conta Receber por Data
Ator	Usuário
Descrição	A função deste caso de uso é emitir todas contas a receber por data.
Cenário Principal	O usuário solicita emissão de relatórios da conta receber por data.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios das contas por data. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 21- Use Case Emitir Relatório Conta Receber por Data

10.2.5. EMITIR RELATÓRIO DE MECÂNICO



Figura 14- Use Case Emitir Relatório Mecânico

Nome Use Case	Emitir Relatório Mecânico
Ator	Usuário
Descrição	A função deste caso de uso é emitir relatórios de todos mecânicos cadastrados.
Cenário Principal	O usuário solicita emissão de relatórios dos mecânicos.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios dos mecânicos. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 22- Use Case Emitir Relatório Mecânico

10.2.6. MANTER ORDEM DE SERVIÇO POR MECÂNICO



Figura 15- Use Case Emitir Relatório Ordem Serviço por Mecânico

Nome Use Case	Emitir Relatório Ordem de Serviço por Mecânico
Ator	Usuário
Descrição	A função deste caso de uso é emitir relatórios de todos os serviços prestados pelo mecânico.
Cenário Principal	O usuário solicita emissão de relatórios ordem de serviço por mecânico.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios da ordem de serviço por mecânico. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 23- Use Case Emitir Relatório Ordem Serviço por Mecânico

10.2.7. MANTER ORDEM DE SERVIÇO POR DATA



Figura 16- Use Case Emitir Relatório Ordem Serviço por Data

Nome Use Case	Emitir Relatório Ordem de Serviço por Data
Ator	Usuário
Descrição	A função deste caso de uso é emitir relatórios de todas as ordens de serviços realizadas, por data.
Cenário Principal	O usuário solicita emissão de relatórios de ordem de serviço por data.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios ordem de serviço por data. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 24- Use Case Emitir Relatório Ordem Serviço por Data

10.2.8. MANTER RELATÓRIO ORDEM DE SERVIÇO POR CÓDIGO



Figura 17- Use Case Emitir Relatório Ordem de Serviço por Código

Nome Use Case	Emitir Relatório Ordem de Serviço por Código.
Ator	Usuário
Descrição	A função deste caso de uso é emitir relatórios de todas as ordens de serviços pelos códigos gerados.
Cenário Principal	O usuário solicita emissão de relatórios da ordem de serviço por código.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios da ordem de serviço por código. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 25- Use Case Emitir Relatório Ordem Serviço por Código

10.2.9. MANTER RELATÓRIO DE SERVIÇO EM ABERTO



Figura 18- Use Case Emitir Relatório Orçamento em Aberto

Nome Use Case	Emitir Relatório Orçamento em Aberto
Ator	Usuário
Descrição	A função deste caso de uso é emitir relatórios de todos os orçamentos em aberto.
Cenário Principal	O usuário solicita emissão de relatórios de orçamento em aberto.
Cenário Alternativo	<ol style="list-style-type: none"> 1- Selecionar Relatórios. 2- Abrir tela de relatórios de orçamento em aberto. 3- Sistema retorna mensagem "Emissão efetuada com sucesso".

Tabela 26- Use Case Emitir Relatório Orçamento em Aberto

11. DIAGRAMA DE CLASSES

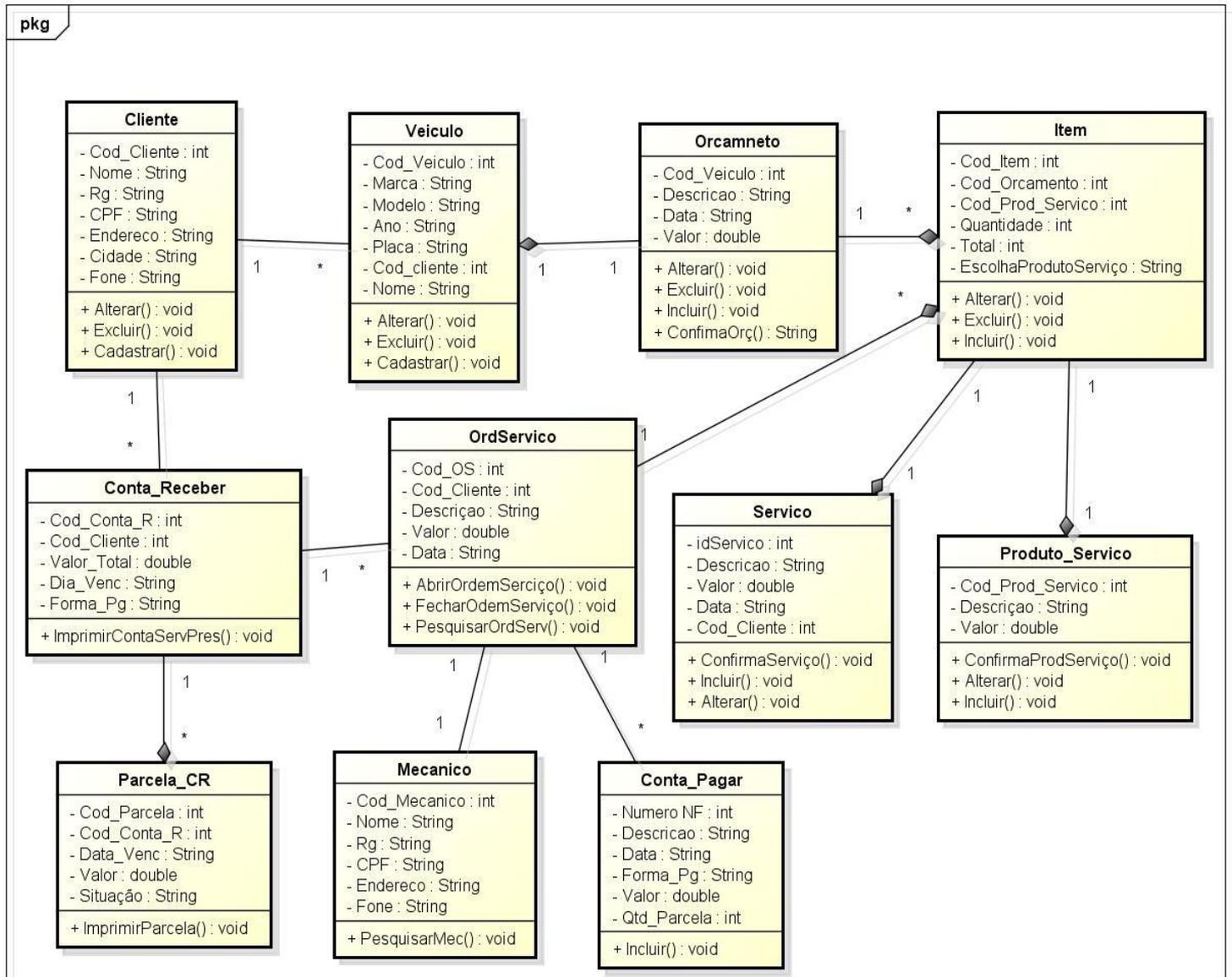


Figura 19- Diagrama de Classes

12. DIAGRAMA DE SEQUÊNCIA

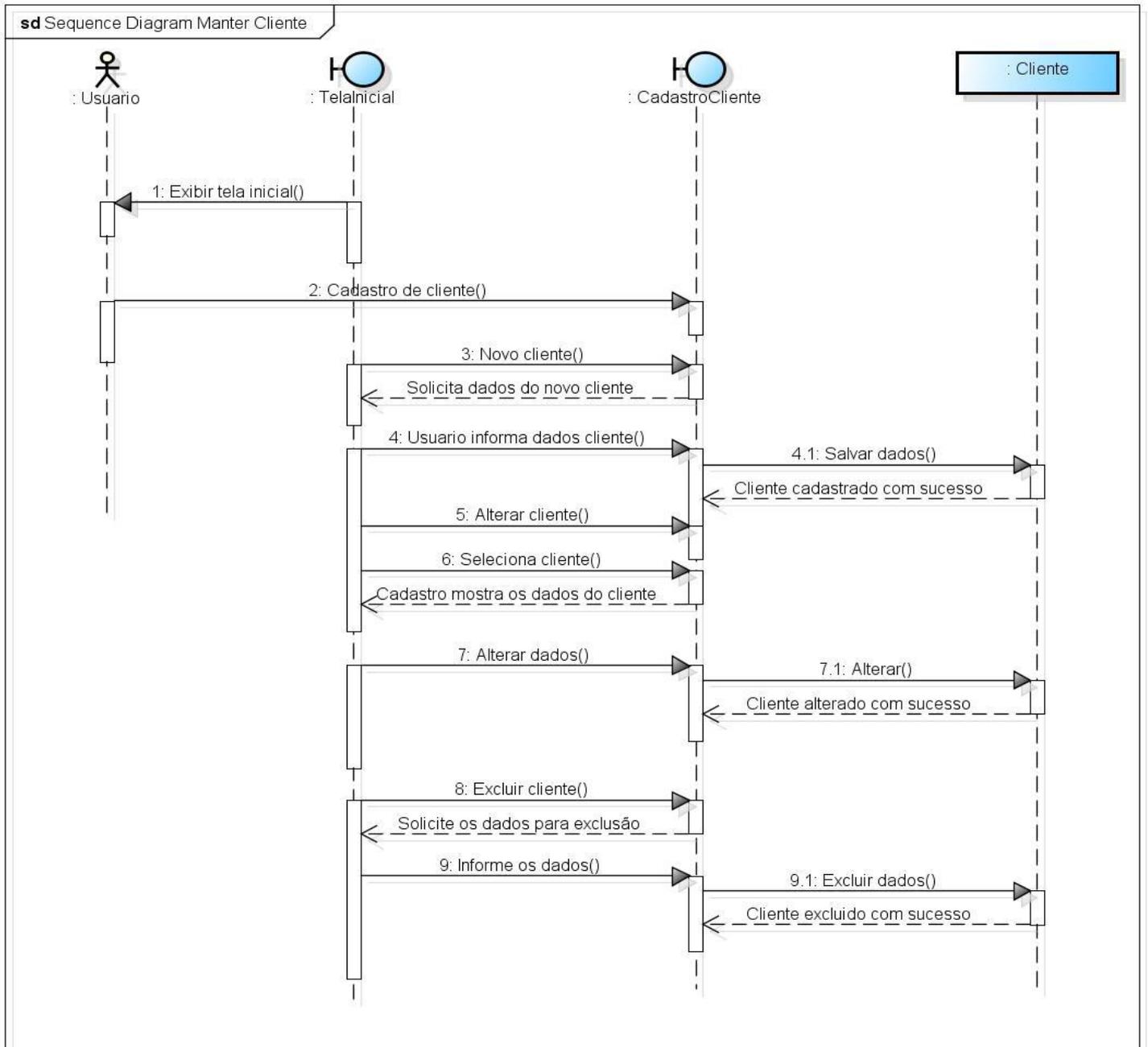


Figura 20- Diagrama de Sequência

13. DIAGRAMA DE ATIVIDADE

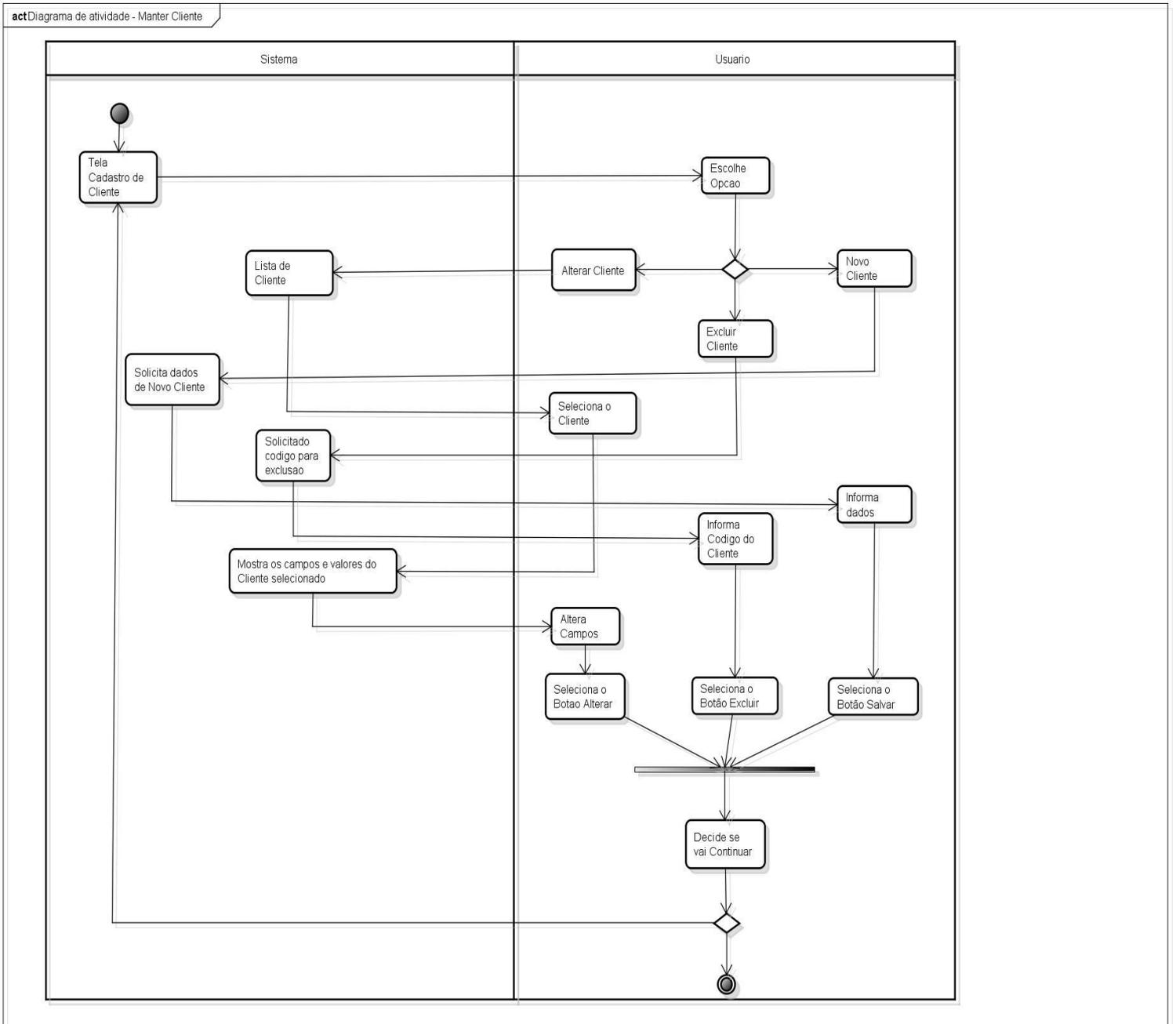


Figura 21- Diagrama de Atividade

14. CONCLUSÃO

Este trabalho foi desenvolvido para, ajudar os usuários autônomos que não tem tanto conhecimento e agilidade com o computador. Esse sistema irá ajudar a fazer cadastros e ter um controle de suas movimentações no negócio.

O sistema foi desenvolvido somente para suprir a necessidade de um usuário conhecido, mas caso houver comercialização do mesmo, este será implementado aplicando-se toda a regra de negócio necessária para torná-lo um sistema de fácil manuseio e completo, atendendo todas as necessidades do cliente.

15. ANEXOS

15.1 CONEXÕES DO BANCO DE DADOS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace OficinaMecanica.clsDados.DAL
{
    public class Banco
    {
        public static string stringdeconexao { get { string conexao = @"Data Source=.\sqlexpress;Initial
Catalog=Oficina;Integrated Security=True;Pooling=False"; return conexao; }}
    }
}
```

15.1.2. MENU



Figura 22- Tela - Menu

15.1.3. CADASTRO DE CLIENTE

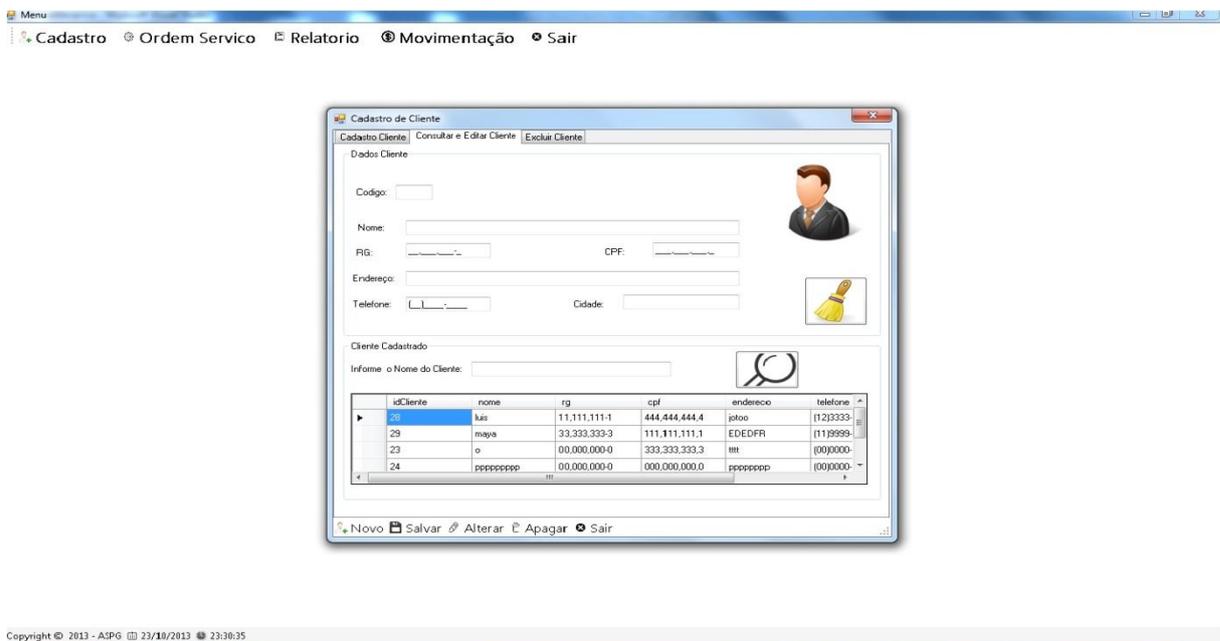


Figura 23- Tela – Cadastro de Cliente

15.1.4. ORÇAMENTO

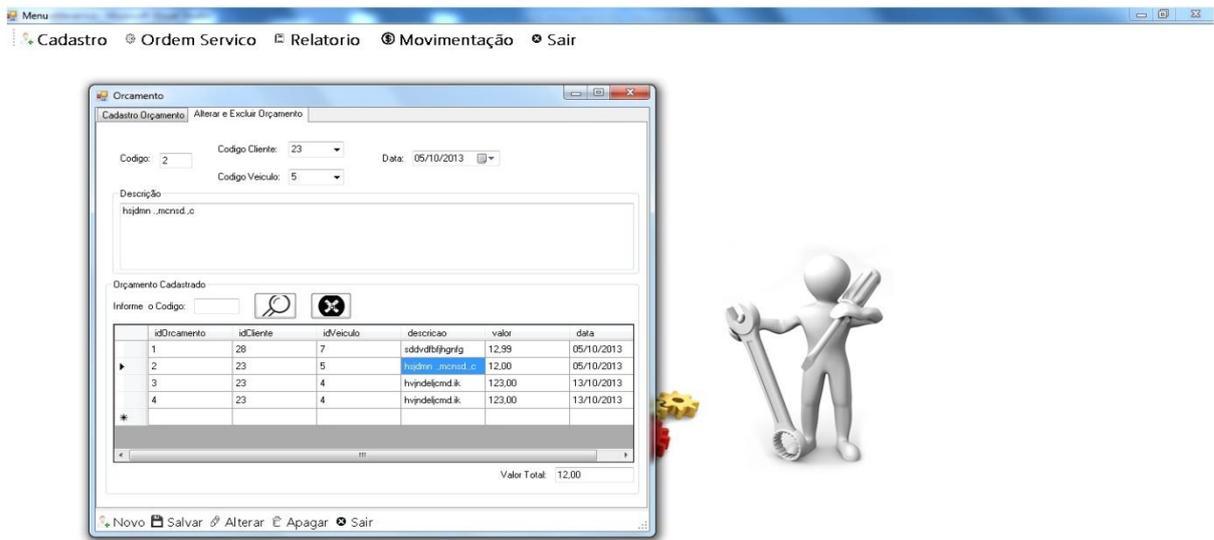


Figura 24- Tela - Orçamento

15.1.5. ORDEM DE SERVIÇO

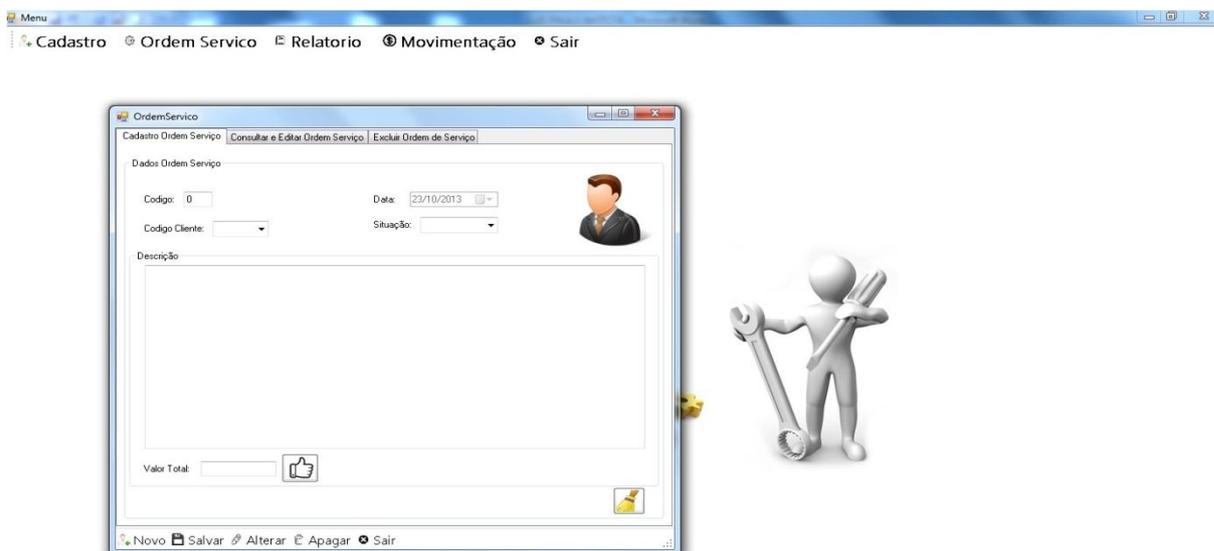


Figura 25- Tela – Ordem Serviço

15.2 COMANDOS



Figura 26- Tela - Comandos

15.2.1 COMANDO NOVO

```
private void toolStripStatusLabel1_Click(object sender, EventArgs e)
{
    txtCodigo.Text = "0";
    txtNome.Text = "";
    MtxtRG.Text = "";
    MtxtCPF.Text = "";
    MtxtTelefone.Text = "";
    txtEndereco.Text = "";
    txtCidade.Text = "";
    txtNome.Focus();
}
```

Botão novo serve para criar o próximo cadastro.

15.2.2. COMANDO SALVAR

```
DialogResult result;
string msg = "Deseja salvar o Cliente: " + txtNome.Text.ToString() + " ?";
result = MessageBox.Show(msg, "Salvar", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);
if (result == DialogResult.Yes)
{
    ocli.nome = txtNome.Text;
    ocli.rg = MtxtRG.Text;
    ocli.cpf = MtxtCPF.Text;
    ocli.endereco = txtEndereco.Text;
    ocli.telefone = MtxtTelefone.Text;
    ocli.cidade = txtCidade.Text;
    dalCli.Insert(ocli);
    MessageBox.Show("O cliente: " + txtNome.Text + "\n\nFoi cadastrado com sucesso!!");
    dtvCliente.DataSource = dalCli.Select();
    dtvCliente2.DataSource = dtvCliente.DataSource;
    dtvCliente.Refresh();
}
```

Botão salvar serve para salvar dados do sistema.

15.2.3. COMANDO ALTERAR

```

DialogResult result;
string msg = "Deseja alterar o Cliente: " + txtNome2.Text.ToString() + " ?";
result = MessageBox.Show(msg, "Alterar", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);
if (result == DialogResult.Yes)
{
    ocli.idCliente = Convert.ToInt32(txtCodigo2.Text);
    ocli.nome = txtNome2.Text;
    ocli.rg = MtxtRG2.Text;
    ocli.cpf = MtxtCPF2.Text;
    ocli.endereco = txtEndereco2.Text;
    ocli.telefone = MtxtTelefone2.Text;
    ocli.cidade = txtCidade2.Text;
    dalCli.Update(ocli);
    MessageBox.Show("O cliente: " + txtNome2.Text + "\n\nFoi alterado com sucesso!!");
    dtvCliente.DataSource = dalCli.Select();
    dtvCliente2.DataSource = dtvCliente.DataSource;
    dtvCliente.Refresh();
}

```

Botão alterar, serve para alterar dados do sistema.

15.2.4. COMANDO APAGAR

```

clsDados.Model.Cliente ocli2 = new clsDados.Model.Cliente();
ocli2.idCliente = int.Parse(txtCodigo3.Text);
clsDados.DAL.Cliente dalCli2 = new clsDados.DAL.Cliente();
//clsDados.BLL.Cliente blCli2 = new clsDados.BLL.Cliente();
DialogResult result;
string msg = "Deseja excluir o Cliente: " + txtNome.Text.ToString() + " ?";
result = MessageBox.Show(msg, "Excluir", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);
if (result == DialogResult.Yes)
{
    dalCli2.Delete(ocli2);
    MessageBox.Show("O cliente: " + txtNome.Text + "\n\nFoi excluído com sucesso!!");
    dtvCliente.DataSource = dalCli2.Select();
    dtvCliente2.DataSource = dtvCliente.DataSource;
    dtvCliente.Refresh();
}

```

Botão apagar, serve para excluir arquivos sem uso.

15.2.5.COMANDO SAIR

```

private void toolStripStatusLabel5_Click(object sender, EventArgs e)
{
    this.Close();
}

```

Botão sair, serve para fechar o sistema.

15.3 DAL. CLIENTE

```

public class Cliente
{
    public DataTable Select()
    {
        SqlConnection con = new SqlConnection(Banco.stringdeconexao);
        DataTable consulta = new DataTable();
        string sql = "SELECT * FROM Cliente order by nome";
        con.Open();
        SqlDataAdapter da = new SqlDataAdapter(sql, con);
        da.Fill(consulta);
        return consulta;
    }

    public void Insert(Model.Cliente oCli) //oCli="Objeto cliente"
    {
        SqlConnection con = new SqlConnection(Banco.stringdeconexao);
        string sql = "INSERT INTO Cliente (nome,rg,cpf,endereco,telefone,cidade)
VALUES (@nome,@rg,@cpf,@endereco,@telefone,@cidade)";
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.Parameters.AddWithValue("@nome", oCli.nome);
        cmd.Parameters.AddWithValue("@rg", oCli.rg);
        cmd.Parameters.AddWithValue("@cpf", oCli.cpf);
        cmd.Parameters.AddWithValue("@endereco", oCli.endereco);
        cmd.Parameters.AddWithValue("@telefone", oCli.telefone);
        cmd.Parameters.AddWithValue("@cidade", oCli.cidade);
        con.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        finally
        {
            con.Close();
        }
    }

    public void Update(Model.Cliente oCli)
    {
        SqlConnection con = new SqlConnection(Banco.stringdeconexao);
        string sql = "UPDATE Cliente SET nome = @nome, rg = @rg, cpf= @cpf,
endereco = @endereco,telefone=@telefone,cidade=@cidade WHERE idCliente = @idCliente";
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.Parameters.AddWithValue("@idCliente", oCli.idCliente);
        cmd.Parameters.AddWithValue("@nome", oCli.nome);
        cmd.Parameters.AddWithValue("@rg", oCli.rg);
        cmd.Parameters.AddWithValue("@cpf", oCli.cpf);
        cmd.Parameters.AddWithValue("@endereco", oCli.endereco);
        cmd.Parameters.AddWithValue("@telefone", oCli.telefone);
        cmd.Parameters.AddWithValue("@cidade", oCli.cidade);
        con.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        finally
        {
    }
}

```

```
        con.Close();
    }
}

public void Delete(Model.Cliente oCli)
{
    SqlConnection con = new SqlConnection(Banco.stringdeconexao);
    string sql = "DELETE Cliente WHERE idCliente = @idCliente";
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.Parameters.AddWithValue("@idCliente", oCli.idCliente);
    con.Open();
    try
    {
        cmd.ExecuteNonQuery();
    }
    finally
    {
        con.Close();
    }
}
```

REFERÊNCIAS

ALECRIM, Emerson. Tecnologia da informação Disponível em: <<http://www.infowester.com/ti.php>>. acesso em 20 janeiro de 2013.

PRÉ-PROJETO PARA MONOGRAFIA Disponível em: <<http://www.fbb.org.br/upload/biblioteca1145549889468.doc>>. acesso em 20 janeiro de 2013

YOURDON, Edward; ARGILA, Carl. Análise e Projeto Orientados a Objetos: Estudo de Casos, 1º edição. Tradução de Angelina Carvalho Gomes e Álvaro Antunes. São Paulo: Editora MAKRON Books, 1999.

SILVA, Nelson Peres da . Projeto e Desenvolvimento de Sistemas: 7º edição São Paulo: Editora Érica,1998.

NOGUEIRA, Julio Da silva. Aprendendo a modelar no DBdesigner Disponível em: <<http://www.ifes.edu.br/informatica/leonardop/MD/DBDesigner%204.pdf>>. acesso em 20 de janeiro de 2013.

BOLZAN, Willian . Modelagem e projeto de banco de dados com o Dbdesigner Disponível em: <<http://www.devmedia.com.br/modelagem-e-projeto-de-banco-de-dados-com-dbdesigner/7773#ixzz2Y805U0ZX>>. acesso em 01 junho de 2013.

PACIEVITCH, Yuri. SQL Server – Banco de dados. Disponível em: <<http://www.infoescola.com/informatica/sql-server/>>. acesso em: 28 setembro 2013

PRADO, Chico Dal Santo . Linguagem de Programação C#. Disponível em:<<http://analisedesistemasunopar.blogspot.com.br/2013/06/linguagem-de-progracao-c.html>>. acesso em 28 setembro de 2013

JÚNIOR, Carlos Olavo de Azevedo Camacho. Desenvolvimento em Camadas com C# .NET, Florianópolis: Editora Visual Books, 2008.

LOTAR, Alfredo. Como Programar com ASP. NET e C#, 2º edição. São Paulo: Editora Novatec,2010