

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE INFORMÁTICA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

HELITON RODRIGO DA SILVA

SISTEMA DE TRANSPORTE ESCOLAR

TRABALHO DE DIPLOMAÇÃO

CORNÉLIO PROCÓPIO

2014

HELITON RODRIGO DA SILVA

SISTEMA DE TRANSPORTE ESCOLAR

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina Trabalho de Diplomação do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR – Campus Cornélio Procópio, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. André Luiz Przybysz

CORNÉLIO PROCÓPIO

2014

Dedico este trabalho a Deus sobre todas as coisas, aos meus pais, familiares, amigos e a minha noiva Nara, por me ajudarem tanto nesta caminhada e concretização deste trabalho.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus que iluminou o meu caminho durante esta caminhada.

Agradeço também toda minha família, amigos e minha noiva Nara que de forma especial estiveram ao meu lado, me dando força, coragem e me apoiando nos momentos de dificuldade nessa importante fase da minha vida.

Aos meus professores da UTFPR campus Cornélio Procópio pela atenção e contribuição dedicadas a este trabalho, em especial ao meu orientador e professor André Luiz Przybysz, pela sua dedicação, apoio e incentivo no amadurecimento dos meus conhecimentos e conceitos que me levaram a execução e conclusão deste trabalho.

“Se Deus é por nós, quem será
contra nós.”
(Romanos 8:31).

RESUMO

SILVA, H. **Sistema de Transporte Escolar**. 2013. 135 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2013.

O objetivo do presente trabalho foi o desenvolvimento de um software denominado Sistema de Transporte Escolar. A aplicação deste sistema é de controlar as atividades cadastrais e financeiras de empresas de transporte escolar. Para o desenvolvimento deste sistema foi utilizada uma variação da *Xtreme Programming*, que é uma metodologia de desenvolvimento de software que segue os princípios do Manifesto Ágil, a modelagem foi baseada na *Unified Modeling Language* (UML). As linguagens de programação de computadores adotadas foram C#.Net, Asp.Net e HTML com o auxílio da ferramenta Visual Studio 2010, e os banco de dados utilizados foram o PostgreSQL e SQL Server CE.

Palavras-chave: Software. Transporte. Escolar. *Nhibernate*, Padrões de Projeto.

ABSTRACT

SILVA, H. **School Transportation System**. 2013. 135 f. Major Completion Work (Technology Analysis and Systems Development) Federal Technological University of Paraná. Cornélio Procópio, 2013.

The aim of the present work was the development of software called School Transportation System. The application of this system is to control the cadastral and financial activities of School Transportation Companies. For the development of this system, a variation of Xtreme Programming was used, which is a methodology of software development that follows the principles of Agile Manifesto, the modeling was based on the Unified Modeling Language (UML). The adopted language programs were C#Net, Asp.Net e HTML, with the aid of the tool Visual Studio 2010 and the database used was PostgreSQL and SQL Server CE.

Keywords: Software. Transportation. School. Nhibernate, Design Patterns.

LISTA DE TABELAS

TABELA 1 - PAPÉIS E RESPONSABILIDADES.....	47
TABELA C.1 - REQUISITOS FUNCIONAIS.....	66
TABELA C.2 - REQUISITOS NÃO-FUNCIONAIS DO SISTEMA.....	67
TABELA F1.1 - ESPECIFICAÇÃO DO CASO DE USO: EFETUAR LOGIN	83
TABELA F1.2 - ESPECIFICAÇÃO DO CASO DE USO: ALTERAR SENHA	84
TABELA F1.3 - ESPECIFICAÇÃO DO CASO DE USO: VISUALIZAR DADOS DO CLIENTE.....	85
TABELA F1.4 - ESPECIFICAÇÃO DO CASO DE USO: GERAR CARTEIRA DECONTROLE	86
TABELA F2.1 - ESPECIFICAÇÃO DO CASO DE USO: EFETUAR LOGIN	89
TABELA F2.2 - ESPECIFICAÇÃO DO CASO DE USO: ALTERAR SENHA	90
TABELA F2.3 - ESPECIFICAÇÃO DO CASO DE USO: GERAR CONTRATOS	92
TABELA F2.4 - ESPECIFICAÇÃO DO CASO DE USO: INCLUIR.....	94
TABELA F2.5 - ESPECIFICAÇÃO DO CASO DE USO: ALTERAR.....	95
TABELA F2.6 - ESPECIFICAÇÃO DO CASO DE USO: PESQUISAR	97
TABELA F2.7 - ESPECIFICAÇÃO DO CASO DE USO: EXCLUIR.....	98
TABELA F2.8 - ESPECIFICAÇÃO DO CASO DE USO: VISUALIZAR CONTRATOS.....	100
TABELA F2.9 - ESPECIFICAÇÃO DO CASO DE USO: GERAR RELATÓRIOS ...	101

LISTA DE FIGURAS

FIGURA 1 - STARUML	22
FIGURA 2 - EXEMPLO DE IMPLEMENTAÇÃO DO PADRÃO <i>SINGLETON</i>	23
FIGURA 3 - DIAGRAMA DE SEQUENCIA DO REPOSITORY	24
FIGURA 4 - CLASSE BASE DO PADRÃO REPOSITORY	25
FIGURA 5 - CLASSE DE REPOSITÓRIO DE CLIENTES	26
FIGURA 6 - TRECHO DE CÓDIGO DA CLASSE FABRICA DE REPOSITÓRIOS	27
FIGURA 7 - MODELO MVC	28
FIGURA 8 - ARQUITETURA DE HARDWARE DO SISTEMA	32
FIGURA 9 - CICLO DE VIDA DE UMA HISTÓRIA EM PROJETO XP	34
FIGURA 10 - CICLO DE VIDA XP	36
FIGURA B.1 - EAP SGVC	55
FIGURA B.2 - CICLO DE VIDA DE UMA ITERAÇÃO XP	57
FIGURA D.2 - HOME	71
FIGURA D.3 - EXCURSÕES	71
FIGURA D.4 - FALECONOSCO	72
FIGURA D.5 - FRETAMENTO	72
FIGURA D.6 - GALERIA	73
FIGURA D.7 - GALERIA1	73
FIGURA D.8 - LOCALIZAÇÃO	74
FIGURA D.9 - LOGIN	74
FIGURA D.10 - TRANSPORTEALUNOSCOMPONENT SCREENS	75
FIGURA D.11 - MASTERPAGE	76
FIGURA F1.1 - CASO DE USO: ATORES DO SISTEMA DESKTOP	81
FIGURA F1.2 - CASO DE USO DO SISTEMA DESKTOP	81
FIGURA F1.3 - CASO DE USO: EFETUAR LOGIN	82
FIGURA F1.4 - CASO DE USO: ALTERAR SENHA	83
FIGURA F1.5 - CASO DE USO: VISUALIZAR DADOS DO CLIENTE	85
FIGURA F1.6 - CASO DE USO: GERAR CARTEIRA DE CONTROLE	86
FIGURA F2.1 - CASO DE USO: ATORES DO SISTEMA WEB	87
FIGURA F2.2 - CASO DE USO: SISTEMA WEB	87
FIGURA F2.3 - CASO DE USO: EFETUAR LOGIN	88
FIGURA F2.4 - CASO DE USO: ALTERAR SENHA	89
FIGURA F2.5 - CASO DE USO: GERAR CONTRATOS	91
FIGURA F2.6 - CASO DE USO: INCLUIR	93
FIGURA F2.7 - CASO DE USO: INCLUIR	94
FIGURA F2.8 - CASO DE USO: PESQUISAR	96
FIGURA F2.9 - CASO DE USO: PESQUISAR	97
FIGURA F2.10 - CASO DE USO: VISUALIZAR CONTRATOS	99

FIGURA F2.11 - CASO DE USO: VISUALIZAR CONTRATOS	100
FIGURA F3.1 - SEQUÊNCIA: EFETUAR LOGIN.....	102
FIGURA F3.2 - SEQUÊNCIA: ALTERAR SENHA.....	102
FIGURA F3.3 - SEQUÊNCIA: VISUALIZAR DADOS DO CLIENTE	103
FIGURA F3.4 - SEQUÊNCIA: GERAR HISTÓRICO	103
FIGURA F3.5 - SEQUÊNCIA: GERAR CONTRATOS	104
FIGURA F3.6 - SEQUÊNCIA: INCLUIR	104
FIGURA F3.7 - SEQUÊNCIA: ALTERAR	105
FIGURA F3.8 - SEQUÊNCIA: PESQUISAR.....	105
FIGURA F3.9 - SEQUÊNCIA: EXCLUIR	106
FIGURA F3.10 - SEQUÊNCIA: VISUALIZAR CONTRATOS	106
FIGURA F3.11 - SEQUÊNCIA: GERAR RELATÓRIOS.....	107
FIGURA F4 - DIAGRAMA DE CLASSE	108
FIGURA G.1 - MER SISTEMA <i>DESKTOP</i>	110
FIGURA G.2 - MER SISTEMA <i>WEB</i>	111

LISTA DE SIGLAS E ABREVIATURAS

3G	Terceira Geração
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos)
BD	Banco de Dados
CLR	<i>Common Language Runtime</i> (Ambiente de Tempo de Execução de Linguagem Comum)
CSS	<i>Cascading Style Sheets</i> (folhas de estilo em cascata)
EAP	Estrutura Analítica do Projeto
ERD	<i>Entity-Relationship Model</i> (Modelo de Entidade-Relacionamento)
HTML	<i>HyperText Markup Language</i> (Linguagem de Marcação de Hipertexto)
IDE	<i>Integrated Development Environment</i> (Ambiente Integrado de Desenvolvimento)
MA	Modelagem Ágil
MER	Modelo de Entidade-Relacionamento
MVC	<i>Model-View-Controller</i> (Modelo-Visão-Controlador)
ORM	<i>Object Relational Mapping</i> (Mapeamento Objeto-Relacional)
SDK	<i>Software Development Kit</i> (Kit de Desenvolvimento de Aplicativos)
SI	Sistema de Informação
SQL	<i>Structure Query Language</i> (Linguagem de Consulta Estruturada)
SQLCE	<i>SqlServer Compact Edition</i>
SSMSE	<i>Microsoft SqlServer Management Studio Express</i>
TCC	Trabalho de Conclusão de Curso
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)
WPF	<i>Windows Presentation Foundation</i>
XML	<i>eXtensible Markup Language</i> (Linguagem de Marcação Extensível)
XP	<i>eXtreme Programming</i> (Programação Extrema)
.NET	<i>Framework</i> de desenvolvimento para plataforma <i>Windows</i> .
ADO	<i>ActiveX Data Objects</i>
CASE	<i>Computer-Aided Software Engineering</i> (Ferramentas baseadas em computador que auxiliam atividade de engenharia de software)
CRUD	<i>Create, Read, Update, Delete</i> (Visualizar, Excluir, Inserir, Alterar)
LINQ	<i>Language Integrated Query</i>
MER	Modelo de Entidade-Relacionamento

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 JUSTIFICATIVA.....	14
1.2 OBJETIVOS ESPECÍFICOS.....	15
1.3 ORGANIZAÇÃO DO TRABALHO.....	16
2 FERRAMENTAS E TECNOLOGIAS UTILIZADAS.....	17
2.1 FERRAMENTA IDE VISUAL STUDIO 2010.....	17
2.2 MICROSOFT EXPRESSION BLEND.....	17
2.3 PLATAFORMA .NET.....	18
2.4 LINGUAGEM DE PROGRAMAÇÃO C#.....	18
2.5 ASP.NET MVC.....	19
2.6 HTML.....	19
2.7 CSS - CASCADING STYLE SHEETS.....	19
2.8 MICROSOFT SQL SERVER MANAGEMENT STUDIO 2008.....	19
2.9 MICROSOFT SQL SERVER COMPACT EDITION.....	20
2.10 UML.....	20
2.10.1 Diagrama de Caso de Uso.....	21
2.10.2 Diagrama de Classes.....	21
2.10.3 Diagrama de Sequência.....	21
2.11 STARUML.....	22
2.12 PADRÕES DE PROJETO.....	22
2.12.1 Padrão Singleton.....	23
2.12.2 Padrão Repository.....	24
2.12.3 Padrão MVC.....	27
2.13 LINQ (LANGUAGE INTEGRATED QUERY).....	29
2.13.1 Query Expressions.....	29
2.14 NHIBERNATE.....	29
2.14.1 CRUD (<i>Create, Read, Update e Delete</i>).....	30
2.15 RECURSOS UTILIZADOS.....	30
3 ARQUITETURA DO SISTEMA.....	32
4 METODOLOGIA DE DESENVOLVIMENTO.....	34
5 CONCLUSÃO.....	37
5.1 TRABALHOS FUTUROS.....	37
5.2 CONSIDERAÇÕES FINAIS.....	37
REFERÊNCIAS.....	39
APÊNDICE A: PLANO DE DESENVOLVIMENTO.....	42
APÊNDICE B: APLICAÇÃO DO PROCESSO DE.....	52
APÊNDICE C: COLEÇÃO DE REQUISITOS.....	60
APÊNDICE D: <i>ABBATUR WPFPROTOTYPE</i>	68

APÊNDICE E - TESTES DO SOFTWARE	77
APÊNCIDE F – DIAGRAMAS DA UML.....	79
APÊNDICE G – DIAGRAMAS MODELO ENTIDADE-RELACIONAMENTO	109
ANEXO A - PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO.....	112

1 INTRODUÇÃO

Um Sistema de Informação (SI), é um sistema que tem como principal objetivo manter as informações, as armazenando, tratando e fornecendo informações no intuito de apoiar as funções ou processos de uma organização. As informações e processos entre as áreas funcionais da organização são compostos de módulos que atendem as necessidades de informação, proporcionando recursos e procedimentos aos usuários, para um gerenciamento eficiente de todos os processos administrativos, comerciais e operacionais. Conforme há uma expansão no mercado e o cenário da economia atual vem apresentando características cada vez mais rápidas e dinâmicas, surge a necessidade de um melhor controle das informações e automatização de processos.

Os *softwares* têm papel fundamental nas organizações, pois são através deles que um administrador consegue ter acesso com facilidade às informações de todos os aspectos de sua organização. A correta administração dessas informações é fundamental para seu sucesso.

Dentro deste contexto, o objetivo deste Trabalho de Conclusão de Curso (TCC), foi o desenvolvimento de um *software* para empresas de turismo, fretamento e transporte escolar, onde a principal atividade é o transporte de alunos para diversas instituições de ensino, no intuito de proporcionar à empresa um melhor controle sobre as contas a receber de seus clientes, gerar e armazenar contratos, controlar a frequência de uso de seus veículos por parte dos contratantes gerando um histórico de uso de cada cliente. Possibilitar a emissão de relatórios e divulgar seu trabalho por meio de um *website*.

1.1 JUSTIFICATIVA

Este *software* foi desenvolvido para empresas de transporte, que trabalham com fretamento de seus veículos para viagens, oferecimento de pacotes turísticos e como principal atividade o transporte intermunicipal de alunos para diversas instituições de ensino.

Visto que algumas empresas possuem vários clientes, carros, contratos, funcionários, e o controle de todos os seus registros ainda são feitos em planilhas eletrônicas ou papel, que podem sofrer violações ou até mesmo desaparecerem por não possuírem um padrão de armazenamento. Isto acaba acarretando diversos problemas e dificuldades, e como os principais destacam-se: a inadimplência de muitos de seus clientes, onde estes usufruem dos veículos da empresa mesmo estando com mensalidades atrasadas ou até mesmo sem ter efetuado nenhum pagamento; a falta de controle dos clientes, funcionários e contratos; nenhum controle e registro de uso dos clientes contratantes nos carros da empresa; sem possibilidade de criar relatórios; etc. Com todas dificuldades e problemas citados, representantes deste tipo de empresa depararam-se com a necessidade de implantação de um *software* com a finalidade de saná-los, também que estes representantes intuíram que nos dias de hoje é de extrema importância a divulgação de um trabalho e sabendo que a Internet é um dos principais e eficientes veículos de comunicação que existem atualmente, perceberam que é de extrema importância à divulgação do seu trabalho em um *website*.

1.2 OBJETIVOS ESPECÍFICOS

Uma aplicação *web* será desenvolvida visando sanar vários problemas para empresas de transporte de alunos escolares, principalmente, o da falta de controle das contas a receber e controle de uso de seus clientes, no que origina a inadimplência de alguns de seus clientes.

Mas além destes problemas relatados acima, o *software* também tem como objetivos:

- Garantir a empresa uma maior segurança dos dados cadastrados;
- Disponibilizar um melhor controle do cadastro de seus clientes e funcionários;
- Agilizar os cadastros e pesquisas;
- Possibilitar a geração de históricos de frequência de todos os clientes que estarão utilizando os carros da empresa por meio de uma máquina leitora de código de barras instalada nos carros da empresa;

- Possibilitar um melhor controle das contas a receber, de acordo com os contratos fechados com os clientes;
- Possibilitar a emissão de relatórios;
- Gerar carteira de identificação com foto e código de barras (somente para alunos).

Objetivando solucionar os problemas mencionados, é necessária a construção de uma versão *desktop* do sistema para ser implantado nos ônibus, no qual o motorista poderá logar no sistema que estará com a base de dados sempre atualizada por meio de uma comunicação com a Internet, assim sendo possível fazer a leitura do código do cliente devidamente cadastrado e com a carteira de identificação impressa através de um leitor de código de barras, que será instalado na entrada de todos os ônibus.

O sistema gerará automaticamente um histórico de frequência de cada cliente usuário que entrou no ônibus e ainda exibindo algumas informações ao motorista como: dados cadastrais e a situação do pagamento das mensalidades do cliente junto à empresa.

E com o propósito de solucionar o problema publicitário, é construído um *website*, no qual a empresa poderá fazer a divulgação de todo o seu trabalho ao público.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho é composto por cinco capítulos, seis apêndices e um anexo organizados da seguinte forma: o primeiro capítulo traz uma introdução do trabalho, a justificativa de desenvolver o sistema, os objetivos específicos e a forma em que o trabalho está organizado. No segundo capítulo são apresentadas as ferramentas, tecnologias e os recursos utilizados no desenvolvimento. No terceiro capítulo é mostrada a arquitetura do sistema. No quarto capítulo é mostrada a metodologia utilizada para o desenvolvimento deste trabalho. Por fim, quinto capítulo é apresentado a conclusão com as considerações finais e os trabalhos futuros.

2 FERRAMENTAS E TECNOLOGIAS UTILIZADAS

Neste capítulo serão apresentadas e descritas todas as tecnologias e ferramentas utilizadas para o desenvolvimento do *software* e *website* construídos.

2.1 FERRAMENTA IDE VISUAL STUDIO 2010

Visual Studio é um conjunto completo de ferramentas de desenvolvimento para construção de aplicações *Web ASP.NET*, serviços *Web XML*, aplicações *desktop* e aplicativos móveis, *Visual Basic*, *Visual C#* e *Visual C++* todos usam o mesmo ambiente de desenvolvimento integrado (IDE), que permite o compartilhamento de ferramentas e facilita a criação de soluções de linguagens mistas. Além disso, essas linguagens usam a funcionalidade do *.NET Framework*, que fornece acesso às tecnologias chaves que simplificam o desenvolvimento de aplicativos *Web* em *ASP* e serviços *Web XML* (MSDN, 2013).

Esta foi a ferramenta mais utilizada para o desenvolvimento do sistema, com ela foi feita a implementação e depuração dos códigos do sistema.

2.2 MICROSOFT EXPRESSION BLEND

Microsoft Expression Blend é uma ferramenta visual para design e prototipagem de *desktop* e aplicações *web*. Com esta ferramenta é possível construir controles como botões, listas, inserir imagens, áudio e clips de vídeo e controles personalizados de bibliotecas SDK ou de terceiros.

E com o *SketchFlow*, um conjunto de recursos disponível apenas no *Expression Studio Ultimate*, pode-se criar facilmente protótipos funcionais de aplicações em vez de uma série de imagens estáticas (MICROSOFT, 2012).

Com esta ferramenta foi possível criar um protótipo interativo da interface gráfica das páginas *web*.

2.3 PLATAFORMA .NET

A plataforma *Microsoft* oferece suporte a aplicativos e soluções projetadas para os menores dispositivos, bem como para as maiores empresas. Os desenvolvedores de software que usam a plataforma *Microsoft* podem:

- Trabalhar com uma variedade de linguagens, padrões e ferramentas de programação;
- Criar aplicativos para o *Windows*, o *Office*, a *Web* e para os dispositivos móveis;
- Beneficiar-se dos recursos que a *Microsoft* oferece para ajudar os desenvolvedores de software a criar e projetar aplicativos de classe mundial.

O código escrito para o *.NET Framework* é denominado código gerenciado. Em contraste com o código nativo tradicional, o código gerenciado é escrito em linguagens como *Visual C#* e *Visual Basic*, e é compilado no *Intermediate Language* e depois traduzido em código de máquina antes de ser executado.

O *.NET Framework* oferece suporte a várias linguagens de programação. Ele é executado em plataformas *Windows* como o *Windows 7* e o *Windows Server 2008 R2*.

O CLR (*common language runtime*) fornece serviços que ajudam no gerenciamento e execução das aplicações.

As bibliotecas de classes do *Framework* facilitam o processo de desenvolvimento e fornecem acesso a funções essenciais do sistema (MSDN, 2012).

2.4 LINGUAGEM DE PROGRAMAÇÃO C#

C# é uma linguagem elegante e de tipos protegidos, orientada a objeto e que permite aos desenvolvedores construir uma variedade de aplicações seguras e robustas, compatíveis com o *.NET Framework*. Você pode usar o *C#* para criar aplicações tradicionais do *Windows*, *Web services* baseados em

XML, componentes distribuídos, aplicativos cliente-servidor, aplicativos com banco de dados e muito mais. O Visual C# 2008 oferece um avançado editor de código, uma interface de usuário conveniente, depurador integrado e muitas outras ferramentas para facilitar o desenvolvimento de aplicativos baseados na versão 3.0 da linguagem C# e na versão 3.5 do *.NET Framework* (MSDN, 2012).

2.5 ASP.NET MVC

O ASP.NET MVC é uma estrutura para desenvolvimento de aplicativos da Web que segue o padrão de MVC (*Model-View-Controller*).

2.6 HTML

HTML significa '*HyperText Markup Language*', e é uma linguagem universal destinada à elaboração de páginas com hipertexto.

2.7 CSS - CASCADING STYLE SHEETS

É um documento onde são definidas regras de formatação ou de estilos, a serem aplicadas aos elementos estruturais de marcação. A finalidade das CSS é a de retirar do HTML toda e qualquer declaração que vise a formatação, a apresentação do documento.

2.8 MICROSOFT SQL SERVER MANAGEMENT STUDIO 2008

O *Microsoft SQL Server 2008 Management Studio Express* (SSMSE) é um ambiente de desenvolvimento integrado para acessar, configurar, gerenciar e desenvolver todos os componentes do *SQL Server*. O *SQL Server 2008 Management Studio Express* combina um amplo grupo de ferramentas gráficas com editores de scripts sofisticados que fornecem acesso ao *SQL Server* a desenvolvedores e administradores de todos os níveis de experiência.

Desenvolvedores terão uma experiência familiar e os administradores de banco de dados terão um único utilitário abrangente que combina ferramentas

gráficas fáceis de usar com sofisticadas capacidades de script (MICROSOFT, 2012).

2.9 MICROSOFT SQL SERVER COMPACT EDITION

O *SQL Server Compact* permite que você crie bancos de dados compactos que podem ser implantados em computadores desktop e em dispositivos inteligentes. Ele compartilha um modelo de programação comum com outras versões do *SQL Server* para desenvolver aplicativos nativos e gerenciados. O *SQL Server Compact* oferece funcionalidade de banco de dados relacional: uma fonte de dados robusta, um processador para otimização de consultas, e conectividade confiável e escalonável.

O suporte de desenvolvimento para o *SQL Server Compact* é fornecido pelo *Visual Studio*. O suporte de administração de banco de dados é fornecido pelo *SQL Server*. Você poderá acessar bancos de dados do *SQL Server Compact* armazenados em um dispositivo ou em um computador desktop usando o *SQL Server Management Studio* no *SQL Server*. Poderá ainda criar aplicativos gerenciados usando o *Microsoft Visual Basic* ou o *Visual C#*, ou usar o *Visual C++* para criar aplicativos nativos (MSDN, 2012).

Nenhuma instalação ou permissões de segurança adicionais são necessárias para que ele seja executado. Você não precisa ter uma conta de administrador na máquina. Basta copiar sua aplicação em qualquer servidor e ela irá funcionar.

O SQLCE trabalha com APIs de dados existentes baseadas no .NET, e suporta uma sintaxe de consulta compatível com o *SQL Server*. Isto significa que você pode usar APIs de dados existentes como ADO.NET, bem como ORM (*Object Relational Mapping*) de alto nível como o *NHibernate*.

2.10 UML

A *Unified Modeling Language* (UML) é uma linguagem visual utilizada na modelagem de sistemas computacionais orientado a objetos. O objetivo da UML é auxiliar na definição das características do software, como seus

requisitos, comportamento, suas estruturas lógicas e dinâmicas; e seus processos e necessidades físicas em relação ao hardware sobre o qual o sistema deverá ser implementado (GUEDES, 2004).

Para a modelagem do sistema, foram utilizados os seguintes diagramas: Caso de Uso, Classes e de Sequência.

2.10.1 Diagrama de Caso de Uso

É o diagrama “mais geral” e “informal” da UML. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar e suas possíveis intervenções junto a este (como usuários).

2.10.2 Diagrama de Classes

É o diagrama muito utilizado e importante da UML, servindo de apoio para a maioria dos outros diagramas. O Diagrama de Classes define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos contidos em cada classe, além de estabelecer como as classes se relacionam e trocam informações entre si (GUEDES, 2004).

2.10.3 Diagrama de Sequência

Este diagrama preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um Caso de Uso definido pelo diagrama de mesmo nome e apoia-se no Diagrama de Classes para determinar os objetos das classes envolvidas em um processo (GUEDES, 2004).

2.11 STARUML

StarUML é uma ferramenta case que permite manipular e criar os diversos diagramas da UML.

Na Figura 1 é apresentada uma imagem da ferramenta *StarUML*.

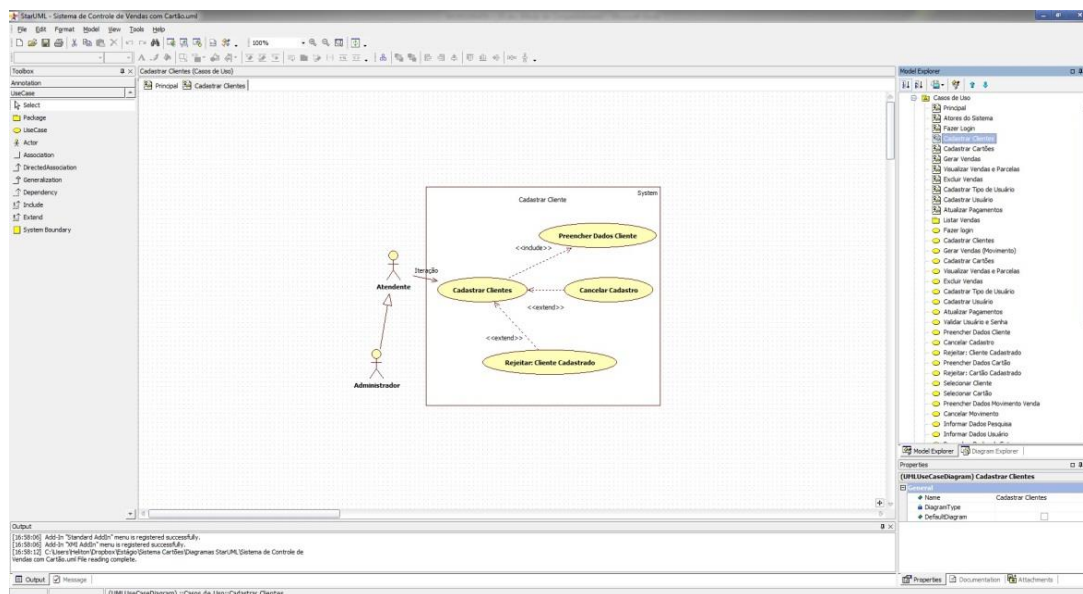


Figura 1 – StarUML

2.12 PADRÕES DE PROJETO

Os padrões de projeto de software ou padrões de desenho de software, também muito conhecido pelo termo original em inglês: *Design Patterns*, descrevem soluções para problemas recorrentes no desenvolvimento de sistemas de software orientados a objetos. Estabelecem um nome e define o problema, a solução, quando aplicar esta solução e suas consequências. Visam facilitar a reutilização de soluções de desenho, isto é, soluções na fase de projeto do *software*, sem considerar reutilização de código. São utilizados em sistemas para melhorar a organização interna do código e facilitar sua manutenção e extensão. Os padrões de projeto nos oferecem soluções prontas para utilizarmos em determinados problemas que podemos enfrentar quando desenvolvemos um *software* orientado a objetos.

Muitas vezes apresentam uma solução comum para determinado problema, a solução é descrita e feita de uma forma abstrata, permitindo o

engenheiro determinar detalhes e algumas especificações de um Padrão de Projeto.

A utilização de Padrões de Projeto possibilita várias vantagens no desenvolvimento de *software* como: Diminuição do processo de aprendizagem de um novo engenheiro de software dentro de um projeto deixando a ação do mesmo muito mais eficiente em muito menos tempo do que os outros modelos de desenvolvimento. É um excelente processo de comunicação para determinar as soluções a serem aplicadas, bem como possibilita a reutilização e customização em projetos de desenvolvimento.

Os Padrões de Projeto estão divididos em três tipos, sendo eles: Criacional, Estrutural e Comportamental.

2.12.1 Padrão Singleton

O Padrão *Singleton* é um padrão de criação de objetos, onde ele é usado para garantir que uma classe tenha somente uma instância e fornecer um ponto global de acesso para a mesma.

Como a classe *Singleton* encapsula a sua única instância, possui total controle sobre como e quando os clientes a acessam. Este padrão representa uma melhoria em relação ao uso de variáveis globais. Ele evita a poluição do espaço de nomes com variáveis globais que armazenam instancias únicas.

Como exemplo de uso, quando o sistema pode ter apenas um gerenciador de janelas, ou gerenciador de impressão, ou então um único ponto de acesso ao banco de dados.

Na Figura 2 é mostrado um simples exemplo da implementação do padrão *Singleton*.

```
public static CtrlSession Instancia
{
    get
    {
        if (instancia.IsNull())
            instancia = new CtrlSession();

        return instancia;
    }
}
```

Figura 2 – Exemplo de implementação do Padrão *Singleton*

2.12.2 Padrão Repository

Padrão *Repository*, ou um “repositório”, é uma camada na aplicação que atua como um gerenciador de coleções de objetos (carregar, contar, excluir, ...). Este funciona para mediar a pesquisa de objetos de domínio entre a camada de domínio e a camada de mapeamento de dados usando uma interface semelhante à de uma coleção. O *Repository* isola a lógica de acesso a dados de qualquer outra lógica da aplicação, permitindo que a mesma seja projetada com o mínimo de dependência de recursos de infraestrutura de acesso a dados.

Na Figura 3, é apresentado um Diagrama de Sequência mostrando como funciona o *Repository*.

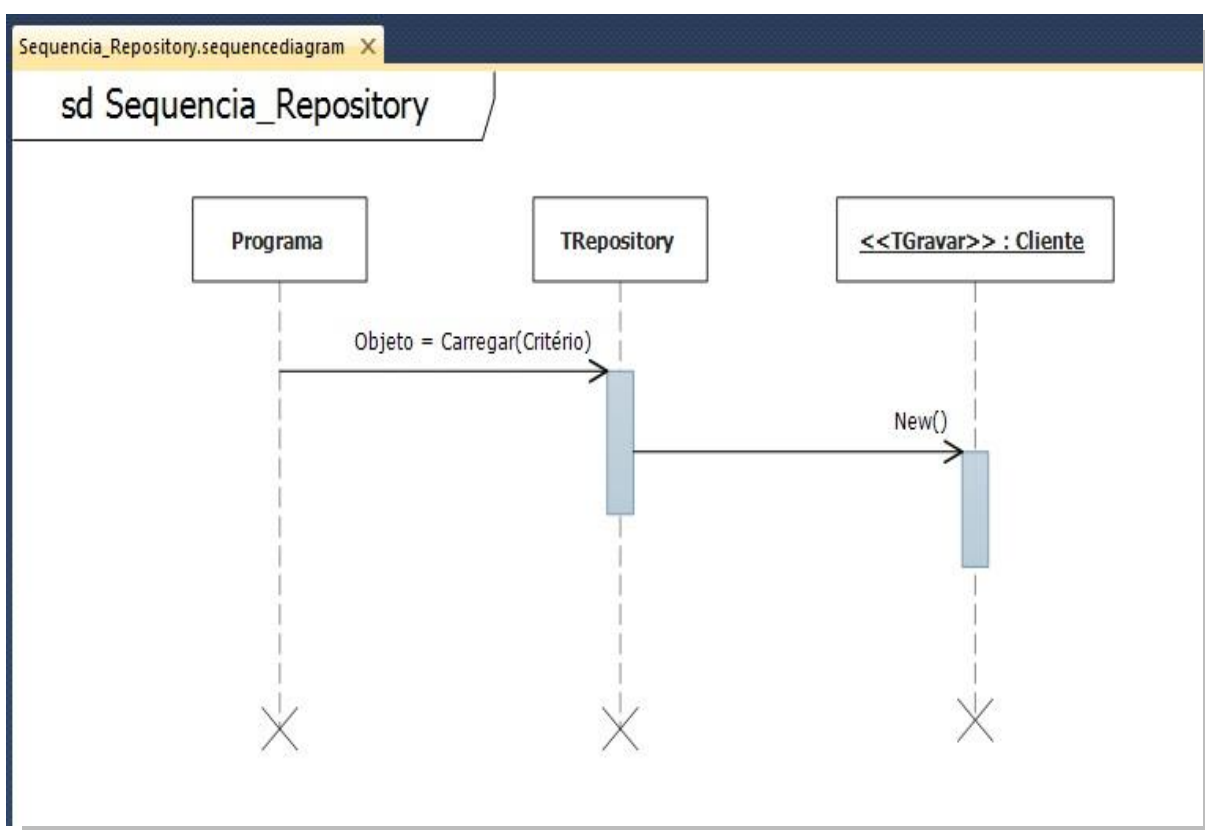


Figura 3 – Diagrama de Sequencia do Repository

Na Figura 4 é apresentado a classe base do Padrão Repository.


```

namespace Repository.Repositorios.Base
{
    public class RepositorioBase<T>
    {
        #region Propriedades

        public ISession Sessao { get; set; }

        #endregion

        #region Métodos Públicos

        public void Delete(T entity)
        {
            Sessao.Delete(entity);
            Sessao.Flush();
        }

        public T SalvarOuAtualizar(T entity)
        {
            using (ITransaction transacao = Sessao.BeginTransaction())
            {
                Sessao.SaveOrUpdate(entity);

                transacao.Commit();

                return entity;
            }
        }

        public T ObterEntidadePeloId(long? id)
        {
            return Sessao.Get<T>(id);
        }

        public IList<T> ObterTodasAsEntidades()
        {
            return Sessao.CreateCriteria(typeof(T)).List<T>();
        }

        #endregion
    }
}

```

Figura 4 - Classe Base do Padrão Repository

Na Figura 5 é apresentada a classe de repositório de clientes.

```

public class RepositorioCliente : RepositorioBase<ICliente>, IRepositorio
Cliente
{
    public IList<ICliente> RetornaTodosOsClientes()
    {
        string sql = "select * from CAD_Cliente";
    }
}

```

```

ISQLQuery query = this.Sessao.CreateSQLQuery(sql);

        query.SetResultTransformer(Transformers.AliasToBean<Cliente>());

        return query.List<ICliente>();
    }
}

```

Figura 5 - Classe de repositório de clientes

Na Figura 6 é apresentado um trecho de código da classe Fabrica de Repositórios.

```

public class RepositorioFactory : IRepositorioFactory
{
    #region Atributos

    private static RepositorioFactory instanciaSessao;

    private static RepositorioFactory instanciaSessaoControlada;

    #endregion

    #region Singleton

    public static IRepositorioFactory GetInstance()
    {
        if (instanciaSessao.IsNull())
            instanciaSessao = new RepositorioFactory();

        AdicionarSessaoParaOsRepositorios(NHibernateHelper.Instance.OpenSession
());

        return instanciaSessao;
    }

    public static IRepositorioFactory GetInstance(ISession sessao)
    {
        if (instanciaSessaoControlada.IsNull())
            instanciaSessaoControlada = new RepositorioFactory();

        AdicionarSessaoParaOsRepositoriosControlados(sessao);

        return instanciaSessaoControlada;
    }

    #endregion

    #region Contrutor

    private RepositorioFactory()
    {
        this.RepositorioAluno = new RepositorioAluno();
        this.RepositorioUsuario = new RepositorioUsuario();
    }

    #endregion
}

```

```

#region Propriedades

    public IRepositoryAluno RepositorioAluno { get; private set; }

    public IRepositoryUsuario RepositorioUsuario { get; private set; }

#endregion

#region Métodos Privados

    private static void AdicionarSessaoParaOsRepositoriosControlados(ISession
sessao)
    {
        instanciaSessaoControlada.RepositorioAluno.Sessao = sessao;
        instanciaSessaoControlada.RepositorioUsuario.Sessao = sessao;
    }

    private static void AdicionarSessaoParaOsRepositorios(ISession sessao)
    {
        instanciaSessao.RepositorioAluno.Sessao = sessao;
        instanciaSessao.RepositorioUsuario.Sessao = sessao;
    }

#endregion

#region Liberação de Recursos

    public void Dispose()
    {
        this.RepositorioAluno.Sessao.Dispose();
        this.RepositorioUsuario.Sessao.Dispose();
    }

#endregion
}

```

Figura 6 - Trecho de código da classe Fabrica de Repositórios

2.12.3 Padrão MVC

O Padrão MVC (*Model-View-Controller*) fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação em três camadas. A arquitetura MVC foi desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interação com o usuário. Usando o padrão MVC fica fácil mapear esses conceitos no domínio de aplicações Web multicamadas.

Na arquitetura MVC o modelo representa os dados da aplicação e as regras do negócio que governam o acesso e a modificação dos dados. O modelo mantém o estado persistente do negócio e fornece ao controlador a

capacidade de acessar as funcionalidades da aplicação encapsuladas pelo próprio modelo.

A arquitetura de três camadas, Figura 7, é uma implementação do modelo MVC. O modelo MVC está preocupado em separar a informação de sua apresentação.

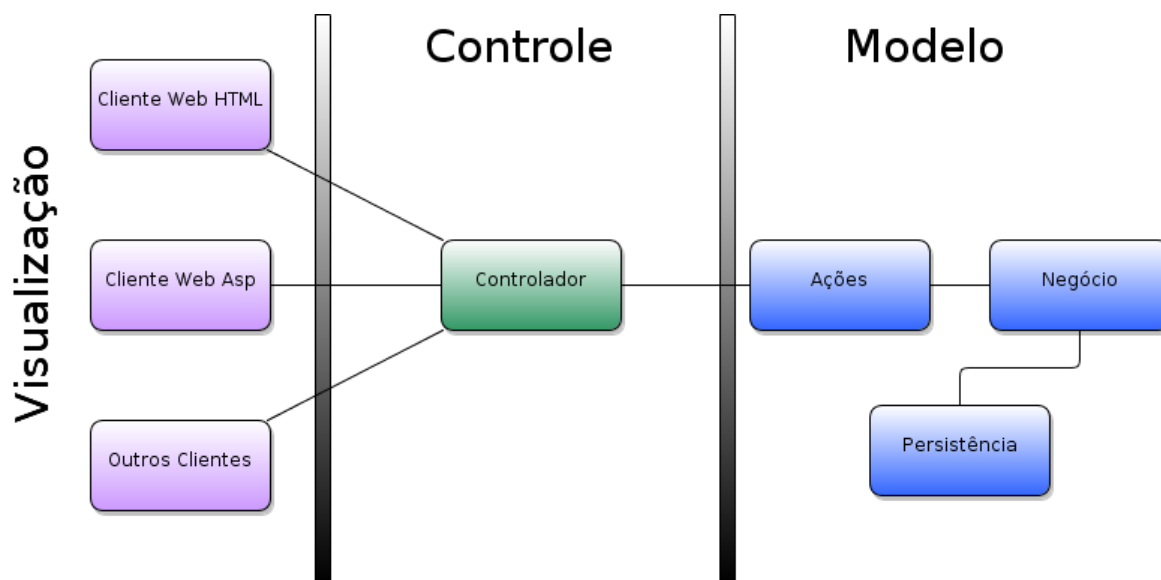


Figura 7 – Modelo MVC

Fonte: [http://msdn.microsoft.com/pt-br/library/dd381412\(v=vs.108\).aspx](http://msdn.microsoft.com/pt-br/library/dd381412(v=vs.108).aspx)

Camada de apresentação ou visualização - Não esta preocupada em como a informação foi obtida ou onde ela foi obtida apenas exibe a informação.

- Inclui os elementos de exibição no cliente: HTML, XML, ASP.
- É a camada de interface com o usuário.
- É usada para receber a entrada de dados e apresentar o resultado.

Camada de lógica da Aplicação - É o coração da aplicação. Responsável por tudo que a aplicação vai fazer.

- Modela os dados e o comportamento por atrás do processo de negócios.
- Se preocupa apenas com o armazenamento, manipulação e geração de dados.
- É um encapsulamento de dados e de comportamento independente da apresentação.

Camada de Controle - determina o fluxo da apresentação servindo como uma camada intermediária entre a camada de apresentação e a lógica.

- Controla e mapeia as ações.

2.13 LINQ (LANGUAGE INTEGRATED QUERY)

A *Language Integrated Query* permite expressões de consulta que se beneficiam de metadados valiosos, verificação de sintaxe em tempo de compilação, tipagem estática e *IntelliSense*, que estavam previamente disponíveis apenas ao código imperativo. A *Language Integrated Query* também permite que uma facilidade de consulta declarativa de propósito geral seja declarada a todas as informações em memória, não apenas as informações derivadas de fontes externas (BOX; HEJLSBERG, 2008).

2.13.1 Query Expressions

A estrutura básica de uma *query expression* inicia com uma cláusula *from* e finaliza com a cláusula *select* ou *group*. A cláusula *from* inicial pode conter nenhuma ou mais cláusulas *from* ou *where*, cada cláusula *from* diz respeito ao objeto LINQ em que as operações serão efetuadas, cada cláusula *where* é um filtro que ignora os itens do resultado. Antes da cláusula *select* ou *group* pode conter a cláusula *orderby* que ordena o resultado esperado, podendo ainda optar pela ordenação descendente ou ascendente (BOX; HEJLSBERG, 2008):

2.14 NHIBERNATE

NHibernate é uma biblioteca baseada em .NET para persistir os objetos para bases de dados relacionais. Baseado em uma ferramenta de persistência de dados do Java, chamado *Hibernate*, o *NHibernate* tem a finalidade de persistir os objetos .NET em uma base de dados relacional subjacente. Isso facilita muito ao invés de escrever códigos SQL dentro e fora da base de

dados, pois o *NHibernate* gera o código SQL necessário, certificando-se que os tipos e o valores são corretamente criados (LINHA DE CÓDIGO, 2012).

O *NHibernate* possui código aberto e sua linguagem de programação é o C#. Ele é um porte do consagrado framework *Hibernate* para Java para a plataforma .NET, dessa forma o *NHibernate* surgiu a partir do *Hibernate* um *framework* muito usado na plataforma Java.

2.14.1 CRUD (*Create, Read, Update e Delete*)

Com o objeto *session* é possível realizar as tarefas básicas (*Create, Read, Update e Delete*) no banco de dados sem a utilização de comando nativos do banco com o SQL, a sessão disponibilizada pelo *NHibernate* possibilita ao programador utilizar as informações do banco utilizando as classes do sistema.

2.15 RECURSOS UTILIZADOS

Os recursos de *hardware* utilizados para a elaboração deste trabalho são definidos nos itens a seguir:

- **Processador:** Intel® Core™ I5-2430M @ 2.40GHz;
- **Memória RAM:** 6 GB;
- **Disco rígido:** 500 GB.
- **Sistema Operacional:** *Windows 7 Professional Edition*.

Além dos recursos de *hardware*, foram utilizados recursos de *software* para a execução das atividades, sendo relacionados na listagem abaixo:

- **Microsoft Office Word 2010:** programa editor de texto utilizado para redigir a monografia;
- **Adobe Photoshop CS2:** *Adobe Photoshop* é um *software* caracterizado como editor de imagens bidimensionais do tipo *raster* (possuindo ainda algumas capacidades de edição típicas dos editores vectoriais) desenvolvido pela *Adobe Systems*.

- **Corel Draw X6:** O *CorelDRAW* é um programa de desenho vetorial bidimensional para design gráfico desenvolvido pela *Corel Corporation*.
- **Flash SlideShow Maker Professional:** *ANVSOFTE Flash SlideShow Maker* é uma ferramenta para desenvolvimento de apresentação de imagens e *slide show*.
- **XMind for Windows:** *XMind* é uma ferramenta de código aberto para a elaboração de mapas mentais.
- **Ganttproject 2.0.10:** O *GanttProject* é um programa criado para agendamento de tarefas de um projeto em suas diferentes etapas baseado no Diagrama de Gantt.
- **Adobe Flash Professional CS6:** O software *Adobe® Flash® Professional CS6* é um ambiente de autoria avançado para criação de conteúdo de animação e multimídia.
- **Dezign For Databses:** A ferramenta *Dezign for Databases* oferece um ambiente de modelagem de dados visual para o desenvolvimento do banco de dados.

O software usa diagramas de entidades de relacionamento (ERDs) para bancos de dados graficamente e gera automaticamente códigos SQL (scripts SQL) para a criação dos esquemas (Meta-Dados) para os bancos de dados mais populares.

3 ARQUITETURA DO SISTEMA

O sistema foi desenvolvido em uma arquitetura em três camadas, Figura 8, sendo:

1. Camada de apresentação: responsável por exibir os dados ao usuário.
2. Camada de negócio: responsável por toda a regra de negócio.
3. Camada de dados: responsável pelo acesso ao banco de dados.

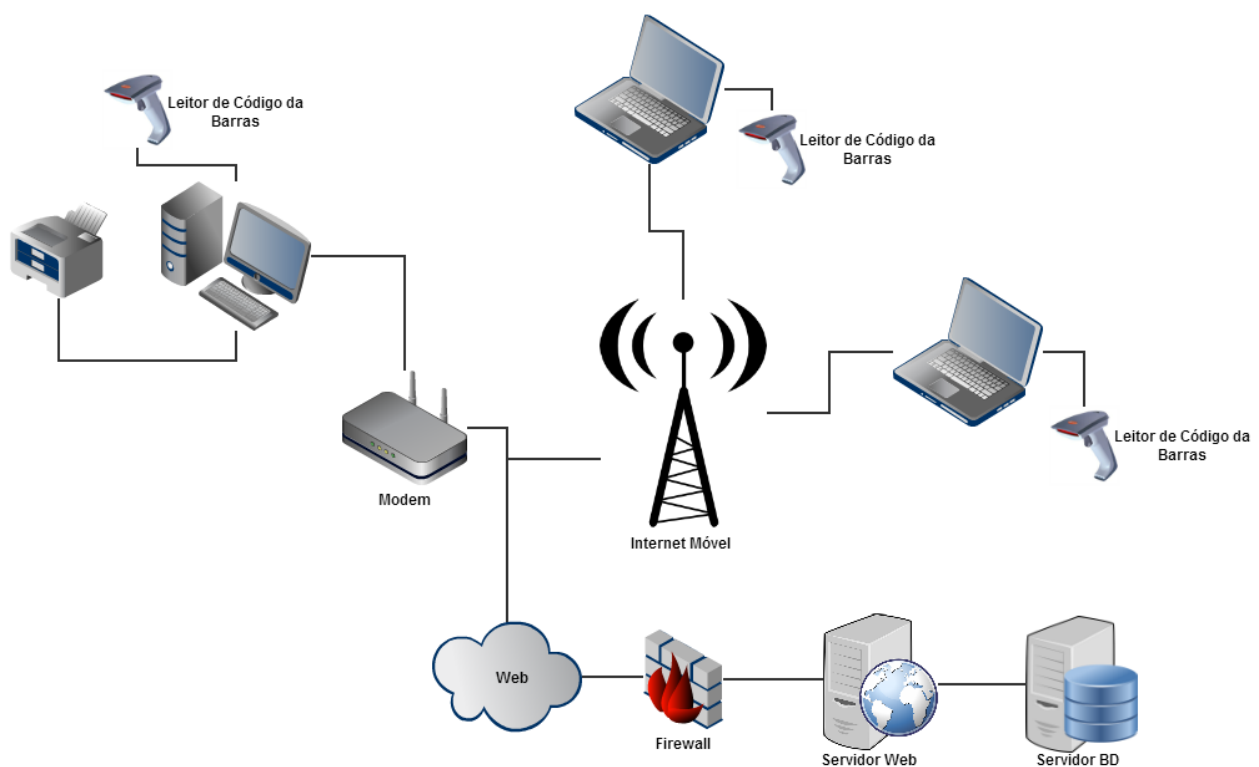


Figura 8 – Arquitetura de Hardware do Sistema

- Servidor BD: É o servidor de banco de dados, tem a função de armazenar os dados manipulados pelos sistemas hospedados no servidor web;

- Servidor Web: Tem a função de processar e armazenar o código dos sistemas *web* hospedados nele. Para rodar a aplicação deve possuir o servidor *Internet Information Services (IIS) 6.0*, com *Microsoft .Net Framework 3.5* instalado;
- *Firewall*: A hospedagem deve garantir a segurança ao acesso e a manipulação das informações armazenadas em seus servidores, isso é dada com um dispositivo de uma rede de computadores que tem por objetivo aplicar uma política de segurança a um determinado ponto da rede.
- Modem Roteador: Conexão com a internet.
- Internet Móvel: Responsável pelo acesso a internet utilizando a tecnologia de internet móvel oferecida pelas operadoras de telefonia móvel.
- Leitores de código de barras: Periférico responsável pela leitura de código de barras.

4 METODOLOGIA DE DESENVOLVIMENTO

Para o desenvolvimento deste trabalho, foi utilizada uma metodologia denominada *eXtreme Programming* ou simplesmente XP, que é uma das mais conhecidas metodologias de desenvolvimento de software que segue os princípios do “Manifesto Ágil”.

Segundo (Scott W. Ambler), a Modelagem Ágil (MA) é um processo baseado na prática que descreve como um modelador ágil deve ser. A Modelagem Ágil é uma metodologia baseada na prática para modelagem e documentação eficazes de sistemas baseados em software. É um conjunto de práticas guiadas por princípios e valores para profissionais de software aplicar em seu dia a dia. A MA não é um processo prescritivo, ou seja, não define procedimentos detalhados sobre como criar um determinado tipo de modelo.

De acordo com Astels (2002), o processo XP, é um método bastante flexível e oferece a possibilidade da comunicação com o usuário ser mais intensa que em outros métodos, devendo os usuários estarem sempre disponíveis para tirarem dúvidas, rever requisitos e atribuir propriedades, utilizando-se de sistema de nomes, em vez de termos técnicos para facilitar a comunicação.

A Figura 9 mostra o ciclo de vida do XP.

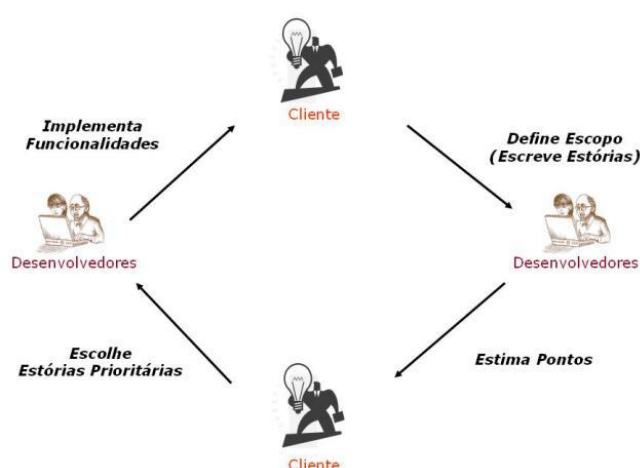


Figura 9 – Ciclo de vida de uma história em projeto XP

Fonte: <http://www.devmedia.com.br/planejando-seu-projeto-com-extreme-programming-parte-i/4273>

De acordo com (Beck, 2000), a estratégia XP é que qualquer um pode desenhar o que desejar, mas assim que for levantada uma questão que possa ser respondida com código, o(s) desenvolvedor(es) deverão criar um código para respondê-la, ou seja, a busca de um retorno rápido para descobrir se o que está sendo criado está acertando o alvo seguindo a prática da modelagem ágil “Comprove o Código”.

Por ser uma metodologia bastante flexível e adaptável, para o desenvolvimento deste trabalho foi utilizada uma adaptação de um modelo do (Beck, 2000) alterando algumas de suas fases e suas determinadas etapas, a fim de proporcionar um melhor rendimento, controle e planejamento do desenvolvimento do projeto.

A definição da metodologia sugere que o processo de Análise de Requisitos possui como principal objetivo o estudo de viabilidade do produto, a identificação e Análise dos Riscos do negócio e do projeto. Define uma arquitetura que ofereça suporte estável à implementação, correspondendo as atividades de exploração e planejamento.

Para este trabalho, o processo de desenvolvimento foi dividido em três fases: Exploração, Manutenção e Transição. Dentro deste mesmo processo, ainda existem três subfases que compõem a Fase de Manutenção, a qual é responsável pelas iterações para entrega, onde ao fim de cada iteração, o usuário em reunião com o desenvolvedor recebe uma versão mais recente do software para avaliação e teste, de tal modo retornando um “*feedback*” no qual o desenvolvedor faz um trabalho de identificação de tarefas, tarefas inacabadas, defeitos, novas funcionalidades, avaliando com precisão cada história de usuário para uma próxima iteração. As três subfases que compõem a Fase de Manutenção são: Planejamento, Iterações para Entrega e Produção.

As fases no desenvolvimento do sistema podem ocorrer iterativamente, pelo fato de ser possível transitar entre a Fase de Exploração e Manutenção.

Este ciclo de vida está melhor definido e visível na Figura 10 – Ciclo de Vida XP.

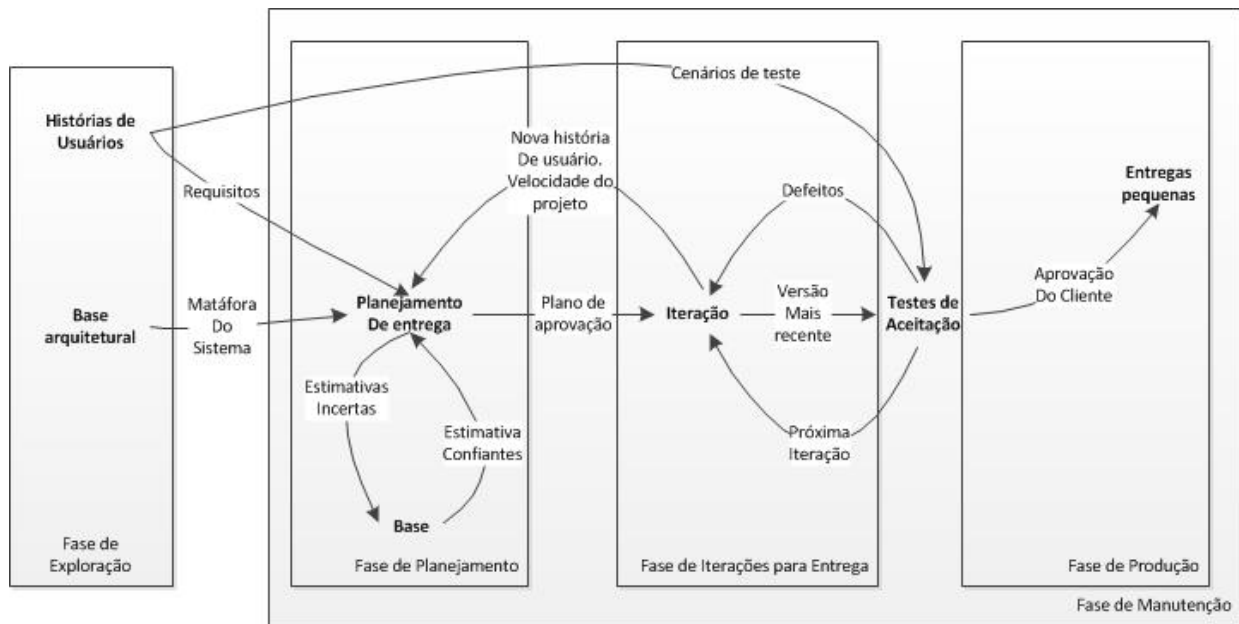


Figura 10 – Ciclo de Vida XP
 Fonte: AMBLER, 2004, p. 185.

A documentação do projeto possui alguns diagramas da UML, como os diagramas de casos de uso, classes e sequência. Essa documentação somente tem início quando é definido o conjunto de requisitos a serem atendidos na versão em desenvolvimento.

Cada uma destas fases e seus artefatos serão explicados em “Aplicação do Processo de Desenvolvimento” no apêndice B deste documento.

5 CONCLUSÃO

Este capítulo conclui o trabalho, indicando trabalhos futuros, além de expor as considerações finais em relação aos objetivos pretendidos.

5.1 TRABALHOS FUTUROS

Para estudos futuros espera-se incluir no sistema uma nova funcionalidade que seria a de impressão de boleto bancário e junto a isso, uma nova categoria de usuários no sistema também será criada, nos quais estes usuários seriam os contratantes do serviço da empresa, possibilitando a estes usuários obter determinadas funcionalidades, como algumas:

- Visualizar algumas informações do seu contrato como: valor, pendências, data de início e término, etc.;
- Verificar pagamentos e datas em que foram efetuados;
- Verificar histórico de frequência;
- Possibilidade de imprimir o boleto bancário para pagamento.

5.2 CONSIDERAÇÕES FINAIS

Este projeto de *software* objetivou demonstrar metodologias e tecnologias para o desenvolvimento de um sistema para ajudar no controle do processo de gestão de empresas de transporte de alunos. Ao final deste trabalho pode-se considerar que ocorreu o cumprimento do escopo delimitado, sendo que tanto a empresa quanto o desenvolvedor do sistema conseguiram identificar as necessidades, fazendo com que este projeto conseguisse chegar a sua fiel conclusão. Através deste trabalho consegue-se identificar a real necessidade da empresa em obter um software que a auxilie. Este projeto servirá de auxílio para empresas que atuarem na área de transporte de alunos, turismo e fretamento de veículos, contribuindo para um melhor controle e rapidez das informações e automatização de processos, tornando-as mais robustas e precisas, e ainda permitindo a empresa fazer a divulgação de seus trabalhos. Pode-se dizer que o desenvolvimento de um sistema, que auxilie e

agilize os processos de negócio de uma empresa, requer tempo e muita dedicação da equipe de desenvolvimento. O desenvolvimento do sistema foi realizado utilizando algumas tecnologias e metodologias vistas ao longo do curso e que foram consideradas de grande importância para que o projeto tenha seguido os padrões atuais de desenvolvimento, fazendo com que o mesmo tenha obtido resultados satisfatórios após a sua implantação. Para tanto, foram realizadas pesquisas bibliográficas visando alcançar o maior conhecimento possível a respeito das tecnologias e metodologias necessárias para o desenvolvimento do sistema. Durante as pesquisas foram identificadas algumas tecnologias que poderiam ter sido utilizadas neste sistema, mas não puderam ser implementadas por motivo de falta de um conhecimento mais aprofundado sobre tais tecnologias e carência de tempo hábil para aprendizado. Conclui-se que com o desenvolvimento do projeto, muitos desafios foram encontrados, mas agregados a eles também vieram vários benefícios, tanto para o crescimento profissional e intelectual do desenvolvedor.

Portanto, foi de grande valia ter contribuído para o desenvolvimento desta área, a qual está sendo muito explorada atualmente e que tende muito a crescer.

REFERÊNCIAS

ALFREDO, L. Como Programar com ASP.NET e C#. 2º Edição. São Paulo: Novatec, 2010. 55p.

AMBLER, Scott W. Modelagem Ágil: Práticas eficazes para a Programação eXtrema e o Processo Unificado. Tradução de Acauan Fernandes. Porto Alegre: Bookman, 2004.

ASTELS, Dave; Miller, Granville; Novak, Miroslave. Extreme programming: Guia Prático. Tradução de Kátia Roque. Rio de Janeiro: Ed. Campus, 2002.

BECK, K. Programação Extrema (XP) Explicada: Acolha as mudanças. Porto Alegre: Bookman, 2004.

BRASIL. Código de Defesa do Consumidor – Nova ed. ver., atual. e ampli. com o Decreto nº 2.181, de 20 de Março de 1.997- Brasília, : Ministério da Justiça, 2010. 124p.

DEV MEDIA **Componentes DevExpress**. Disponível em: <<http://www.devmedia.com.br/curso/componentes-devexpress/178>> Acesso em: 10 jan. 2013.

DEV MEDIA **Componentes que Fazem a Diferença**. Disponível em: <<http://www.devmedia.com.br/devexpress/445>> Acesso em 11 jan. 2013.

DEV MEDIA **Planejando seu Projeto com Extreme Programming**. Disponível em: <<http://www.devmedia.com.br/planejando-seu-projeto-com-extreme-programming-parte-i/4273>> Acesso em 10 nov. 2012.

DEV MEDIA **SQL Server Compact**. Disponível em

<[http://msdn.microsoft.com/pt-br/library/cc645984\(v=SQL.100\).aspx](http://msdn.microsoft.com/pt-br/library/cc645984(v=SQL.100).aspx)> Acesso em 11 out. 2012.

DEV MEDIA **Testes de software - Níveis de testes.** Disponível em <<http://www.devmedia.com.br/testes-de-software-niveis-de-testes/22282#ixzz2M3fM5hyP>> Acesso em 05 dez. 2012.

FREEMAN, Eric; FREEMAN, Elizabeth. Use a Cabeça! Padrões de Projeto (Design Patterns). Rio de Janeiro: Alta Books, 2005.

GAMMA, E. Padrões de Projeto: Soluções reutilizáveis de software orientado a objetos / Erick Gamma, Richard Helm, Ralph Johnson e John Vlissides; trad. Luiz A. Meirelles Salgado. Porto Alegre: Bookman, 2000.

GUEDES, Gilleanes T.A. UML – Uma Abordagem Prática. São Paulo: Novatec, 2004.

LINHA DE CÓDIGO: **Guia Inicial do NHibernate.** Disponível em: <<http://www.linhadecodigo.com.br/artigo/546/net-guia-inicial-do-hibernate.aspx>> Acesso em: 15 jun. 2012.

MACORATTI **Testes em desenvolvimento de Software.** Disponível em <http://www.macoratti.net/tst_sw1.htm> Acesso em 05 dez. 2012.

MICROSOFT. **Microsoft SQL Server 2008 Express.** Disponível em <<http://www.microsoft.com/pt-br/download/details.aspx?id=1695>> Acesso em 10 out. 2012.

MICROSOFT. **Microsoft SQL Server 2008 Express.** Disponível em <<http://www.microsoft.com/pt-br/download/details.aspx?id=7593>> Acesso em 10 out. 2012.

MICROSOFT. **Sobre o Expression Blend.** Disponível em
<[http://msdn.microsoft.com/en-us/library/cc296376\(v=expression.40\).aspx](http://msdn.microsoft.com/en-us/library/cc296376(v=expression.40).aspx)>
Acesso em 02 out. 2012.

MSDN **Introdução a Linguagem C#.** Disponível em:
<[http://msdn.microsoft.com/pt-br/library/z1zx9t92\(v=VS.90\).aspx](http://msdn.microsoft.com/pt-br/library/z1zx9t92(v=VS.90).aspx)> Acesso em
20 nov. 2012.

MSDN **Introdução a Plataforma .Net.** Disponível em:
<<http://msdn.microsoft.com/pt-br/aa702903.aspx>> Acesso em 29 nov. 2012.

MSDN **Introdução ao Visual Studio.** Disponível em:
<[http://msdn.microsoft.com/pt-BR/library/vstudio/fx6bk1f4\(v=vs.90\).aspx](http://msdn.microsoft.com/pt-BR/library/vstudio/fx6bk1f4(v=vs.90).aspx)>
Acesso em: 10 jan. 2013.

MSDN **Visão Geral do ASP.NET MVC.** Disponível em:
<[http://msdn.microsoft.com/pt-br/library/dd381412\(v=vs.108\).aspx](http://msdn.microsoft.com/pt-br/library/dd381412(v=vs.108).aspx)> Acesso em
20 set. 2012.

PERKINS, B. Working with NHibernate 3.0. Boulevard Indianapolis: Wiley
Publishing, Inc. 2011. 228p

PRESSMAN, ROGER S., Engenharia de Software - (6ª edição), São Paulo, Ed.
McGrawHill, 2006.

APÊNDICE A: PLANO DE DESENVOLVIMENTO

SUMÁRIO

1 INTRODUÇÃO.....	44
1.1 PROPÓSITO.....	44
1.2 ESCOPO.....	44
1.3 VISÃO GERAL.....	44
2 VISÃO GERAL DO PROJETO	46
2.1 PROPÓSITO DO PROJETO, ESCOPO E OBJETIVOS.....	46
2.2 SUPOSIÇÕES E RESTRIÇÕES.....	46
3 ORGANIZAÇÃO DO PROJETO	47
4 GERENCIAMENTO DO PROJETO	48
4.1 ESTIMATIVAS	48
4.2 RISCOS	48
4.3 TEMPO DISPONÍVEL:.....	49
4.4 TEMPO DE CONCLUSÃO:.....	49
4.5 TECNOLOGIA DESEJADA INVIÁVEL.....	50
4.6 MUDANÇA DE REQUISITOS.....	50
5 CONTROLE DE VERSÕES E BACKUP DO PROJETO	51

1 INTRODUÇÃO

O Plano de Desenvolvimento de Software é um artefato composto e abrangente que inclui vários artefatos separados, desenvolvidos durante a Fase de Iniciação, e é mantido durante todo o projeto.

1.1 PROPÓSITO

O propósito deste plano de desenvolvimento é reunir todas as informações necessárias para o gerente do projeto, a fim de planejar a programação de datas, seguir metas, controlar recursos e o desenvolvimento das atividades.

É o documento de nível mais alto gerado e nele é abordado todo o planejamento e foco dado ao desenvolvimento do projeto.

1.2 ESCOPO

O *software* produzido pelo projeto denominado Sistema de Transporte Escolar, tem por objetivo controlar as contas a receber e também controlar o uso de seus veículos pelos clientes da empresa, garantindo a empresa uma maior rapidez e segurança dos dados cadastrados, disponibilizando um melhor controle de seus clientes, funcionários e carros, oferecendo a possibilidade de geração de históricos de frequência de todos os clientes que estarão utilizando seus carros, garantindo um melhor controle das contas a receber, de acordo com os contratos fechados com os clientes e também possibilitar a emissão de relatórios.

1.3 VISÃO GERAL

Nas sessões a seguir serão abordadas uma visão geral do projeto apontando sua finalidade, escopo e objetivos, sua organização definindo

papeis e responsabilidades para o gerente do projeto, a parte de gerenciamento do projeto contendo riscos e métricas do sistema.

2 VISÃO GERAL DO PROJETO

2.1 PROPÓSITO DO PROJETO, ESCOPO E OBJETIVOS

O *software* Sistema de Transporte Escolar irá auxiliar a empresa que não possui um programa que controla o uso de seus clientes em seus e também não faz o controle de suas contas a receber de acordo com os contratos fechados.

2.2 SUPOSIÇÕES E RESTRIÇÕES

O *software* proposto não fará controle financeiro da empresa, será feito apenas o controle das contas a receber de seus clientes e não dará a possibilidade do cliente pagar suas respectivas faturas *online*. Tratando-se de um sistema web, será necessário terem instalados navegadores nas máquinas e as mesmas estarem conectadas à internet.

A integridade dos dados será controlada pelo servidor do site em que o *software* será hospedado.

O *software* será construído levando em consideração que a infraestrutura de rede da empresa esteja pronta e configurada.

3 ORGANIZAÇÃO DO PROJETO

A tabela abaixo apresenta os papéis seus devidos responsáveis.

FUNÇÃO	RESPONSÁVEL
GERENTE	Heliton
DESENVOLVEDOR	Heliton
TESTADOR	Heliton
DOCUMENTADOR	Heliton

Tabela 1 – Papéis e Responsabilidades

4 GERENCIAMENTO DO PROJETO

4.1 ESTIMATIVAS

O projeto inicialmente possui um cronograma que aponta 24 semanas para ser concluído. A partir da data de produção deste documento restam 20 semanas para término do prazo para entrega do projeto. Com base neste tempo seguem abaixo as metas a cumprir.

- O primeira versão do projeto tem que está pronto no dia 20 de Abril de 2013.
- Cada iteração irá compreender a confecção ou atualização dos diagramas de um caso de uso, sequência e banco de dados.
- Cada iteração deverá durar no máximo dez dias útil.
- O teste de cada iteração deverá ocorrer em no máximo dois dias.
- Será destinado um dia útil para correção de eventuais erros encontrados.
- Melhorar o protótipo inicial do sistema.

Ao final da realização de cada iteração o release será testado durante um dia, porém a medida que for surgindo defeitos os mesmos já começarão a serem corrigidos. O tempo máximo estipulado de um dia útil é para que não haja atrasos no início da próxima iteração.

4.2 RISCOS

Um projeto, como a vida, é incerto. Os riscos não são simplesmente identificados; eles devem ser identificados para que sejam previstos e diminuídos se possível, ou para que sejam controlados quando houver poucas estratégias para a sua diminuição.

O risco controla os planos de iteração; as iterações são planejadas considerando riscos específicos na tentativa de limitar o risco ou reduzi-lo. A

lista de riscos é revista periodicamente para avaliar a eficácia das estratégias de diminuição de riscos e, conseqüentemente, orientar as revisões no plano de projeto e nos planos de iteração subseqüentes.

O segredo do gerenciamento de risco é não esperar até que haja risco (e isso passe a ser um problema ou uma falha) para decidir o que fazer em relação a ele. Gerenciar o risco antecipadamente é quase sempre menos dispendioso e penoso do que tentar solucioná-lo depois que virar um fato.

4.3 TEMPO DISPONÍVEL:

Gravidade do Risco: Média.

Probabilidade: Moderada.

Descrição: Excesso de trabalho.

Impactos: Desmotivação, Stress.

Indicadores: Diminuição da produtividade, desestímulo.

Estratégia de Diminuição: Tentar seguir fielmente o cronograma.

Plano de Contingência: Utilizar todo tempo possível ou disponível para o desenvolvimento do projeto.

4.4 TEMPO DE CONCLUSÃO:

Gravidade do Risco: Média.

Probabilidade: Baixa, se o planejamento for bem feito.

Descrição: O tempo para desenvolver o projeto é subestimado.

Impactos: Atraso no projeto.

Indicadores: As metas de tempo não estão sendo cumpridas.

Estratégia de Diminuição: Reavaliação do tempo do projeto.

Plano de Contingência: Esforçar e trabalhar mais tempo no projeto.

4.5 TECNOLOGIA DESEJADA INVIÁVEL

Gravidade do Risco: Média.

Probabilidade: Baixa.

Descrição: Não é possível executar o projeto na tecnologia desejada, pois é necessário tempo para estudo.

Impactos: Atraso no projeto devido ao tempo necessário para estudar as novas tecnologias.

Indicadores: Tempo de conclusão e entrega do projeto com prazo estipulado.

Estratégia de Diminuição: Opção de não utilizar tecnologias desconhecidas.

Plano de Contingência: Não utilizar tecnologia desconhecida.

4.6 MUDANÇA DE REQUISITOS

Gravidade: Média.

Probabilidade: Média.

Descrição: Novos esforços são necessários para a atualização de toda documentação e implementação do projeto.

Impactos: Atraso no projeto devido ao tempo necessário para atualização.

Estratégia de Diminuição: Coleta de requisitos em cada iteração o mais consistente possível.

Plano de Contingência: Não implementar requisitos recém modificados afim de não haver atraso na entrega do projeto.

Os riscos identificados precisam ser monitorados para que se possa saber se a probabilidade de ocorrência está se tornando maior ou menor. Caso é detectado que será certa a sua ocorrência o plano de contingência deverá ser acionado.

5 CONTROLE DE VERSÕES E BACKUP DO PROJETO

A principal função em manter um controle de versão é de armazenar todo o histórico de desenvolvimento, desde o primeiro envio até sua última versão, permitindo que seja possível resgatar uma determinada versão de qualquer data mais antiga, evitando desperdício de tempo no desenvolvimento para desfazer alterações quando se toma algum rumo equivocado.

Com a finalidade de manter um controle de versões e backup do projeto, será utilizado o *Team Foundation Server*, ou simplesmente TFS, que oferece uma série de recursos, como alguns: controle de versões, gerenciamento de mudanças e de compilações.

O TFS permite arquivar alterações pendentes do projeto, assim possibilitando recuperar esses códigos alterados caso aconteça um eventual erro de implementação, arquivos corrompidos ou algum problema de hardware no computador.

**APÊNDICE B: APLICAÇÃO DO PROCESSO DE
DESENVOLVIMENTO**

SUMÁRIO

1 INTRODUÇÃO.....	54
2 FASES DO PROCESSO DE DESENVOLVIMENTO.....	56
2.1 FASE DE EXPLORAÇÃO	56
2.2 FASE DE MANUTENÇÃO	56
2.2.1 Fase de Planejamento	57
2.2.2 Fase de Iterações para Entrega.....	57
2.2.3 Fase de Produção.....	58

1 INTRODUÇÃO

Por meio da imagem abaixo (B.1) – EAP Sistema de Transporte Escolar é explicado como o processo de desenvolvimento XP foi aplicado no desenvolvimento do sistema proposto, e também será apresentando o que foi trabalhado dentro de cada fase.

A imagem mostra a estrutura analítica do projeto, no qual foi elaborada com base na metodologia de desenvolvimento apresentada e demonstra as tarefas realizadas durante o desenvolvimento do sistema. O processo foi dividido em três fases e três subfases no qual foram realizadas dez iterações com todas as atividades de cada fase do processo de desenvolvimento do software em questão.

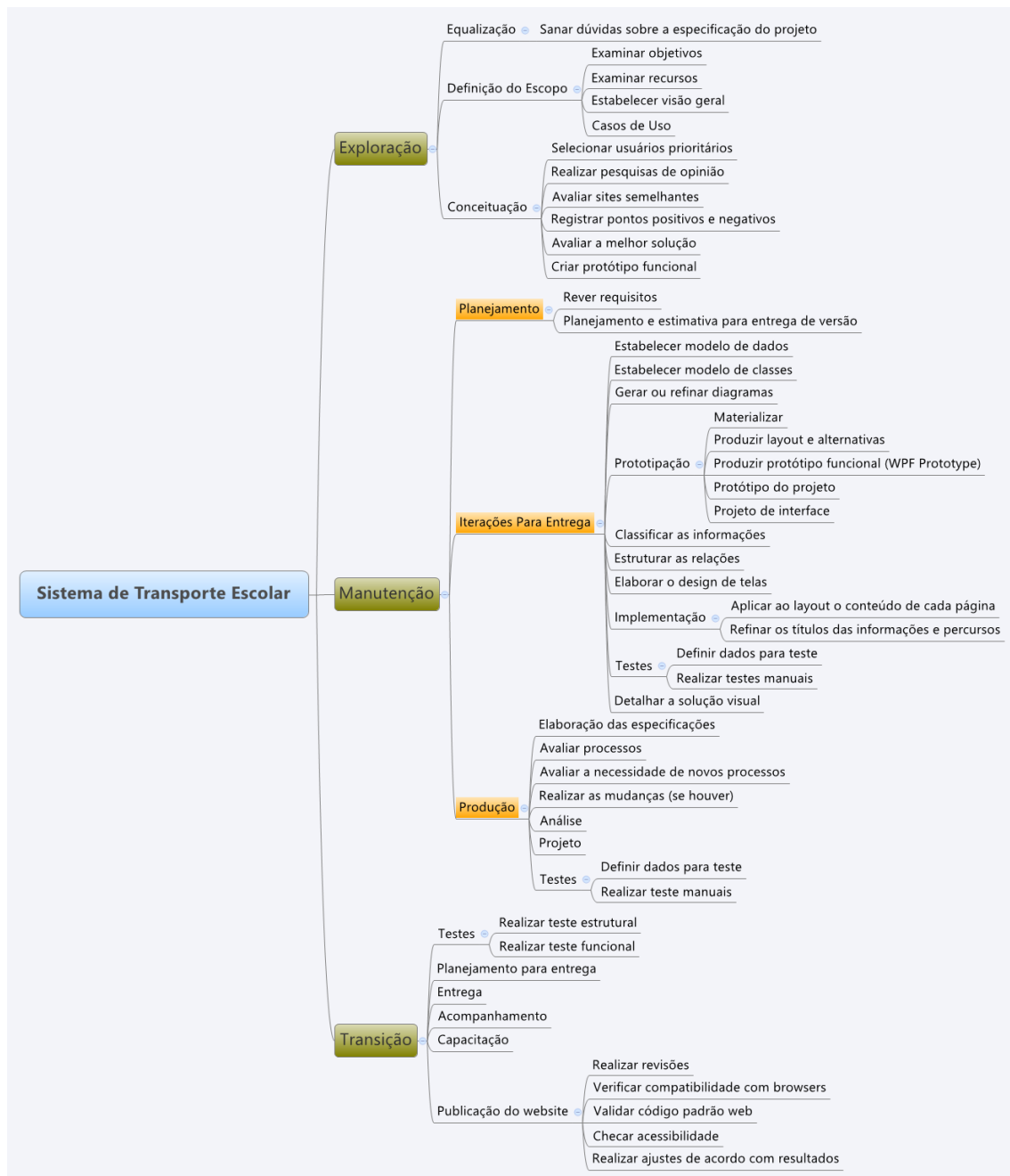


Figura B.1 – EAP SGVC

2 FASES DO PROCESSO DE DESENVOLVIMENTO

2.1 FASE DE EXPLORAÇÃO

A primeira fase pelo qual o projeto XP passa é a Exploração (Beck, 2000). Nela é inclusa o desenvolvimento da base da arquitetura e as primeiras histórias dos usuários. Nesta fase é muito importante coletar material suficiente para fazer uma primeira versão madura do que será o sistema. É nesta fase que é gerado o escopo baseado num conjunto inicial de requisitos.

Os requisitos do sistema que foram coletados de acordo com as histórias de usuários são de alto nível, e como consequência foi necessário um conjunto dessas histórias.

Após a coleta de dados, foi realizada uma filtragem destes utilizando a técnica de *BrainStorm*, onde apenas os dados relevantes foram mantidos. Com o uso desta técnica, foram apresentadas algumas sugestões de melhorias para o projeto, como: a filtragem dos dados que seriam cadastrados; como seriam cadastradas as taxas cobradas após o vencimento da fatura; qual o prazo para pagamento após vencimento da fatura; os dados que serão apresentados ao motorista; definição do layout e conteúdo do *website*; a forma que deveria ser cadastrados os contratos; definir quais relatórios seria relevante; etc.

Os artefatos gerados nesta fase foram: Documento de Coleção de Requisitos no apêndice C; Diagramas de Caso de Uso no apêndice F; Protótipo Funcional do *WebSite* no apêndice D.

2.2 FASE DE MANUTENÇÃO

Esta fase compreende as fases de Planejamento, Iterações para Entrega e Produção para entrega das várias versões evolutivas do sistema.

Neste sistema proposto foram realizadas dez iterações, onde foi possível deixar o *software* bem específico.

Durante esta fase foram definidos alguns pontos de verificação (testes) no intuito de examinar se o sistema atende a determinada funcionalidade

associada a cada história de usuário, sendo que estes testes foram executados manualmente.

Os artefatos gerados são: atualização do documento de Coleção de Requisitos e atualizações dos diagramas.

2.2.1 Fase de Planejamento

Após a fase de Exploração está a de Planejamento (Beck, 2000). Seu principal propósito é de o desenvolvedor e o cliente fazer um planejamento e estimativas para a entrega das versões.

Esses prazos foram discutidos verbalmente, e por isso não foi gerado artefato nesta fase.

2.2.2 Fase de Iterações para Entrega

Segundo (Beck, 2000), esta fase inclui o principal trabalho do projeto XP, geralmente chamado de construção de iterações ou simplesmente de iterações, pois é o momento em que ocorre seu maior trabalho de desenvolvimento, incluindo modelagem, programação, teste e integração.

Sob o ponto de vista da modelagem, o aspecto interessante da iteração é mostrado no aspecto do desenvolvimento da figura B.1.

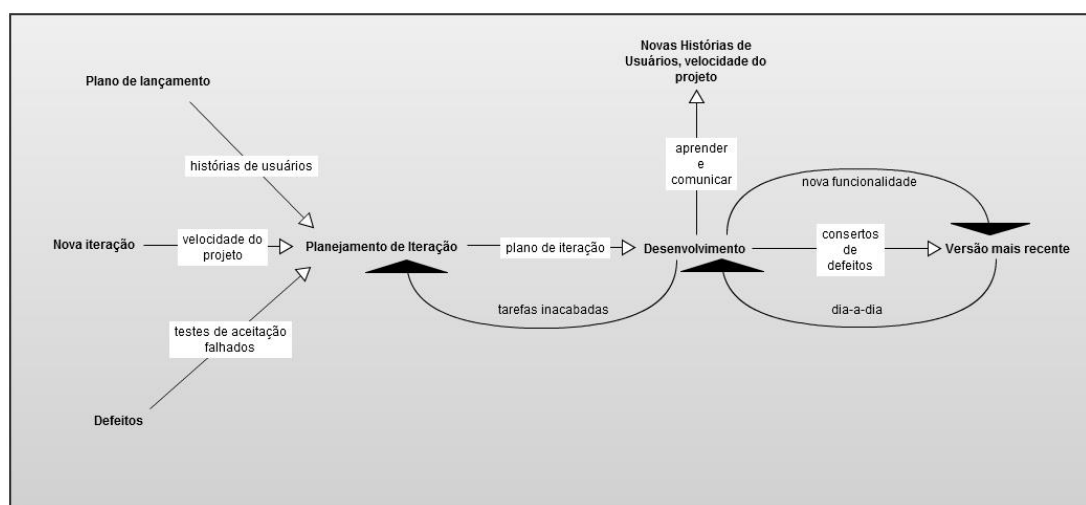


Figura B.2 – Ciclo de vida de uma iteração XP

Fonte: AMBLER, 2004, p. 188.

Os artefatos gerados nesta fase são: diagramas de caso de uso refinados; construção ou refinamento dos diagramas de sequência, de classes, MER (Modelo de Entidade-Relacionamento) e do protótipo o sistema. Os diagramas de sequência e classes constam no apêndice F, já o MER esta no anexo A - Proposta de Trabalho de Conclusão de Curso.

2.2.3 Fase de Produção

A fase de Produção da XP (Beck, 2000) focaliza a garantia de que o software esteja pronto para entrar em produção. Depois de feito os testes de aceitação, e o resultado forem a aprovação do usuário, são produzidos pequenas partes do sistema de acordo com o que foi apresentado na fase anterior ao longo de cada iteração onde juntas estas partes formarão o sistema.

Com o objetivo de confirmar se o *software* está funcionando conforme o esperado, ou seja, prover a confiabilidade de que esteja de acordo com os requisitos, foram efetuadas avaliações da qualidade do *software* através de testes relacionados às necessidades dos usuários, requisitos e processos de negócio, onde possibilitou fornecer informações sobre os riscos da implantação do sistema em um determinado momento. Estes testes foram realizados de forma manual na finalidade de avaliar as funcionalidades acrescentadas dentro de cada iteração, confiabilidade, usabilidade e eficiência.

Os artefatos gerados nesta fase são alguns módulos implementados, versões incompletas do sistema e refinamento nos diagramas UML.

2.3 FASE DE TRANSIÇÃO

O principal objetivo desta fase, é a entrega do sistema completamente funcional ao cliente.

Nesta fase foi feito um plano de implantação, a entrega, acompanhamento e treinamento dos usuários do sistema.

Ainda nesta fase, durante alguns dias também foram executados os testes estruturais e funcionais com a finalidade de avaliar o sistema implantado, no qual foi possível obter um *feedback* dos usuários do sistema, no intuito de detectar defeitos, de tal modo possibilitando fazer ajustes finos necessários, correções e modificações. Mais detalhes sobre os testes executados nesta fase estão no apêndice E.

Os artefatos gerados nesta fase é o sistema concluído de acordo com os requisitos do produto e um manual de instalação do *software*.

APÊNDICE C: COLEÇÃO DE REQUISITOS

SUMÁRIO

1 IDÊNTIFICAÇÃO DOS USUÁRIOS DO SISTEMA.....	63
1.1 HIERARQUIA ENTRE OS USUÁRIOS.....	63
2 REQUISITOS FUNCIONAIS.....	65
3 REQUISITOS NÃO-FUNCIONAIS.....	67

Um requisito é definido como uma condição ou uma capacidade com o qual o sistema deve estar de acordo. Pressman (2006) define que os requisitos fornecem um mecanismo adequado para entender melhor o que o cliente deseja e analisar suas necessidades.

Os requisitos funcionais especificam ações que um sistema deve ser capaz de executar, sem levar em consideração restrições físicas. Geralmente, isso é mais bem descrito em um modelo de casos de uso e em especificação de casos de uso. Os requisitos funcionais especificam, portanto, o comportamento de entrada e saída de um sistema. (Pressman, 2006).

Os requisitos não funcionais descrevem apenas atributos do sistema ou atributos do ambiente do sistema. São as restrições nas funções oferecidas pelo sistema, podendo incluir restrições de tempo, restrições no processo de desenvolvimento, padrões e qualidades globais de um software, como manutenibilidade, usabilidade, desempenho e custos. (Pressman, 2006).

1 IDÊNTIFICAÇÃO DOS USUÁRIOS DO SISTEMA

Usuários Internos: Administrador, Atendente, Motorista.

Usuários Externos: Visitante.

1.1 HIERARQUIA ENTRE OS USUÁRIOS

- Administrador: Somente o administrador terá total acesso ao sistema, podendo assim controlar todos os usuários, clientes, funcionários, contratos, faturas e também efetuar qualquer tipo de manutenção como atualizar algumas informações que serão exibidas no *website*, alterar contratos, valores, etc.
- Atendente: Este nível de usuário poderá cadastrar e visualizar clientes, funcionários e contratos.
- Motorista: Devidamente logado, poderá apenas visualizar os dados dos clientes que estão entrando no carro (Sistema *Desktop*).
- Visitante: Poderá visualizar o conteúdo do *web site* e enviar mensagem de contato.

A Figura C.1 apresenta um diagrama de caso de uso mostrando a herança dos atores do sistema *Web*.

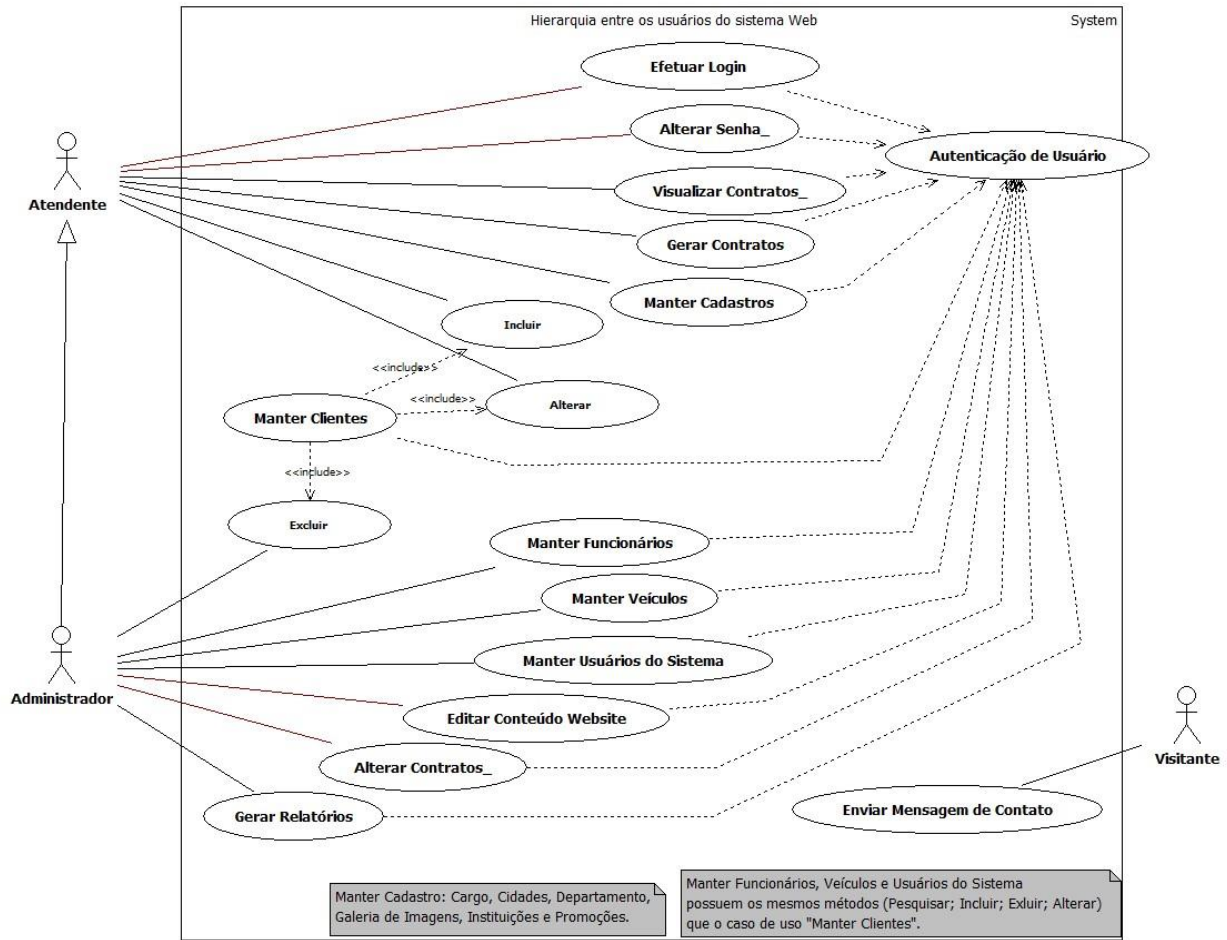


Figura C.1 – Hierarquia entre os usuários do sistema Web.

2 REQUISITOS FUNCIONAIS

Os Requisitos Funcionais do sistema estão organizados com base nas principais funcionalidades do sistema.

Na tabela C.1 são apresentados os requisitos funcionais do sistema.

Identificador	Requisito
RF01	O sistema deve possuir um ou mais Administradores. O objetivo da existência de administradores é controlar os dados e usuários do sistema.
RF02	Permitir a autenticação por meio de <i>login</i> .
RF03	O sistema deverá possuir dois módulos diferentes, um local e um online.
RF04	Deverá haver a sincronia de dados entre os módulos do sistema.
RF05	O módulo local deverá ser capaz de fazer a leitura do código de barras e exibir os dados do cliente.
RF06	O módulo local deverá apresentar ao motorista um sinal de status de cada cliente. (Situação do pagamento do cliente).
RF07	No módulo local o sistema fará um controle dos clientes que entram nos ônibus, gerando automaticamente um histórico de frequência de cada usuário.
RF09	O sistema <i>web</i> possibilitará ao usuário administrador controlar todos os dados cadastrados.
RF10	O sistema <i>web</i> possibilitará ao usuário administrador alterar o status de pagamento dos clientes. (Pago ou pendente).
RF11	O sistema <i>web</i> possibilitará ao usuário administrador criar, editar, excluir e visualizar contratos.
RF12	O sistema <i>web</i> deverá gerar um código de barras para cada cliente cadastrado.
RF13	O sistema <i>web</i> deverá possuir suporte para impressão de carteira de identificação do cliente com os dados cadastrados e seu respectivo código de barras.
RF14	O sistema <i>web</i> deverá possuir suporte para impressão de contratos.
RF15	O sistema <i>web</i> deverá permitir ao administrador alterar alguns

	dados do <i>website</i> como: promoções, etc.
RF16	O sistema <i>web</i> deverá emitir relatórios.
RF17	O sistema deverá possibilitar a alteração da senha de qualquer usuário.

Tabela C.1 – Requisitos Funcionais

3 REQUISITOS NÃO-FUNCIONAIS

Os Requisitos Não-Funcionais estão organizados de acordo com as principais propriedades e restrições do sistema.

Identificador	Requisito	Categoria
RNF01	O sistema deverá trabalhar com a moeda brasileira (Real).	Cultural Político
RNF02	O sistema desenvolvido deverá conter um banco de dados para armazenamento das informações.	Padronização
RNF03	O sistema deverá fazer autenticação dos usuários.	Segurança
RNF04	Os tempos de resposta das consultas não devem ultrapassar 7 segundos.	Desempenho
RNF05	O sistema deverá ter uma interface de fácil entendimento.	Usabilidade
RNF06	A base de dados deve ser protegida para acesso apenas de usuários autorizados	Segurança
RNF07	O software deve ser operacionalizado no sistema Windows XP ou Superior.	Operacional
RNF08	O sistema deverá ter alta disponibilidade.	Confiabilidade

Tabela C.2 – Requisitos Não-Funcionais do Sistema

APÊNDICE D: *Abatur WpfPrototype*

SUMÁRIO

1 ESBOÇO DO MAPA DE FLUXO DO SITE.....	70
2 TELAS DE NAVEGAÇÃO.....	71
2.1 HOME.....	71
2.2 EXCURSÕES.....	71
2.3 FALECONOSCO.....	72
2.4 FRETAMENTO.....	72
2.5 GALERIA.....	73
2.6 GALERIA1.....	73
2.7 LOCALIZAÇÃO.....	74
2.8 LOGIN.....	74
2.9 TRANSPORTEALUNOS.....	75
2.10 MASTERPAGE.....	76

Este documento foi gerado com o *SketchFlow*, que é um dos produtos da família *Expression* da *Microsoft*. Com ele é possível criar protótipos interativos de forma fácil e rápida, diferente de quando fazemos o protótipo da estrutura de um site no *Photoshop* ou mesmo no *Powerpoint*, onde o cliente só terá uma imagem para ver e imaginar o funcionamento. Com o *SketchFlow* é possível criar algo funcional, com os menus funcionando, animações, e tudo que terá no projeto, com a opção do mesmo escrever suas observações, desenhar, sublinhar em qualquer lugar do protótipo.

1 Esboço do Mapa de Fluxo do Site

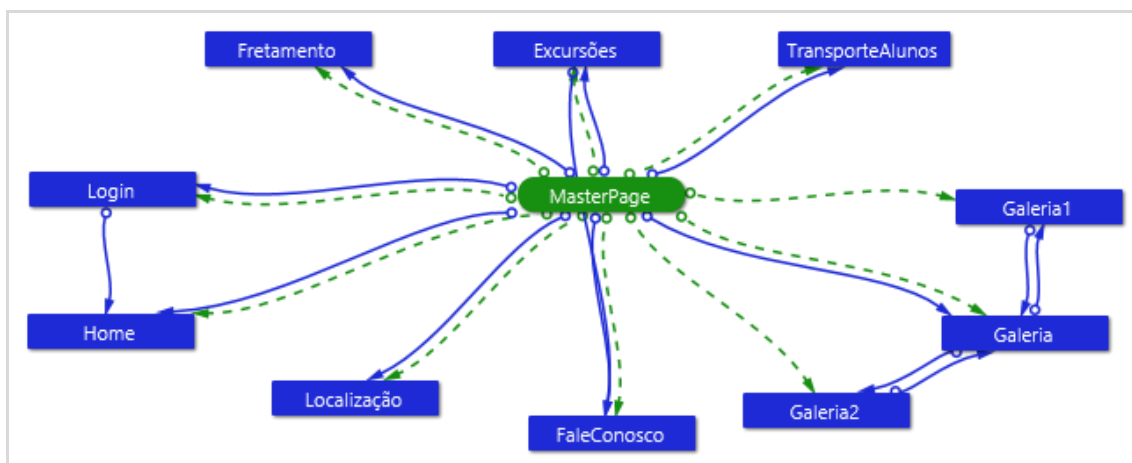


Figura D.1 - SketchFlow Map

2 Telas de Navegação

2.1 Home

A figura D.2: Home esta apresentando uma imagem de como estarão dispostos os itens na página principal do site.

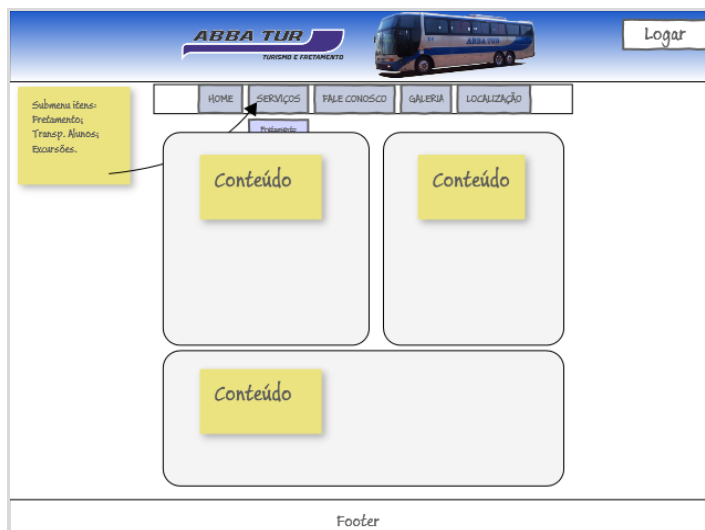


Figura D.2 - Home

2.2 Excursões

A figura D.3: Excursões esta apresentando uma imagem de como serão apresentados os itens de promoções e excursões do site.

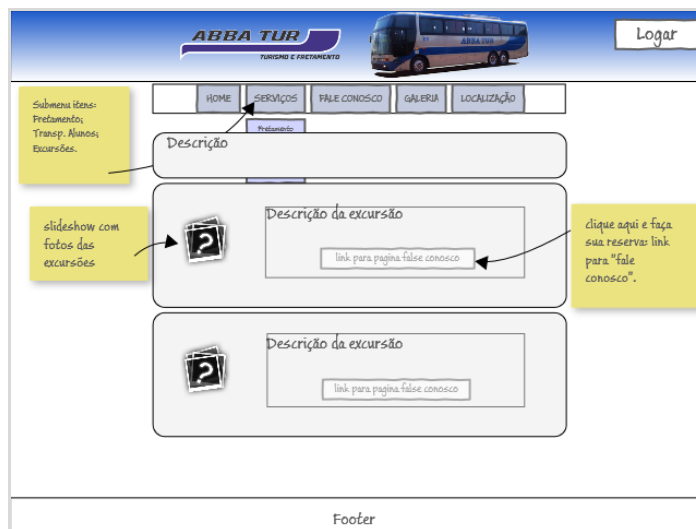


Figura D.3 - Excursões

2.3 FaleConosco

A figura D.4: FaleConosco esta apresentando uma imagem de como será o formulário para contato.

The screenshot displays the 'Fale Conosco' page of the ABBA TUR website. At the top, there is a navigation menu with 'HOME', 'SERVIÇOS', 'FALE CONOSCO', 'GALERIA', and 'LOCALIZAÇÃO'. A 'Logar' button is located in the top right corner. Below the navigation menu, there is a sub-menu on the left with the text 'Submenu itens: Fretamentos, Transp. Alunos, Excursões.' and a yellow box containing the same text. The main content area features a contact form with the following fields: 'NOME:', 'EMAIL:', 'ASSUNTO:', and 'MENSAGEM:'. Below these fields is an 'Enviar' button. The form is titled 'CONTATO' and includes contact information: 'Entre em Contato ou Preencha o Formulário para enviar um email.', 'Rua Dr. Gervásio Moraes, 325 - Sertãoópolis, Pr.', 'Celular(VIVO): (43) 9962-1176', and 'Celular(TIM): (43) 9163-1015'. A 'Footer' is visible at the bottom of the page.

Figura D.4 - FaleConosco

2.4 Fretamento

A figura D.5: Fretamento esta apresentando uma imagem de como estarão dispostas as informações sobre fretamento do site.

The screenshot displays the 'Fretamento' page of the ABBA TUR website. The layout is similar to the previous page, with the same navigation menu and sub-menu. The main content area is a large, empty box with the title 'Fretamento' at the top. The 'Footer' is visible at the bottom of the page.

Figura D.5 - Fretamento

2.5 Galeria

A figura D.6: Galeria esta apresentando uma imagem de como estarão dispostas as imagens na galeria de fotos do site.

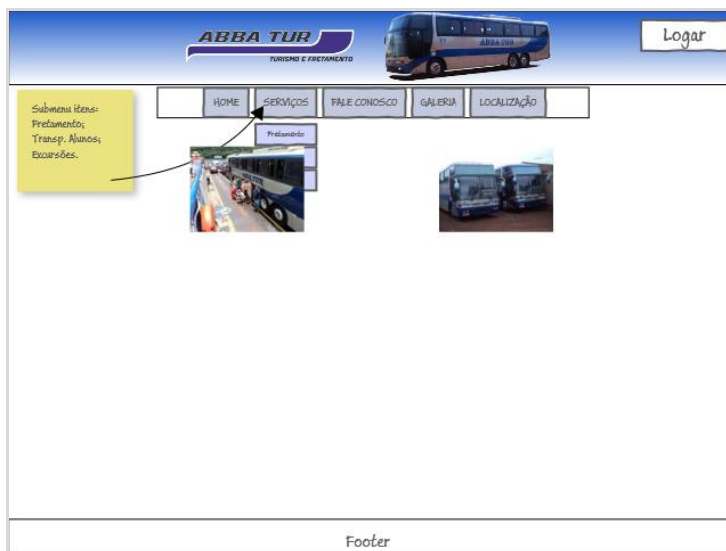


Figura D.6 - Galeria

2.6 Galeria1

A figura D.7: Galeira1 esta apresentando uma imagem de como a imagem e a página se comportará, quando houver algum clique em alguma imagem na página “Galeria” do site.



Figura D.7 - Galeria1

2.7 Localização

A figura D.8: Localização esta apresentando uma imagem da estrutura de como ficará a página de Localização do site.



Figura D.8 - Localização

2.8 Login

A figura D.9: Login esta apresentando uma imagem da tela de login do site.

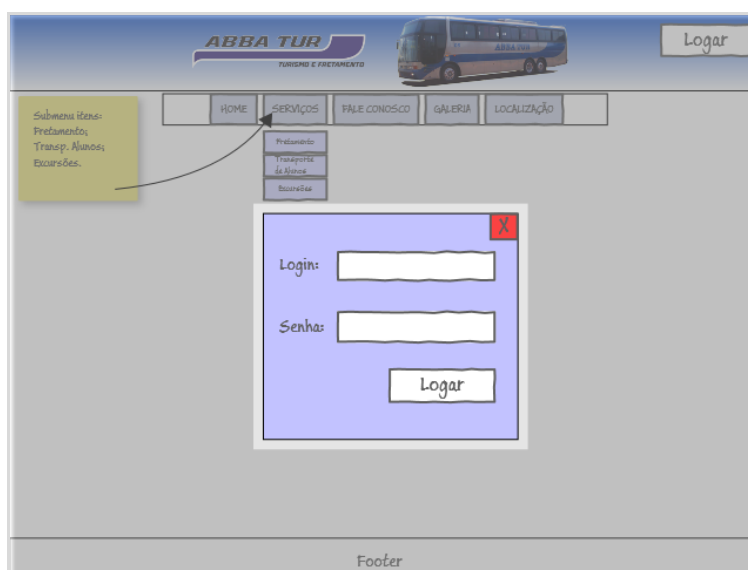


Figura D.9 - Login

2.9 TransporteAlunos

A figura D.10: TransporteAlunos esta apresentando uma imagem da página de Transporte de Alunos do site.

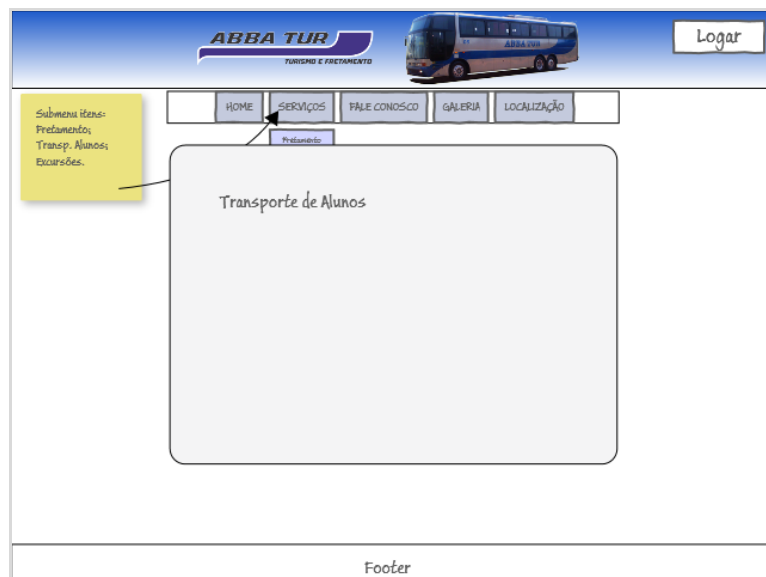


Figura D.10 - TransporteAlunos

Component Screens

2.10 MasterPage

A figura D.11: MasterPage esta apresentando uma imagem da Página Principal do site, onde todas as outras páginas filhas serão carregadas.

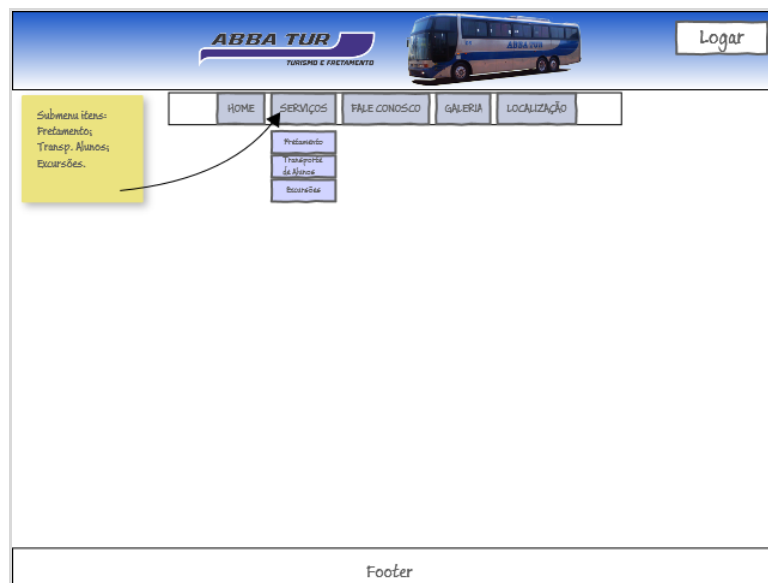


Figura D.11 - MasterPage

APÊNDICE E - Testes do Software

"O teste consiste em executar o programa com a intenção de encontrar erros (*bugs*)". (Myers, 1979).

Este documento apresenta quais e como foram realizados os testes do software (Sistema de Transporte Escolar), focalizando a prevenção de erros e fornecer diagnósticos claros para que os erros sejam facilmente corrigidos.

Os testes foram realizados em dois níveis no desenvolvimento de software, os quais são:

- **Teste Estrutural (Caixa Branca):** Este teste consiste em testar o código, os fluxos, as condições, loops, etc. Foram executadas todas as decisões lógicas para valores falsos ou verdadeiros e todos os laços, confirmando que cada estrutura fosse executada ao menos uma vez.
- **Teste Funcional (Caixa Preta):** Este teste é executado tomando como base os requisitos e as funcionalidades do software, ou seja, são usados para demonstrar que as funções dos softwares são operacionais, que a entrada é adequadamente aceita e a saída é corretamente produzida e a integridade das informações externas é mantida.

Para realizar este teste, os requisitos do sistema foram analisados, e de acordo com eles entradas válidas foram escolhidas para determinar se o produto em teste se comporta corretamente. Entradas inválidas também foram escolhidas para verificar se são detectadas e manipuladas adequadamente. As saídas esperadas para as entradas escolhidas foram determinadas, sendo que as saídas obtidas foram comparadas com as saídas esperadas.

Como os testes foram executados ao longo da produção do projeto, assim permitindo que com os erros encontrados nas fases de testes durante cada iteração, fazer o uso da depuração do sistema, objetivando encontrar a causa do erro detectado no teste, assim, projetando e implementando as modificações no programa para correção do erro antes da implantação do sistema.

APÊNCIDE F – Diagramas da UML

SUMÁRIO

1 DIAGRAMAS E ESPECIFICAÇÕES DOS CASOS DE USO DO MÓDULO DESKTOP	81
1.1 ATORES DO SISTEMA DESKTOP	81
1.2 CASO DE USO: SISTEMA DESKTOP	81
1.3 CASO DE USO: EFETUAR LOGIN	82
1.4 CASO DE USO: ALTERAR SENHA	83
1.5 CASO DE USO: VISUALIZAR DADOS DO CLIENTE	85
1.6 CASO DE USO: GERAR CARTEIRA DE CONTROLE.....	86
2 DIAGRAMAS E ESPECIFICAÇÕES DOS CASOS DE USO DO MÓDULO WEB.....	87
2.1 ATORES DO SISTEMA WEB	87
2.2 CASO DE USO: SISTEMA WEB	87
2.3 CASO DE USO: EFETUAR LOGIN	88
2.4 CASO DE USO: ALTERAR SENHA	89
2.5 CASO DE USO: GERAR CONTRATOS	91
2.6 CASO DE USO: INCLUIR	92
2.7 CASO DE USO: ALTERAR.....	94
2.8 CASO DE USO: PESQUISAR	96
2.9 CASO DE USO: EXCLUIR.....	97
2.10 CASO DE USO: VISUALIZAR CONTRATOS	99
2.11 CASO DE USO: GERAR RELATÓRIOS.....	100
3 DIAGRAMAS DE SEQUÊNCIA DOS MÓDULOS DESKTOP E WEB	102
3.1 SEQUÊNCIA: EFETUAR LOGIN	102
3.2 SEQUÊNCIA: ALTERAR SENHA.....	102
3.3 SEQUÊNCIA: VISUALIZAR DADOS DO CLIENTE	103
3.4 SEQUÊNCIA: GERAR HISTÓRICO	103
3.5 SEQUÊNCIA: GERAR CONTRATOS.....	104
3.6 SEQUÊNCIA: INCLUIR.....	104
3.7 SEQUÊNCIA: ALTERAR	105
3.8 SEQUÊNCIA: PESQUISAR	105
3.9 SEQUÊNCIA: EXCLUIR	106
3.10 SEQUÊNCIA: VISUALIZAR CONTRATOS.....	106
3.11 SEQUÊNCIA: GERAR RELATÓRIOS	107
4 DIAGRAMA DE CLASSE.....	108

1 DIAGRAMAS E ESPECIFICAÇÕES DOS CASOS DE USO DO MÓDULO DESKTOP

1.1 ATORES DO SISTEMA DESKTOP

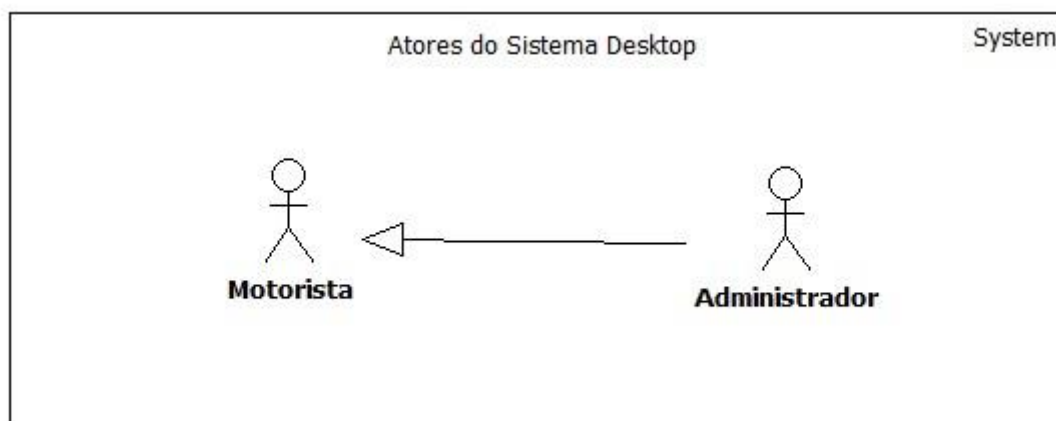


Figura F1.1 – Caso de Uso: Atores do Sistema Desktop

1.2 CASO DE USO: SISTEMA DESKTOP

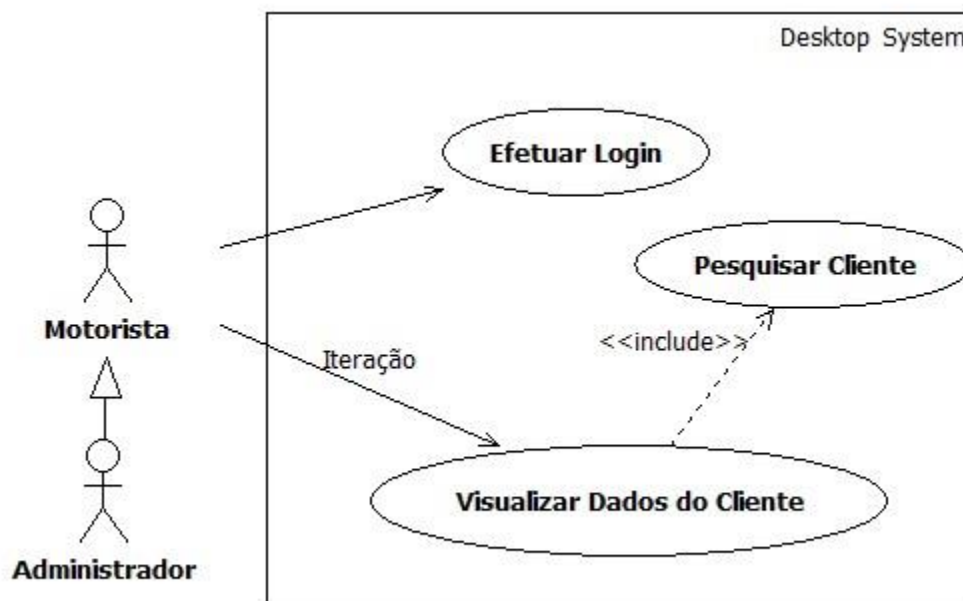


Figura F1.2 – Caso de Uso do Sistema Desktop

1.3 CASO DE USO: EFETUAR LOGIN

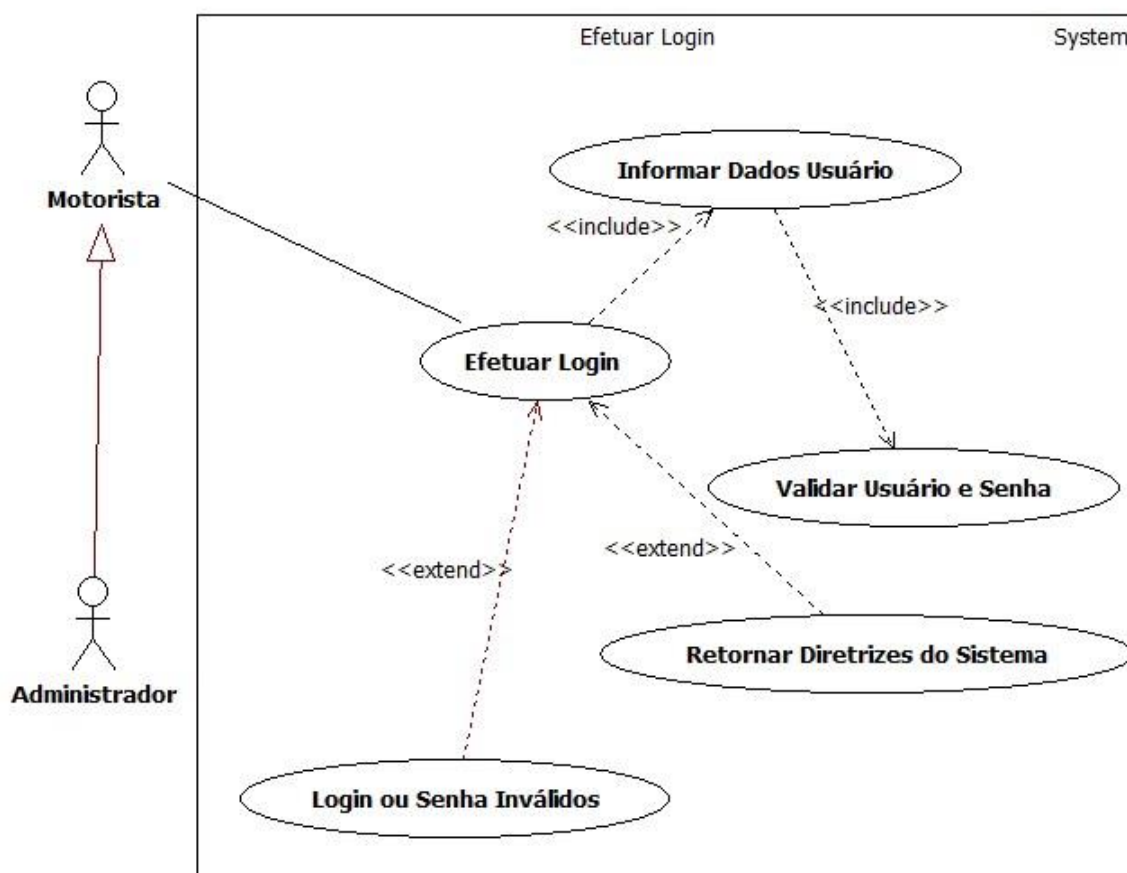


Figura F1.3 – Caso de Uso: Efetuar Login

Caso de Uso	Efetuar Login
Descrição	Este caso de uso é utilizado por administrador/motoristas previamente cadastrados no sistema, visando validar /autenticar as permissões para que os mesmos acessem ao sistema.
Atores	Administrador, Motorista.
Pré-condições	O usuário deve estar cadastrado no sistema.
Fluxo de Eventos	1 - O caso se inicia quando aciona o sistema; 2 - O sistema exibe a tela de login; 3 - O usuário preenche o login e a senha; 4 - Uma vez que o usuário solicite executar a operação (Logar ou Cancelar), um dos fluxos alternativos é executado; [FA01], [FA02] . 5 - O caso de uso está encerrado.
Fluxo Alternativo	FA01: Logar no sistema:

	<p>1 - O usuário acionou a opção “Logar”;</p> <p>2 - O sistema valida os dados (R01);</p> <p>3 - O sistema verifica se o Usuário existe na base de dados (R02);</p> <p>4 - O sistema verifica se a senha esta correta;</p> <p>5 - O sistema realiza o Login do Usuário e libera o acesso ao sistema;</p> <p>FA02: Cancelamento da operação no sistema:</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação;</p>
Restrições	<p>R01 - Campos de usuário e senha deverão ser preenchidos.</p> <p>R02 – Usuário deverá existir na base de dados.</p>
Pós-condições	<p>1 - Login efetuado com sucesso;</p> <p>2 - Usuário não existente;</p> <p>3 - Senha incorreta.</p>

Tabela F1.1 – Especificação do Caso de Uso: Efetuar Login

1.4 CADO DE USO: ALTERAR SENHA

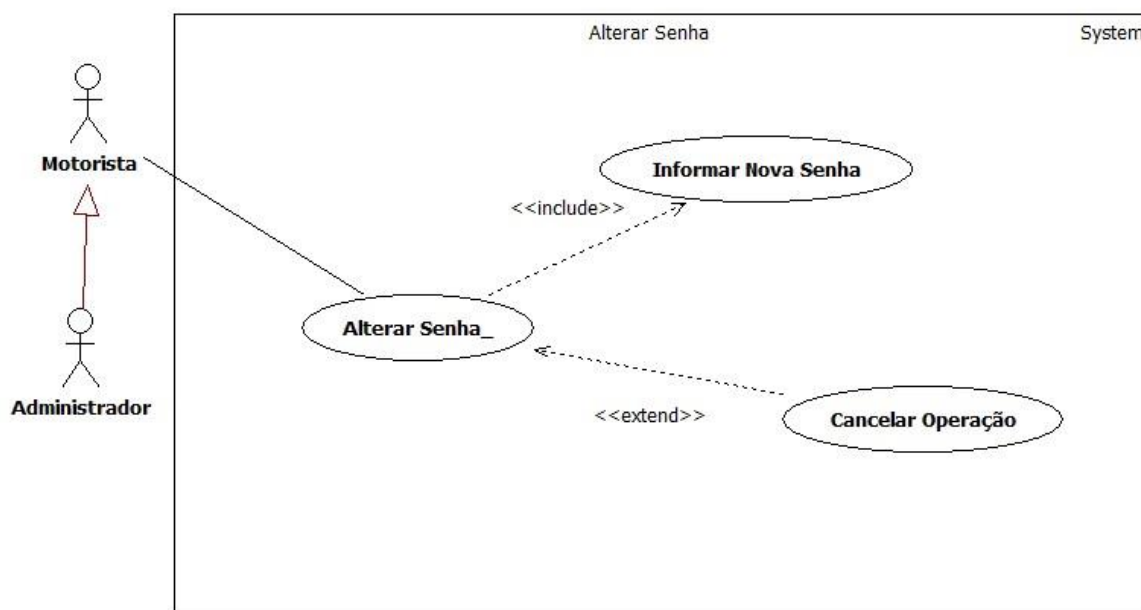


Figura F1.4 – Caso de Uso: Alterar Senha

Caso de Uso	Alterar Senha
Descrição	Este caso de uso é utilizado por administrador/motoristas previamente cadastrados no sistema, visando alterar a senha de acesso no sistema.
Atores	Administrador, Motorista.
Pré-condições	O usuário deve estar logado no sistema.
Fluxo de Eventos	<p>1 - O caso se inicia quando selecionar a opção de “alterar senha de acesso”;</p> <p>2 - O sistema exibe a tela de alteração de senha;</p> <p>3 - O usuário preenche a antiga e a nova senha (R1);</p> <p>4 - Uma vez que o usuário executar uma das opções desejadas (Salvar ou Cancelar), um dos fluxos alternativos é executado; [FA01], [FA02].</p> <p>5 - O caso de uso está encerrado.</p>
Fluxo Alternativo	<p>FA01: Selecionada a opção salvar:</p> <p>1 - O usuário aciona a opção “Salvar”;</p> <p>2 - O sistema valida os dados (R2);</p> <p>3 - O sistema verifica se o Usuário existe na base de dados (R3);</p> <p>4 - O sistema realiza a atualização da Senha.</p> <p>FA02: Cancelamento da operação no sistema:</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação;</p>
Restrições	<p>R1 – Deverá preencher a nova senha duas vezes.</p> <p>R2 – Todos os campos deverão ser preenchidos.</p> <p>R3 – Senha antiga deverá ser validada.</p>
Pós-condições	<p>1 - Senha alterada;</p> <p>2 - Senha antiga incorreta;</p>

Tabela F1.2 – Especificação do Caso de Uso: Alterar Senha

1.5 CASO DE USO: VISUALIZAR DADOS DO CLIENTE

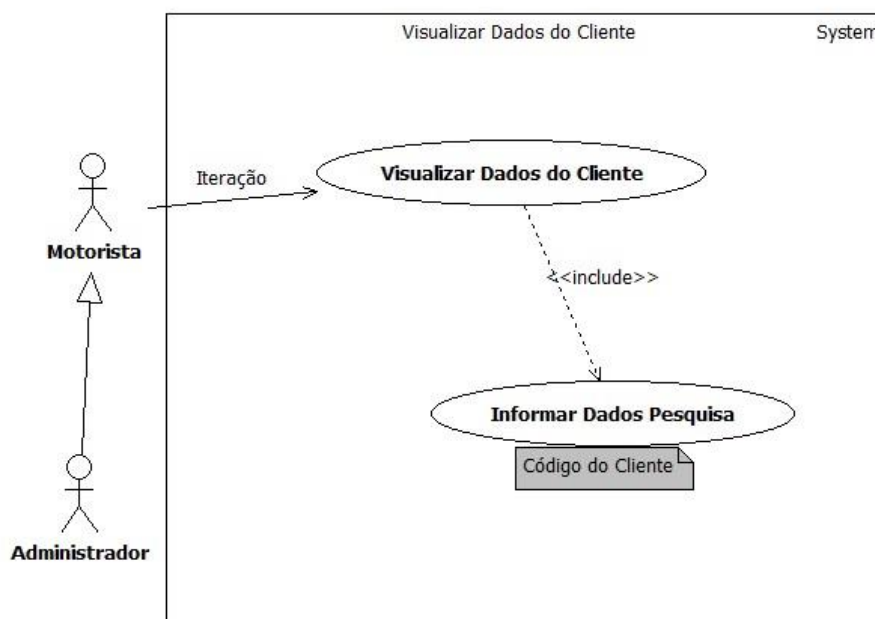


Figura F1.5 – Caso de Uso: Visualizar Dados do Cliente

Caso de Uso	Visualizar Dados do Cliente
Descrição	Este caso de uso é utilizado por administrador/motoristas previamente cadastrados no sistema, visando visualizar os dados do cliente.
Atores	Administrador, Motorista.
Pré-condições	O usuário deve estar logado no sistema.
Fluxo de Eventos	1 - O caso se inicia quando é efetuado o login no sistema; 2 - O sistema exibe a tela principal aguardando a entrada de dados do cliente (código); 3 - O usuário preenche o campo de código do cliente manualmente ou com o uso do leitor de código de barras. 4 - O sistema valida os dados (R1); 5 - O sistema lista e carrega uma grade com as informações do cliente; 6 - O caso de uso está encerrado.
Fluxo Alternativo	Cancelamento da operação no sistema.
Restrições	R1 – Verifica o código.

Tabela F1.3 – Especificação do Caso de Uso: Visualizar Dados do Cliente

1.6 CASO DE USO: GERAR CARTEIRA DE CONTROLE

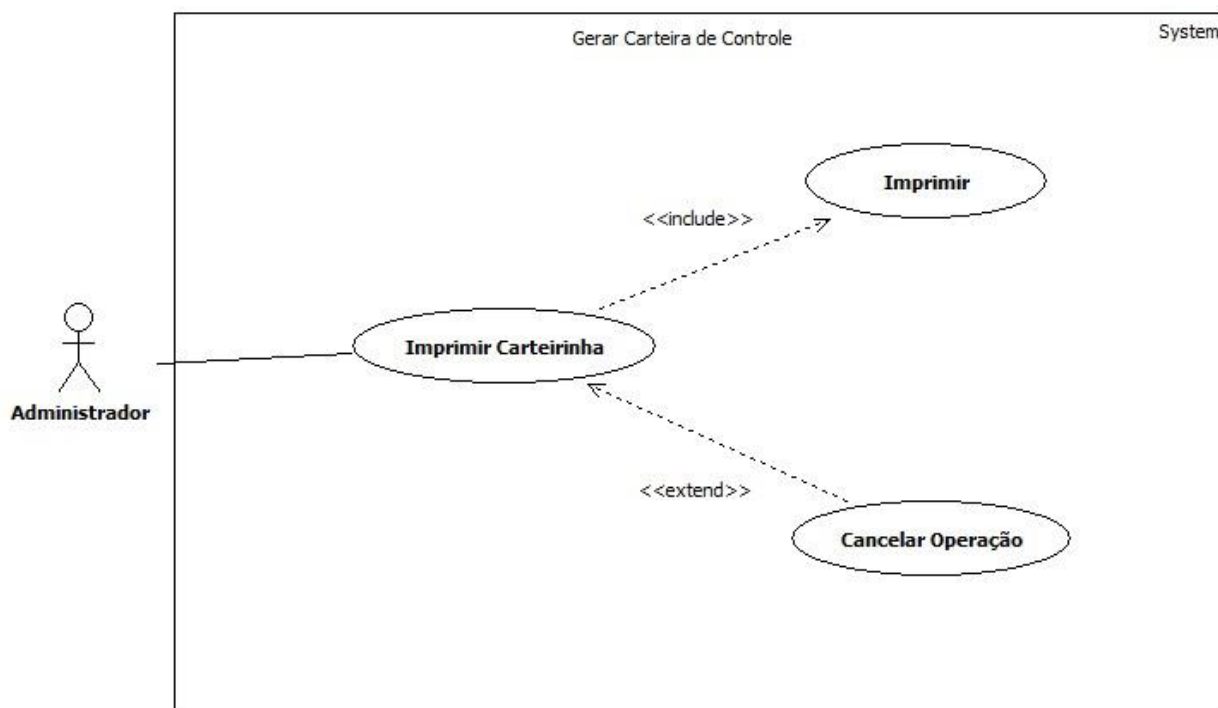


Figura F1.6 – Caso de Uso: Gerar Carteira de Controle

Caso de Uso	Gerar Carteira de Controle
Descrição	Este caso de uso é utilizado pelo Administrador do sistema, visando imprimir a carteira com os dados do cliente.
Atores	Administrador
Pré-condições	1 - O usuário deve estar logado no sistema como perfil de Administrador. 2 – Deverá ter algum cliente selecionado.
Fluxo de Eventos	1 - O caso se inicia quando é clicado em “Gerar Carteira”; 2 - O sistema valida os dados (R1); 3 - O sistema exibe uma tela com a carteira de controle preenchida com os dados do cliente; 4 - O usuário clica em imprimir carteira; 5 - O sistema imprime a carteira de controle; 6 - O caso de uso está encerrado.
Fluxo Alternativo	Cancelamento da operação no sistema.
Restrições	R1 – Verifica se o usuário é Administrador.

Tabela F1.4 – Especificação do Caso de Uso: Gerar Carteira de Controle

2 DIAGRAMAS E ESPECIFICAÇÕES DOS CASOS DE USO DO MÓDULO WEB

2.1 ATORES DO SISTEMA WEB

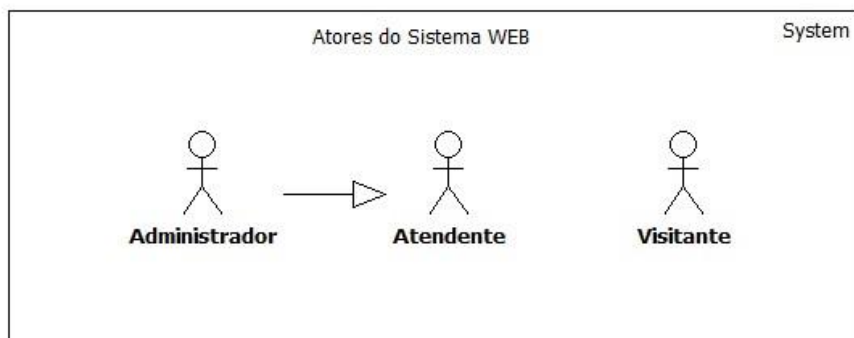


Figura F2.1 – Caso de Uso: Atores do Sistema Web

2.2 CASO DE USO: SISTEMA WEB

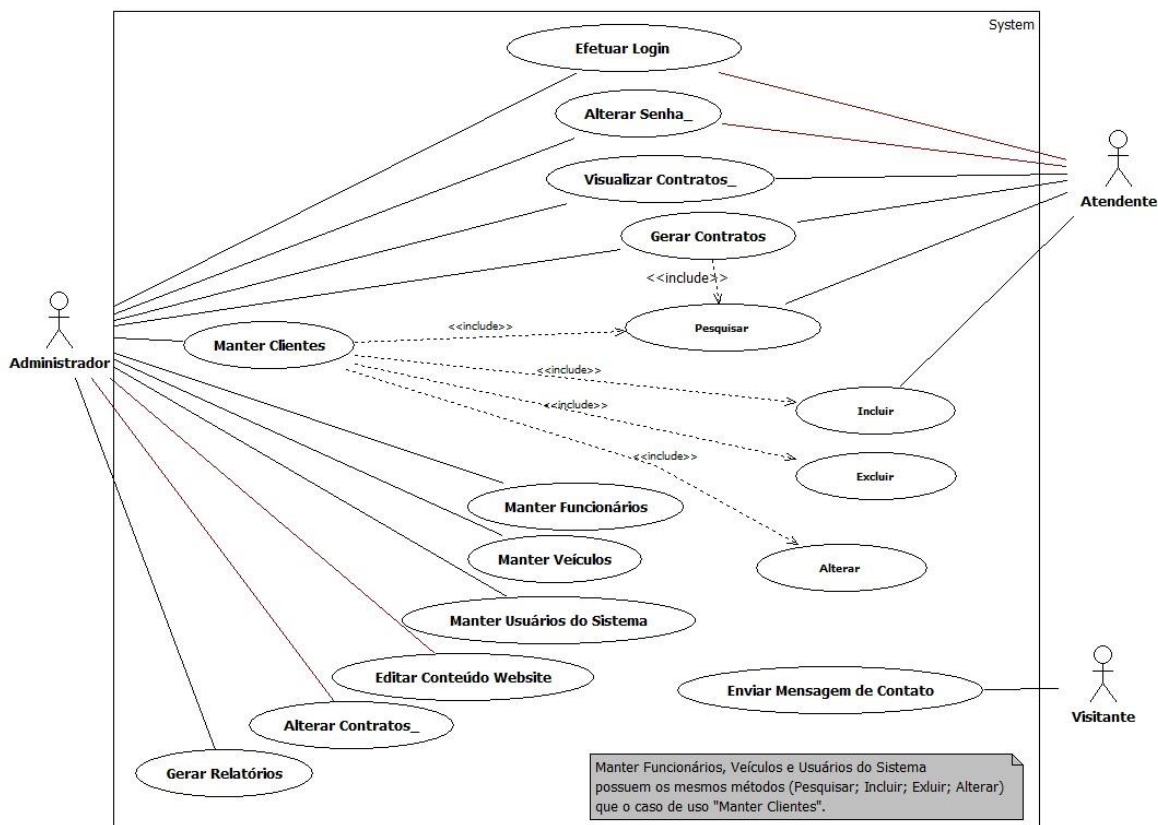


Figura F2.2 - Caso de Uso: Sistema Web

2.3 CASO DE USO: EFETUAR LOGIN

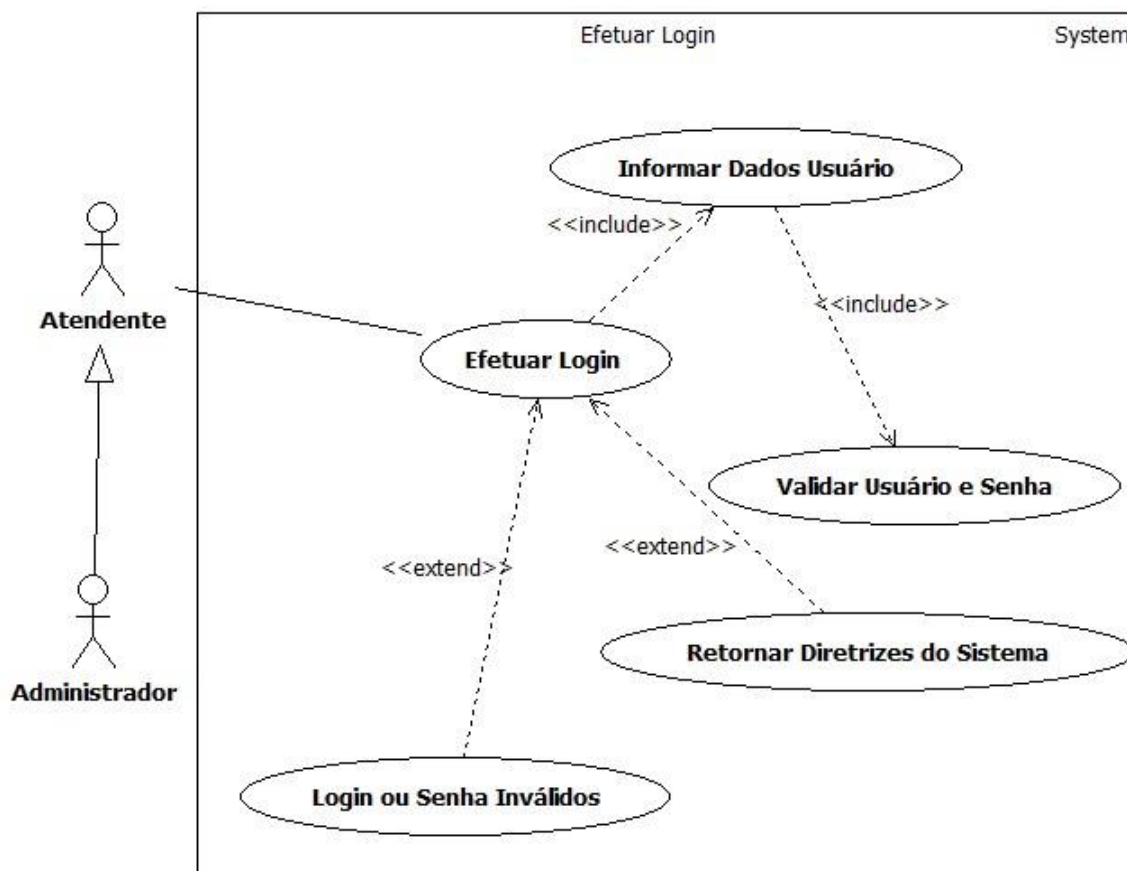


Figura F2.3 - Caso de Uso: Efetuar Login

Caso de Uso	Efetuar Login
Descrição	Este caso de uso é utilizado por administrador/atendente previamente cadastrados no sistema, visando validar /autenticar as permissões para que os mesmos acessem ao sistema.
Atores	Administrador, Atendente.
Pré-condições	O usuário deve estar cadastrado no sistema.
Fluxo de Eventos	1 - O caso se inicia quando se clica em login de usuário na página web; 2 - O sistema exibe a tela de login; 3 - O usuário preenche o login e a senha; 4 - Uma vez que o usuário solicite executar alguma operação (Logar ou Cancelar), um dos fluxos alternativos é executado; [FA01], [FA02] . 5 - O caso de uso está encerrado.

Fluxo Alternativo	<p>FA01 - Logar no sistema:</p> <ol style="list-style-type: none"> 1 - O usuário aciona a opção “Logar”; 2 - O sistema valida os dados (R1); 3 - O sistema verifica se o Usuário existe na base de dados (R2); 4 - O sistema verifica se a senha esta correta; 5 - O sistema realiza o Login do Usuário e libera o acesso ao sistema; <p>FA02 - Cancelamento da operação no sistema:</p> <ol style="list-style-type: none"> 1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação; 2 - O sistema cancela a operação;
Restrições	<p>R1 - Campos de usuário e senha deverão ser preenchidos.</p> <p>R2 – Usuário deverá existir na base de dados.</p>
Pós-condições	<ol style="list-style-type: none"> 1 - Login efetuado com sucesso; 2 - Usuário não existente; 3 - Senha incorreta.

Tabela F2.1 - Especificação do Caso de Uso: Efetuar Login

2.4 CASO DE USO: ALTERAR SENHA

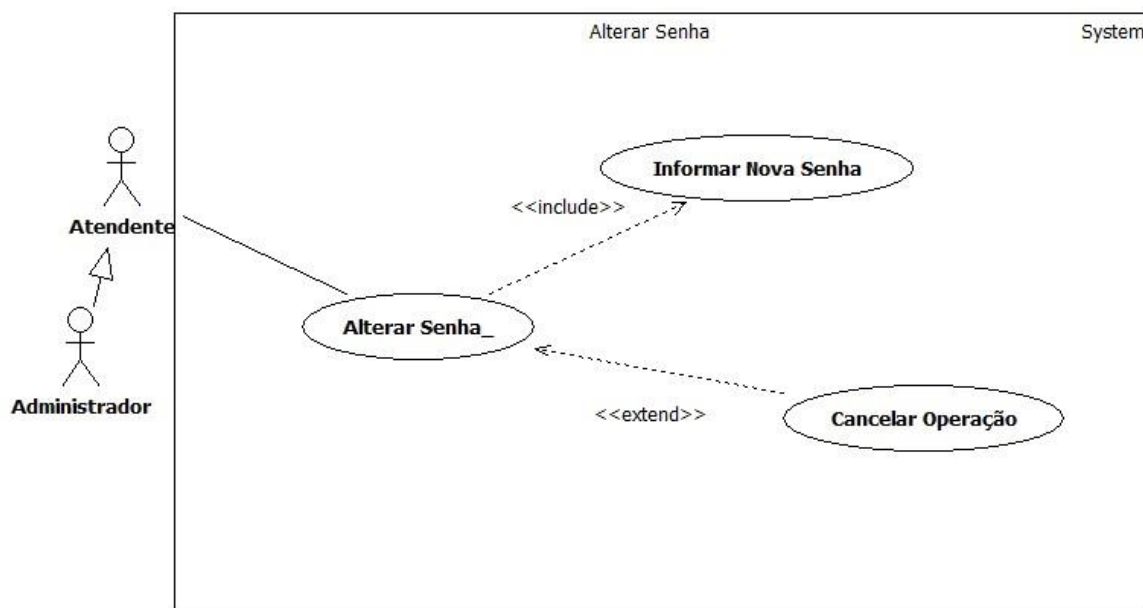


Figura F2.4 - Caso de Uso: Alterar Senha

Caso de Uso	Alterar Senha
Descrição	Este caso de uso é utilizado por administrador/atendente previamente cadastrados no sistema, visando alterar a senha de acesso no sistema.
Atores	Administrador, Atendente.
Pré-condições	O usuário deve estar logado no sistema.
Fluxo de Eventos	<p>1 - O caso se inicia quando selecionar a opção de alterar senha de acesso;</p> <p>2 - O sistema exibe a tela de alteração de senha;</p> <p>3 - O usuário preenche a antiga e a nova senha (R1);</p> <p>4 - Uma vez que o usuário executar uma das opções desejadas (Salvar Nova Senha ou Cancelar), um dos fluxos alternativos é executado; [FA01], [FA02].</p> <p>5 - O caso de uso está encerrado.</p>
Fluxo Alternativo	<p>FA01: Salvar nova senha:</p> <p>1 - O usuário aciona a opção “Salvar”;</p> <p>2 - O sistema valida os dados (R2);</p> <p>3 - O sistema verifica se o Usuário existe na base de dados (R3);</p> <p>4 - O sistema realiza a atualização da Senha.</p> <p>FA02: Cancelamento da operação no sistema:</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação.</p>
Restrições	<p>R1 – Deverá preencher a nova senha duas vezes.</p> <p>R2 – Todos os campos deverão ser preenchidos.</p> <p>R3 – Senha antiga deverá ser validada.</p>
Pós-condições	<p>1 - Senha alterada;</p> <p>2 - Senha antiga incorreta.</p>

Tabela F2.2 - Especificação do Caso de Uso: Alterar Senha

2.5 CASO DE USO: GERAR CONTRATOS

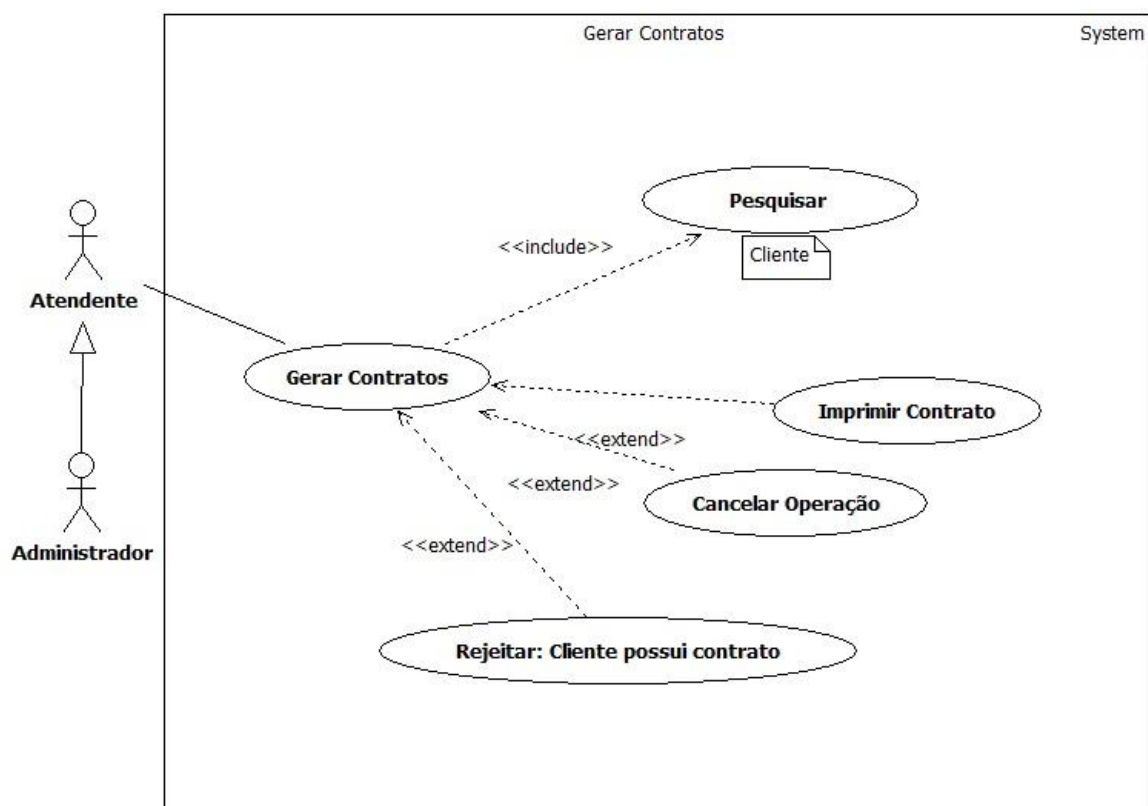


Figura F2.5 - Caso de Uso: Gerar Contratos

Caso de Uso	Gerar Contratos
Descrição	Este caso de uso é utilizado por administrador/atendente previamente cadastrados no sistema, visando gerar novos contratos.
Atores	Administrador, Atendente.
Pré-condições	O usuário deve estar logado no sistema. O cliente deve estar cadastrado no sistema.
Fluxo de Eventos	1 - O caso se inicia quando selecionar a opção “Gerar novos contratos”; 2 - O sistema exibe a tela de busca de cliente para seleciona-lo e incluí-lo no novo contrato (R1); 3 - O usuário preenche os dados do contrato; 4 - Uma vez que o usuário executar uma das opções desejadas (Gerar Novo Contrato ou Cancelar), um dos fluxos alternativos é executado; [FA01] , [FA02] .

	5 - O caso de uso está encerrado.
Fluxo Alternativo	<p>FA01 - Gerar novo contrato:</p> <p>1 - O usuário aciona a opção “Gerar Contrato”;</p> <p>2 - O sistema valida os dados (R2, R3);</p> <p>3 - O sistema grava as informações na base de dados e gera uma página pra impressão do contrato criado.</p> <p>FA02 - Cancelamento da operação no sistema;</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação;</p>
Restrições	<p>R1 - Verifica se cliente já possui contrato.</p> <p>R2 - Todos os campos deverão ser preenchidos.</p> <p>R3 - Verifica as datas do contrato, data de início deverá ser menor que data de término.</p>
Pós-condições	1 - Contrato criado e salvo;

Tabela F2.3 - Especificação do Caso de Uso: Gerar Contratos

2.6 CASO DE USO: INCLUIR

O seguintes casos de uso: Incluir, Alterar, Pesquisar e Excluir, servirá como base para os casos de uso: Manter Clientes; Manter Veículos; Manter Usuários do Sistema; Manter Funcionários.

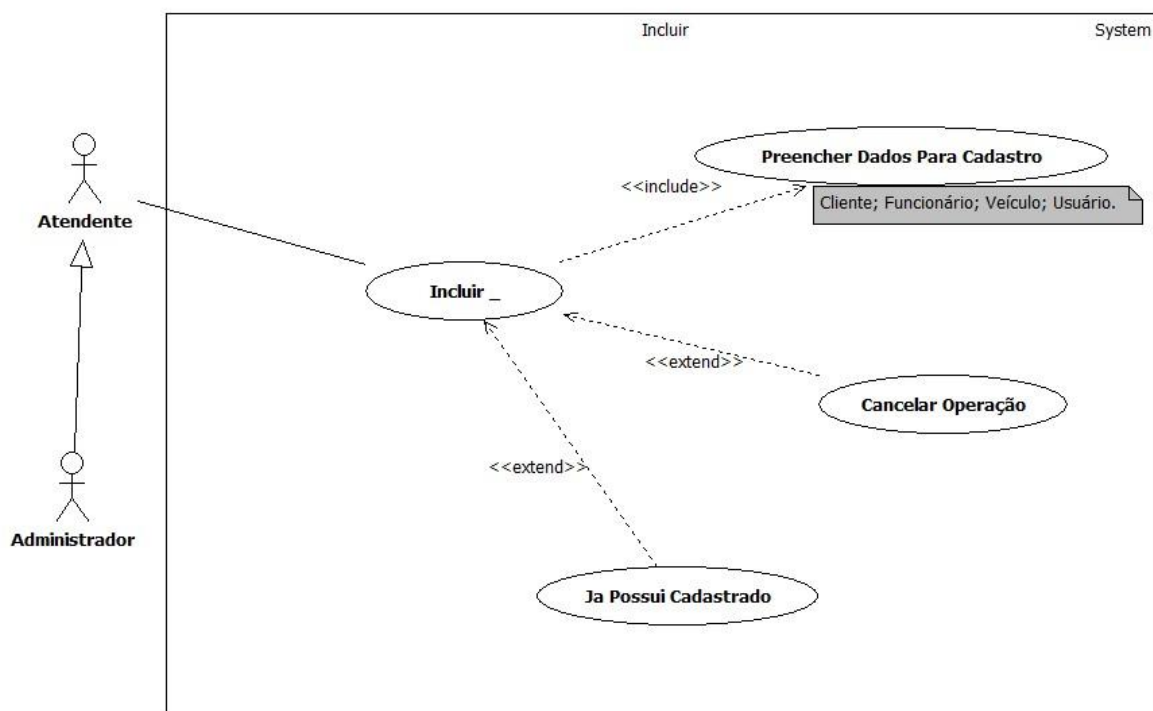


Figura F2.6 - Caso de Uso: Incluir

Caso de Uso	Incluir
Descrição	<p>Este caso de uso é utilizado para incluir novos clientes, funcionários, veículos ou usuários do sistema.</p> <p>Ele poderá ser utilizado pelo usuário Atendente somente no caso de incluir clientes, sendo que os demais casos somente o usuário Administrador terá acesso.</p>
Atores	Administrador, Atendente.
Pré-condições	<p>O usuário deve estar logado no sistema.</p> <p>O cliente, funcionários, veículo ou usuários do sistema não devem possuir cadastro no sistema.</p> <p>Para cadastro de funcionário, veículo e usuário do sistema, o usuário deverá estar logado como Administrador.</p>
Fluxo de Eventos	<p>1 - O caso se inicia quando selecionar a opção “Incluir Novo”;</p> <p>2 - O sistema exibirá uma tela de seleção, no qual o usuário poderá selecionar entre cliente, funcionários, veículos ou usuário do sistema de acordo com nível de acesso do usuário que estará logado;</p> <p>3 - O sistema exibe a tela de cadastro de acordo com a opção selecionada;</p> <p>4 - O usuário preenche os dados;</p> <p>5 - O sistema verifica os dados (R1);</p> <p>6 - Uma vez que o usuário executar uma das opções desejadas (Salvar ou Cancelar), um dos fluxos alternativos é</p>

	executado; [FA01] , [FA02] . 7 - O caso de uso está encerrado.
Fluxo Alternativo	FA01: Incluir ou Salvar: 1 - O usuário aciona a opção "Salvar"; 2 - O sistema valida os dados (R2); 3 - O sistema grava as informações na base de dados. FA02: Cancelamento da operação no sistema: 1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação; 2 - O sistema cancela a operação;
Restrições	R1 - Todos os campos deverão ser preenchidos. R2 - Verifica se o que está sendo incluído já possui cadastro.

Tabela F2.4 - Especificação do Caso de Uso: Incluir

2.7 CASO DE USO: ALTERAR

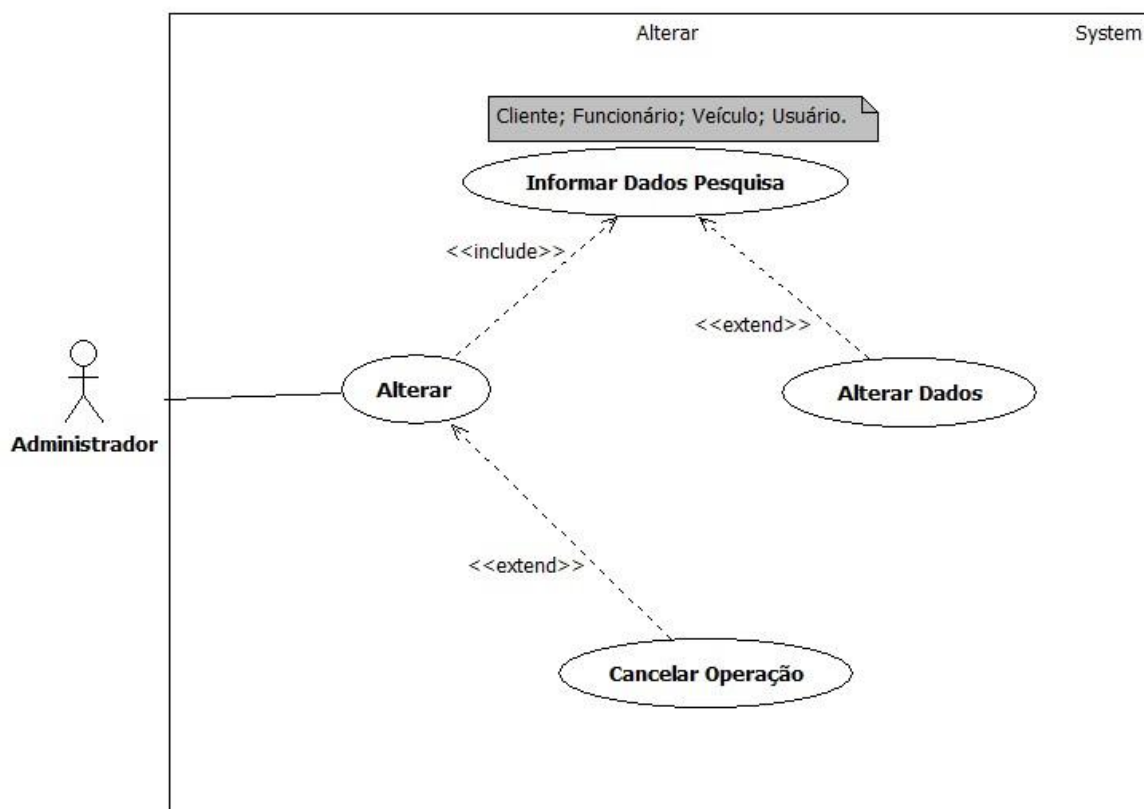


Figura F2.7 - Caso de Uso: Incluir

Caso de Uso	Alterar
Descrição	Este caso de uso é utilizado pelo Administrador visando alterar os dados de clientes, funcionários, veículos ou usuários cadastrados.
Atores	Administrador.
Pré-condições	O usuário deve estar logado no sistema com perfil de Administrador.
Fluxo de Eventos	<p>1 - O caso se inicia quando selecionar a opção “Alterar”;</p> <p>2 - O sistema exibirá uma a tela de alteração com a opção de selecionar o que será alterado, como: cliente, funcionários, veículos ou usuário do sistema.</p> <p>3 - O sistema exibe um campo de texto e uma caixa de seleção com a finalidade de filtrar a busca do que vai ser alterado;</p> <p>4 - O usuário preenche os dados do filtro e clica em “Buscar”;</p> <p>5 - O sistema exibe a tela de alteração com os dados preenchidos de acordo com a opção e filtros selecionados;</p> <p>5 - O usuário altera os dados;</p> <p>6 - O sistema verifica os dados (R1);</p> <p>7 - Uma vez que o usuário executar uma das opções desejadas (Alterar ou Cancelar), um dos fluxos alternativos é executado; [FA01], [FA02].</p> <p>8 - O caso de uso está encerrado.</p>
Fluxo Alternativo	<p>FA01 - Selecionada o opção “Alterar”:</p> <p>1 - O usuário aciona a opção “Alterar”;</p> <p>2 - O sistema valida os dados (R2);</p> <p>3 - O sistema grava as alterações na base de dados.</p> <p>FA02: Cancelamento da operação no sistema:</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação;</p>
Restrições	<p>R1 - Todos os campos deverão ser preenchidos.</p> <p>R2 - Verifica os dados que estão sendo alterados.</p>
Pós-condições	1 - Alteração realizada.

Tabela F2.5 - Especificação do Caso de Uso: Alterar

2.8 CASO DE USO: PESQUISAR

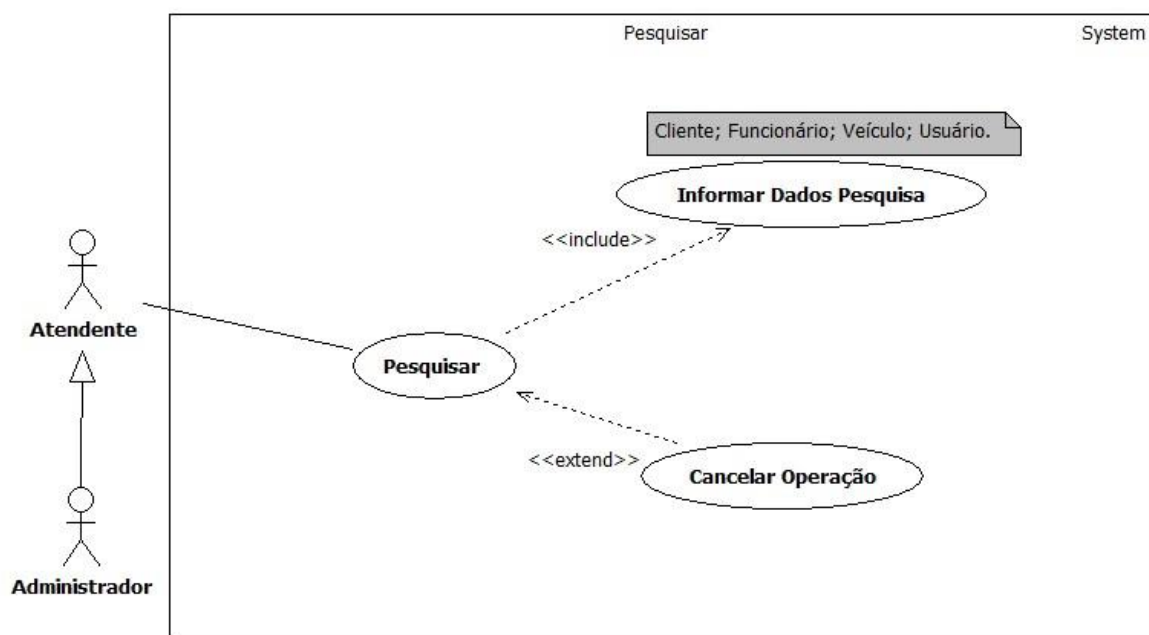


Figura F2.8 - Caso de Uso: Pesquisar

Caso de Uso	Pesquisar
Descrição	Este caso de uso é utilizado para pesquisar clientes, funcionários, veículos ou usuários cadastrados. Ele poderá ser utilizado pelo usuário Atendente somente no caso de pesquisar clientes, sendo que os demais casos somente o usuário Administrador terá acesso.
Atores	Administrador, Atendente.
Pré-condições	O usuário deve estar logado no sistema. O cliente, funcionários, veículo ou usuários deverão estar cadastrados. Para pesquisa de funcionário, veículo e usuário do sistema, o usuário deverá estar logado com perfil de Administrador.
Fluxo de Eventos	1 - O caso se inicia quando selecionar a opção "Pesquisar"; 2 - O sistema exibirá uma tela de seleção, no qual o usuário poderá selecionar entre cliente, funcionários, veículos ou usuário do sistema de acordo com nível de acesso do usuário que estará logado; 3 - O sistema exibe a tela de pesquisa de acordo com a opção selecionada;

	<p>4 - O usuário preenche os dados para pesquisa;</p> <p>5 - O sistema verifica os dados (R1);</p> <p>6 - Uma vez que o usuário executar uma das opções desejadas (Pesquisar ou Cancelar), um dos fluxos alternativos é executado; [FA01], [FA02].</p> <p>7 - O caso de uso está encerrado.</p>
Fluxo Alternativo	<p>FA01: Selecionada a opção pesquisar:</p> <p>6 - O usuário aciona a opção “Pesquisar”;</p> <p>7 - O sistema valida os dados (R2);</p> <p>8 - O sistema preenche os campos com as informações da base de dados.</p> <p>FA02: Cancelamento da operação no sistema:</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação;</p>
Restrições	<p>R1 - Todos os campos deverão ser preenchidos.</p> <p>R2 - Verifica se o que está sendo pesquisado possui cadastro.</p>

Tabela F2.6 - Especificação do Caso de Uso: Pesquisar

2.9 CASO DE USO: EXCLUIR

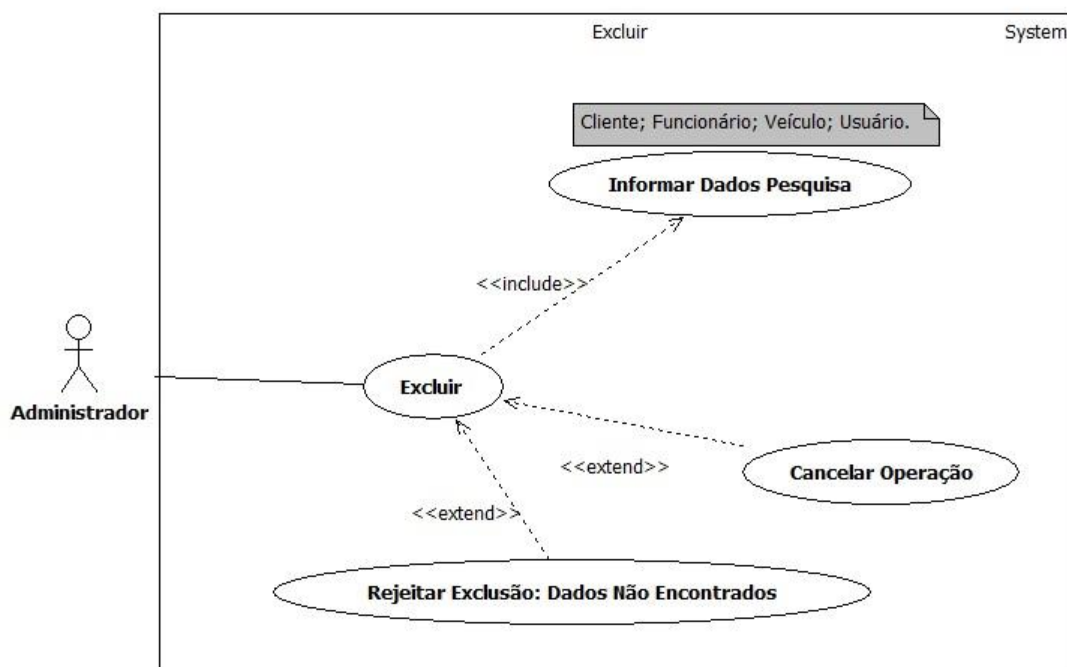


Figura F2.9 - Caso de Uso: Pesquisar

Caso de Uso	Excluir
Descrição	Este caso de uso é utilizado pelo Administrador visando excluir clientes, funcionários, veículos ou usuários cadastrados.
Atores	Administrador.
Pré-condições	O usuário deve estar logado no sistema com perfil de Administrador.
Fluxo de Eventos	<p>1 - O caso se inicia quando selecionar a opção “Excluir”;</p> <p>2 - O sistema exibirá uma tela de seleção, no qual o usuário poderá selecionar entre cliente, funcionários, veículos ou usuário do sistema.</p> <p>3 - O sistema exibe a tela de pesquisa de acordo com a opção selecionada;</p> <p>4 - Usuário preenche dados para pesquisa;</p> <p>5 - O usuário visualiza dados carregados e seleciona o que vai ser excluído;</p> <p>6 - Uma vez que o usuário executar uma das opções desejadas (Excluir ou Cancelar), um dos fluxos alternativos é executado; [FA01], [FA02].</p> <p>7 - O caso de uso está encerrado.</p>
Fluxo Alternativo	<p>FA01: Selecionada a opção excluir:</p> <p>1 - O usuário aciona a opção “Excluir”;</p> <p>2 - O sistema valida os dados (R1);</p> <p>3 - O sistema exibe mensagem de confirmação;</p> <p>4 - O usuário confirma ou interrompe exclusão;</p> <p>FA02: Cancelamento ou interrupção da operação no sistema:</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação;</p>
Restrições	R1 - Verifica se há algum item marcado para exclusão.
Pós-condições	1 - Exclusão efetuada;

Tabela F2.7 - Especificação do Caso de Uso: Excluir

2.10 CASO DE USO: VISUALIZAR CONTRATOS

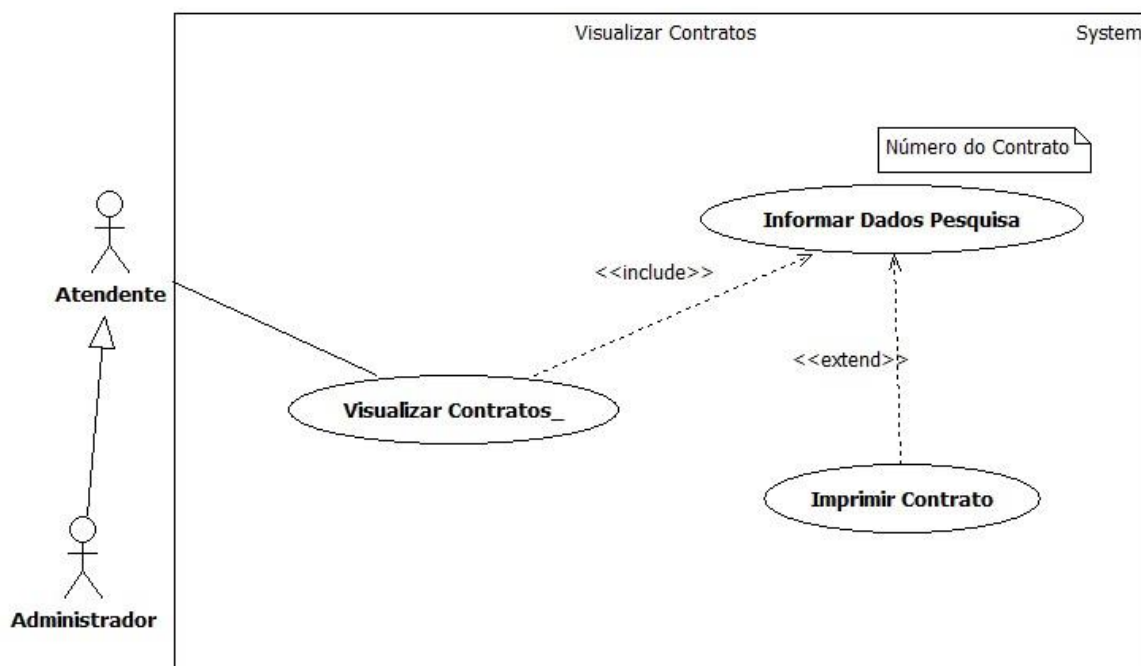


Figura F2.10 - Caso de Uso: Visualizar Contratos

Caso de Uso	Visualizar Contratos
Descrição	Este caso de uso é utilizado pelo Administrador ou Atendente no intuito de visualizar contratos de acordo com filtro selecionado.
Atores	Administrador, Atendente.
Pré-condições	O usuário deve estar logado no sistema.
Fluxo de Eventos	<p>1 - O caso se inicia quando selecionar a opção “Visualizar Contratos”;</p> <p>2 - O sistema exibirá a tela com opções de filtro para pesquisa, como: todos os contratos, contratos finalizados, contratos em vigor, cliente, etc.</p> <p>3 - O usuário clica em visualizar;</p> <p>4 - O sistema valida os dados (R1);</p> <p>5 - O sistema lista os contratos com alguns dados e apresenta algumas opções (Mostrar Mais, Imprimir ou Cancelar); Ao executar uma das opções um dos fluxos alternativos serão executados; [FA01] [FA02] [FA03].</p> <p>6 - O caso de uso está encerrado.</p>
Fluxo Alternativo	FA01: Opção “mostrar mais”:

	<p>1 - O usuário clica na opção “mostrar mais”, e será exibido o contrato completo.</p> <p>FA02: Opção imprimir:</p> <p>1 - O sistema proporcionará ao usuário a opção de imprimir o contrato. (R2).</p> <p>FA02: Cancelamento da operação no sistema:</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação;</p>
Restrições	<p>R1 - Algum filtro deverá ser selecionado ou preenchido.</p> <p>R2 - A opção de imprimir só estará disponível após executar a opção “mostrar mais”.</p>

Tabela F2.8 - Especificação do Caso de Uso: Visualizar Contratos

2.11 CASO DE USO: GERAR RELATÓRIOS

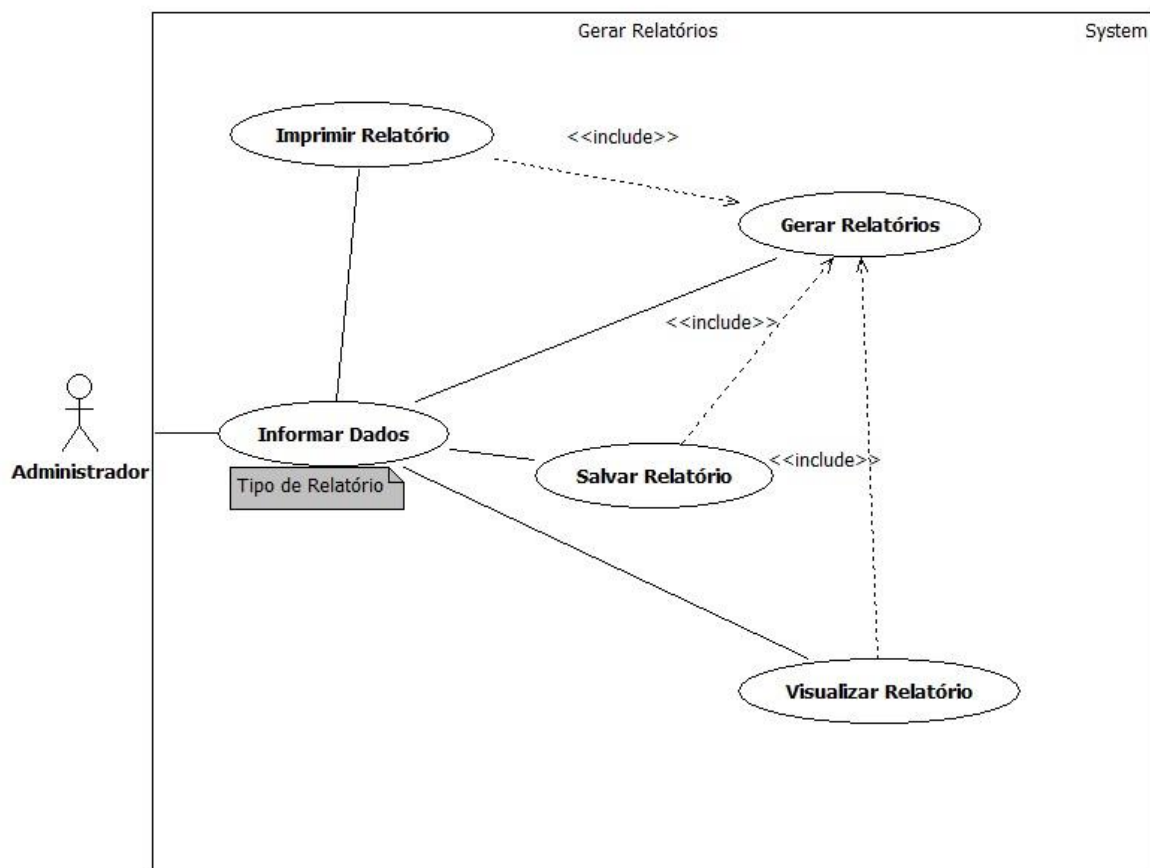


Figura F2.11 - Caso de Uso: Visualizar Contratos

Caso de Uso	Gerar Relatórios
Descrição	Este caso de uso permitirá emitir relatório de acordo com o filtro informado.
Atores	Administrador.
Pré-condições	O usuário deve estar logado no sistema com perfil de Administrador.
Fluxo de Eventos	<p>1 - O caso se inicia quando ator requisita emitir relatório;</p> <p>2 - O sistema exibirá uma tela de seleção, no qual o usuário poderá selecionar qual tipo de relatório;</p> <p>3 - O usuário seleciona um tipo;</p> <p>4 - Uma vez que o usuário executar uma das opções desejadas (Emitir Relatório ou Cancelar), um dos fluxos seguintes alternativos é executado; [FA01], [FA02].</p> <p>5 - O caso de uso está encerrado.</p>
Fluxo Alternativo	<p>FA01:</p> <p>1 - Este fluxo alternativo se inicia quando o ator solicita emitir relatórios;</p> <p>2 - O sistema valida os dados informados (R01);</p> <p>3 - O sistema exibe o relatório ordenado crescentemente com as opções de imprimir;</p> <p>FA02 - Cancelamento da operação no sistema:</p> <p>1 - Este fluxo alternativo se inicia quando o usuário solicita cancelar a operação;</p> <p>2 - O sistema cancela a operação;</p>
Restrições	<p>R01:</p> <p>1 - Nenhum resultado para o filtro informado. (Se a consulta não retornar nenhum resultado, o sistema exibe a mensagem: "Nenhum dado encontrado para o filtro de pesquisa informado.");</p> <p>2 - O caso de uso retorna ao passo 2 do fluxo de eventos.</p>
Pós-condições	Relatório Emitido. Relatório impresso.

Tabela F2.9 - Especificação do Caso de Uso: Gerar Relatórios

3 DIAGRAMAS DE SEQUÊNCIA DOS MÓDULOS DESKTOP E WEB

3.1 SEQUÊNCIA: EFETUAR LOGIN

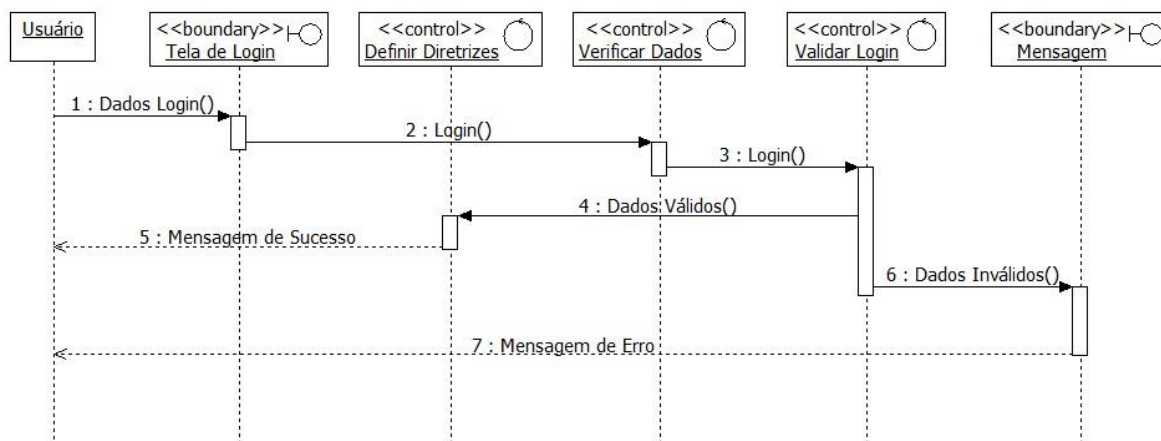


Figura F3.1 - Sequência: Efetuar Login

3.2 SEQUÊNCIA: ALTERAR SENHA

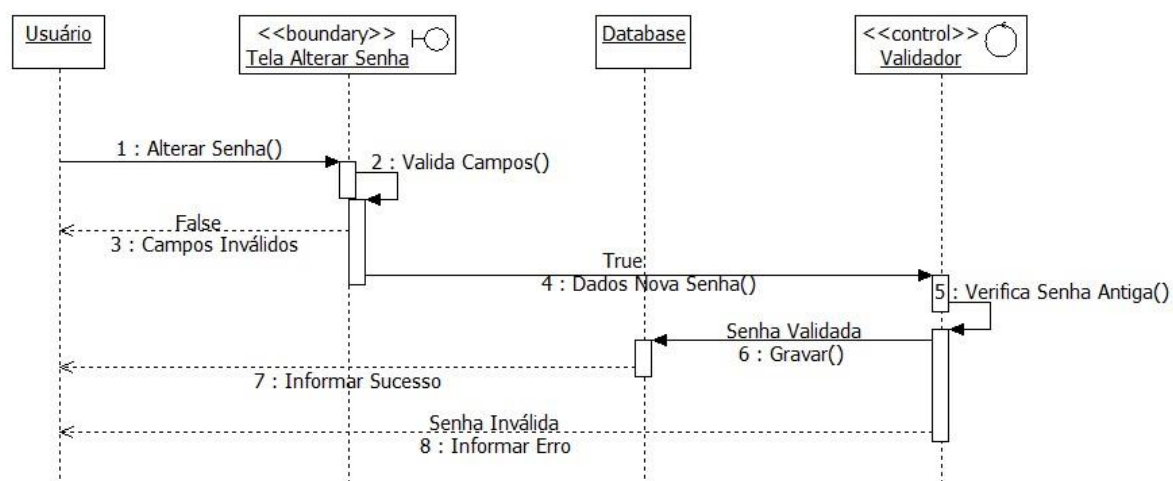


Figura F3.2 - Sequência: Alterar Senha

3.3 SEQUÊNCIA: VISUALIZAR DADOS DO CLIENTE

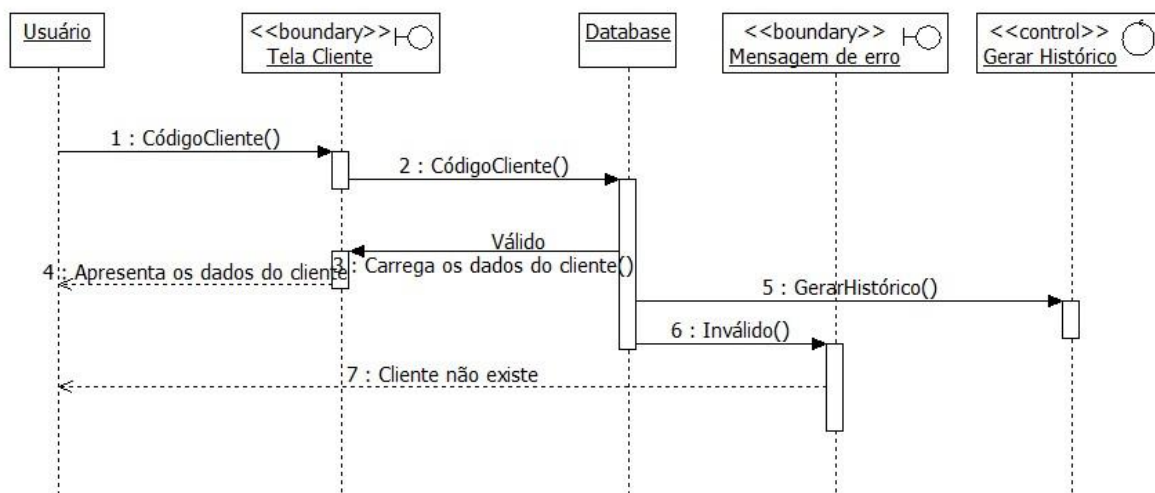


Figura F3.3 - Sequência: Visualizar Dados do Cliente

3.4 SEQUÊNCIA: GERAR HISTÓRICO

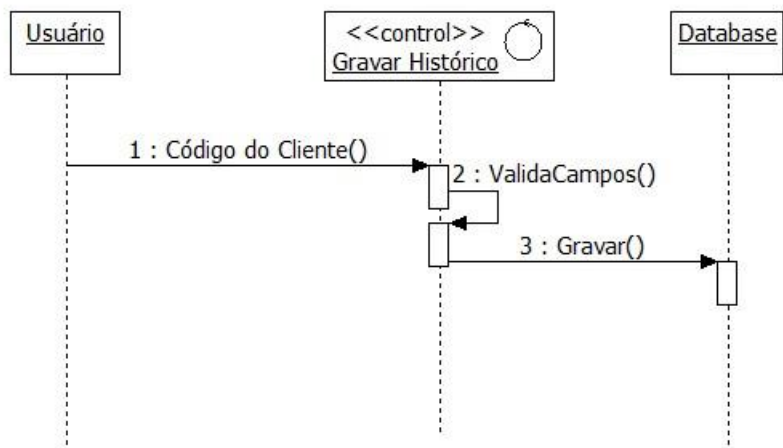


Figura F3.4 - Sequência: Gerar Histórico

3.5 SEQUÊNCIA: GERAR CONTRATOS

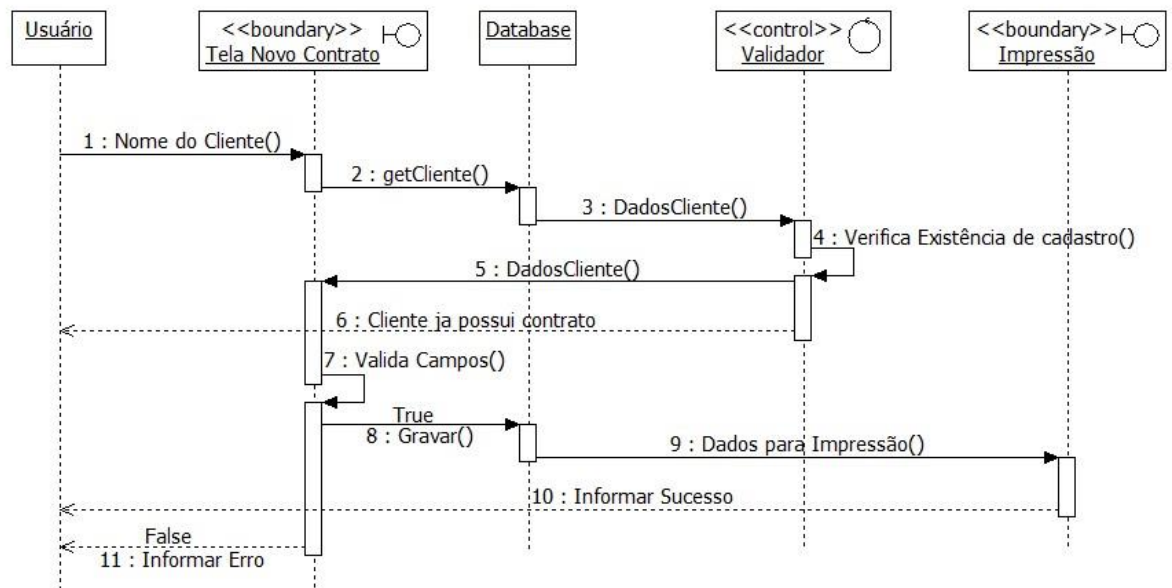


Figura F3.5 - Sequência: Gerar Contratos

3.6 SEQUÊNCIA: INCLUIR

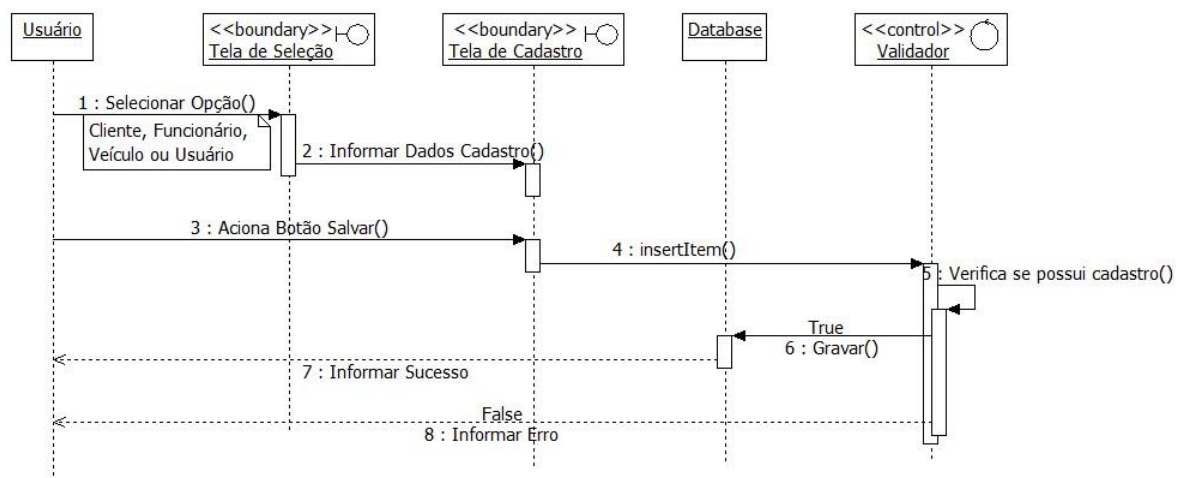


Figura F3.6 - Sequência: Incluir

3.7 SEQUÊNCIA: ALTERAR

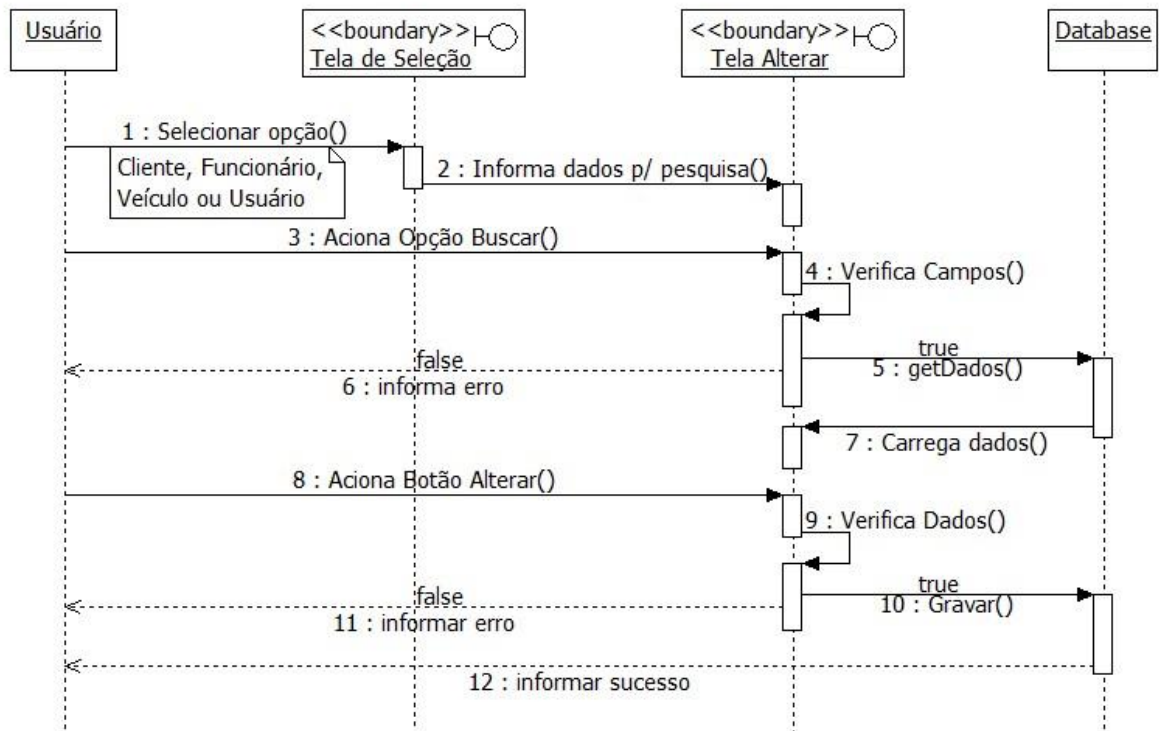


Figura F3.7 - Sequência: Alterar

3.8 SEQUÊNCIA: PESQUISAR

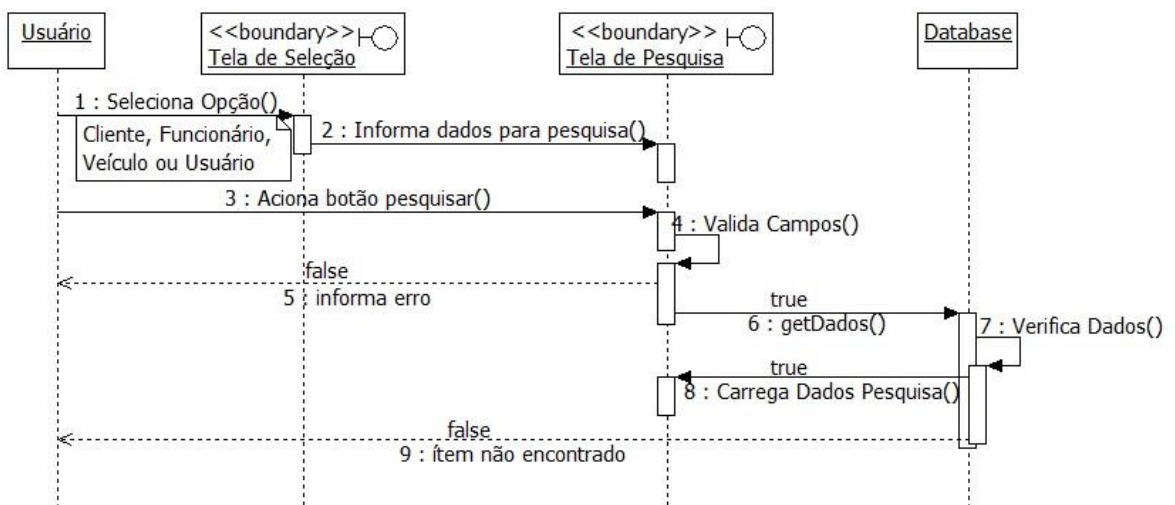


Figura F3.8 - Sequência: Pesquisar

3.9 SEQUÊNCIA: EXCLUIR

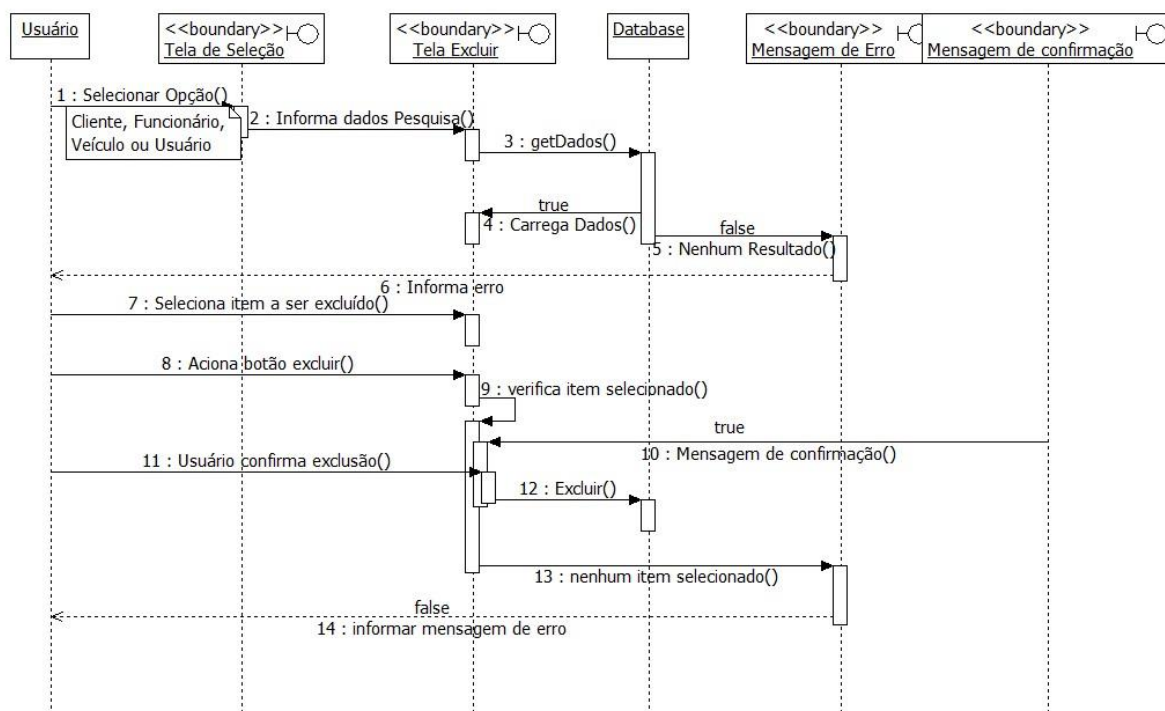


Figura F3.9 - Sequência: Excluir

3.10 SEQUÊNCIA: VISUALIZAR CONTRATOS

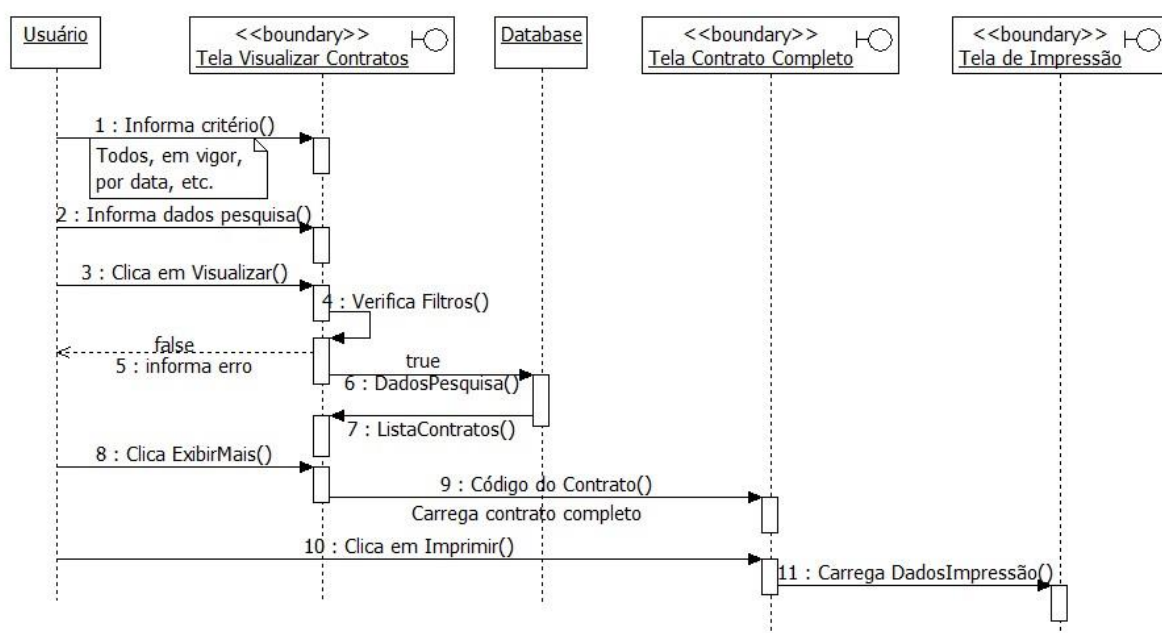


Figura F3.10 - Sequência: Visualizar Contratos

3.11 SEQUÊNCIA: GERAR RELATÓRIOS

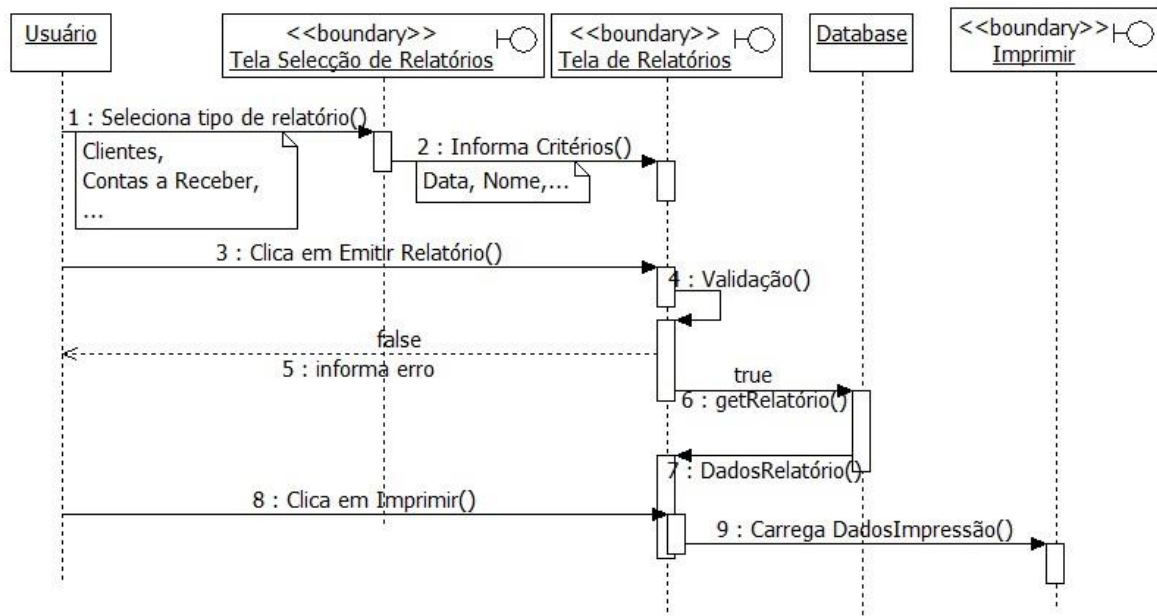


Figura F3.11 - Sequência: Gerar Relatórios

4 DIAGRAMA DE CLASSE

Como as demais classes do sistema se comportam da mesma maneira, foi criado um Diagrama de Classes apresentado a seguir, que exhibe como as classes se comportam no sistema.

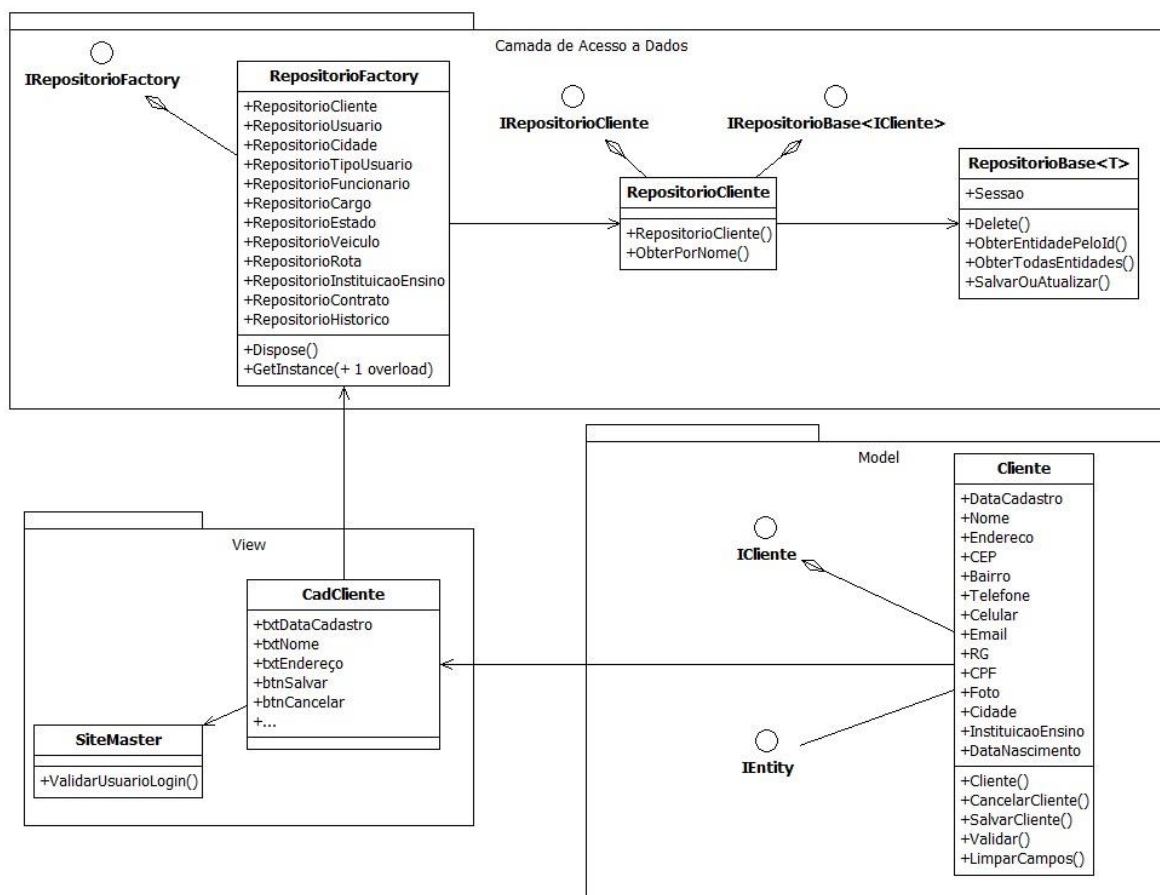


Figura F4 - Diagrama de Classe

**APÊNDICE G – DIAGRAMAS MODELO ENTIDADE-
RELACIONAMENTO**

Na Figura G.1 é mostrado o diagrama modelo entidade-relacionamento do banco de dados do sistema *desktop*.

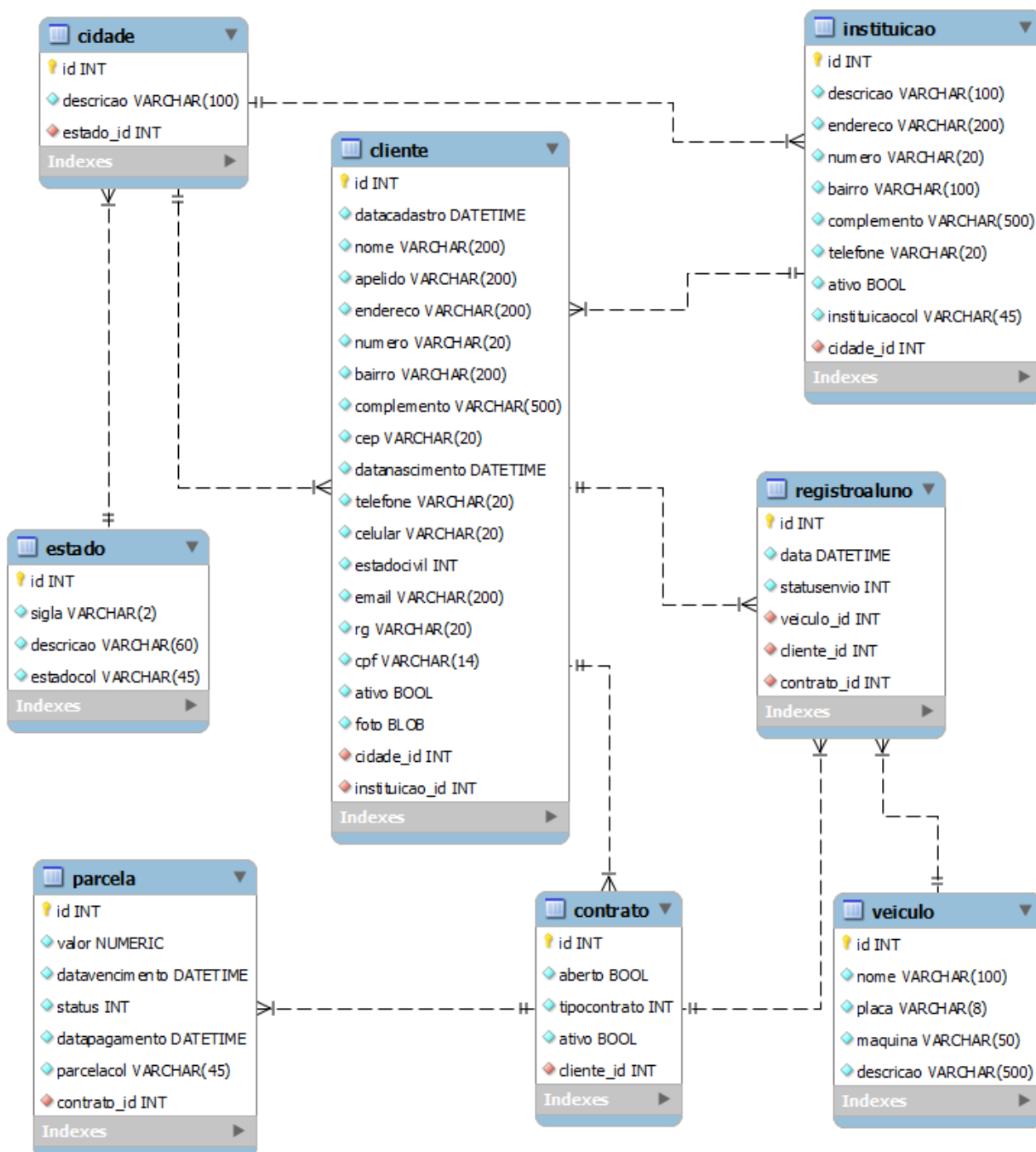


Figura G.1 – MER sistema *desktop*

Na Figura G.2 é mostrado o diagrama modelo entidade-relacionamento do banco de dados do sistema Web.

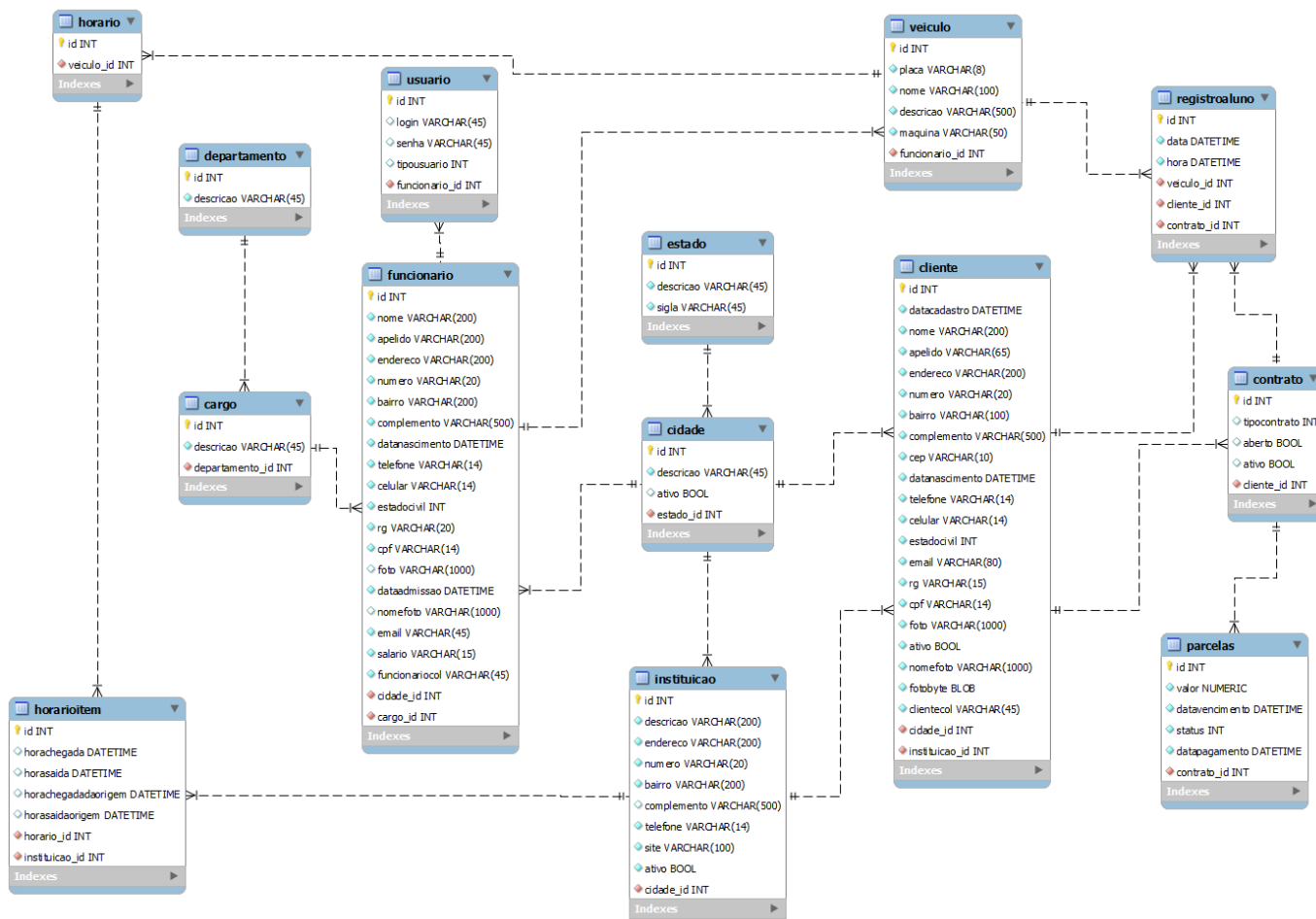


Figura G.2 – MER Sistema Web

**ANEXO A - PROPOSTA DE TRABALHO DE CONCLUSÃO DE
CURSO**

**MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO**

Heliton Rodrigo da Silva

**PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO
Sistema de Transporte Escolar**

CORNÉLIO PROCÓPIO

NOVEMBRO – 2011

**MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO**

Heliton Rodrigo da Silva

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

Sistema de Transporte Escolar

Proposta de trabalho de conclusão de curso apresentada como requisito parcial a obtenção do grau de Tecnólogo em Desenvolvimento de Sistemas de Informação pela Universidade Tecnológica Federal do Paraná – Campus Cornélio Procópio.

Orientador: prof. Andre Luiz Przybysz

CORNÉLIO PROCÓPIO

NOVEMBRO – 2011

LISTA DE FIGURAS

FIGURA 1 – ARQUITETURA DE HARDWARE DO SISTEMA	129
FIGURA 2 – CICLO DE VIDA (XP)	133
FIGURA 3 – CRONOGRAMA INICIAL RESUMIDO.....	135

LISTA DE TABELAS

TABELA 1 – PROBLEMAS ENCONTRADOS.....	122
TABELA 2 – OBJETIVOS ESPECÍFICOS	125
TABELA 3 – DESCRIÇÃO DO USUÁRIO DO SISTEMA – VISITANTE	128
TABELA 4 – DESCRIÇÃO DO USUÁRIO DO SISTEMA – ATENDENTE	128
TABELA 5 – DESCRIÇÃO DO USUÁRIO DO SISTEMA – ADMINISTRADOR.....	128
TABELA 6 – DESCRIÇÃO DO USUÁRIO DO SISTEMA – MOTORISTAS CADASTRADOS	128
TABELA 7 – PONTOS POR FUNÇÃO.....	139

LISTA DE ABREVIATURAS E SIGLAS

UTFPR	Universidade Tecnológica Federal do Paraná
CDC	Código de Defesa do Consumidor
UML	<i>Unified Modeling Language</i>
SGBD	Sistema Gerenciador de Banco de Dados
ASP	<i>Active Server Pages</i>
XML	<i>eXtensible Markup Language</i>
C#	<i>C Sharp</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
BD	Banco de Dados
3G	3º Geração

SUMÁRIO

1 INTRODUÇÃO.....	120
2 JUSTIFICATIVA.....	121
3 VISÃO DA SOLUÇÃO – OBJETIVOS	123
4 ESCOPO DA SOLUÇÃO – OBJETIVOS ESPECÍFICOS.....	124
5 LIMITES E RESTRIÇÕES DA SOLUÇÃO.....	126
6 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA	127
6.1 IDENTIFICAÇÃO DOS USUÁRIOS.....	127
6.2 HIERARQUIA ENTRE OS USUÁRIOS.....	127
7 TECNOLOGIAS UTILIZADAS	129
7.1 WINDOWS FORMS (C#).....	129
7.2 VISUAL STUDIO 2010.....	129
7.3 LINGUAGEM DE PROGRAMAÇÃO C#	129
7.4 NHIBERNATE.....	129
7.5 MICROSOFT SQL SERVER 2008.....	130
7.6 MICROSOFT ASP.NET 4.0	130
7.7 CSS - CASCADING STYLE SHEETS.....	130
7.8 HTML	130
7.9 VISUAL PARADIGM FOR UML	130
8 ARQUITETURA DO SISTEMA	131
9 METODOLOGIA DE DESENVOLVIMENTO	133
10 CRONOGRAMA	135
11 RECURSOS NECESSÁRIOS	136
12 REFERÊNCIAS.....	137
13 BIBLIOGRAFIA	138
14 ANEXO A - TABELA DE PONTOS POR FUNÇÃO.....	139
15 ANEXO B - MODELO ENTIDADE-RELACIONAMENTO.....	140

1 INTRODUÇÃO

Este documento tem por objetivo descrever uma visão inicial para o projeto intitulado Sistema de Transporte Escolar, referente à proposta de trabalho de diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná (UTFPR). Esse escopo servirá como base para acompanhamento e avaliação do projeto.

O projeto será conduzido pela J.SL Ferracini Transporte LTDA, empresa fundada na data de dezessete de Agosto de 2004, situada na cidade de Sertanópolis no Estado do Paraná, com seu escritório situado na rua Dr. Gervásio Morales, 465, e que tem como ramo de atividade o fretamento de seus veículos para excursões ou viagens, e como principal o transporte intermunicipal de alunos de diversas instituições de ensino.

O tema refere-se à criação de um sistema que visa facilitar o controle das informações da empresa citada e um *site* onde ela poderá divulgar o seu trabalho.

Nas próximas seções deste documento, serão apresentados os principais problemas que motivam a realização deste trabalho; as necessidades principais que a empresa enfrenta atualmente; as funcionalidades do sistema, ambiente, plataforma e tecnologias necessárias para a implementação; como o sistema será desenvolvido; os limites e restrições da solução; a arquitetura utilizada para o desenvolvimento do sistema e, por fim, o cronograma para o projeto.

2 JUSTIFICATIVA

Visto que a empresa cresceu e agora possui vários clientes, carros, contratos, funcionários, e o controle de todos os seus registros ainda são feitos em planilhas e sem nenhum padrão de armazenamento pelo fato dela não possuir um sistema informatizado implantado, surgiu a necessidade da implantação de um sistema para controle de suas informações.

E pelos motivos descritos acima, a empresa atualmente esta sofrendo vários problemas, onde alguns são: a inadimplência de muitos de seus clientes, onde estes usufruem de seus veículos com mensalidades atrasadas ou até mesmo sem ter efetuado nenhum pagamento; a falta de controle do cadastro dos clientes e funcionários; nenhum controle de uso dos clientes contratantes nos carros da empresa; relatórios; etc.

Para solucionar o problema da inadimplência, os representantes da empresa sabem da impossibilidade de manter um edital de clientes inadimplentes, pelo motivo de estar de acordo com o artigo 42 da seção V “da cobrança de dívidas”, do código de proteção e defesa do consumidor (CDC) (1997), onde diz que o consumidor inadimplente não pode ser exposto a ridículo, nem ser submetido a qualquer tipo de constrangimento ou ameaça na hora da cobrança, então se faz necessário o desenvolvimento de um sistema no qual o empresário terá total controle sobre suas contas a receber, sendo essas contas, os boletos de pagamento de seus clientes e também um controle de frequência de uso de seus veículos por parte dos contratantes.

Os representantes também sabem que um *site* é um dos instrumentos de publicidade mais eficientes que existem hoje em dia, sendo assim perceberam que é de extrema importância à divulgação do seu trabalho em um *site*.

Na tabela 1 abaixo estão descritos quais são os principais problemas encontrados, o que estão afetando, qual o impacto gerado, e a solução encontrada para os problemas citados.

O principal problema é:	Controle de clientes, controle de contas a receber, históricos de uso, relatórios.
Que afeta...	Empresa fica sem nenhum controle de seus usuários e contas a receber.
O impacto disto é...	Inadimplência,
A solução seria...	Sistema para controlar seus clientes e suas respectivas contas junto à empresa, controle de frequência de uso e emissão de relatórios.

Tabela 1 – Problemas encontrados

3 VISÃO DA SOLUÇÃO – OBJETIVOS

O sistema a ser desenvolvido visa sanar vários problemas da empresa citada, principalmente, o da falta de controle de suas contas a receber, no que acarreta na inadimplência de seus clientes, e também tem como principais objetivos:

- Garantir a empresa uma maior agilidade e segurança dos dados cadastrados;
- Disponibilizar um melhor controle de seus clientes e funcionários;
- Possibilitar a geração de históricos de frequência de todos os clientes que estarão utilizando os carros da empresa;
- Possibilitar um melhor controle das contas a receber, de acordo com os contratos fechados com os clientes;
- Possibilitar a emissão de relatórios.

Nos ônibus será implantada uma versão local do sistema, onde possibilitará a leitura do código do cliente através de um leitor de código de barras na entrada do ônibus, assim gerando um histórico de frequência de cada cliente usuário do ônibus e ainda retornando ao motorista a situação do cliente junto à empresa.

E com o propósito de solucionar o problema publicitário, também será construído um *site*, no qual a empresa poderá fazer a divulgação de todo o seu trabalho ao público.

4 ESCOPO DA SOLUÇÃO – OBJETIVOS ESPECÍFICOS

Nessa seção, estão descritas na Tabela 2 abaixo, as principais funcionalidades da aplicação a ser desenvolvida para a empresa citada, sendo que cada uma esta classificada como essencial ou desejável.

As funcionalidades abaixo serviram de base para a contagem dos pontos por função do sistema, onde pode ser vista na Tabela 7 no ANEXO A.

Necessidades	Categoria
Controle de usuários do sistema. <ul style="list-style-type: none"> • Fornecer papéis aos usuários do sistema, proporcionando um sistema de login para aumentar a segurança das informações. 	<i>Essencial</i>
Controle de funcionários da empresa. <ul style="list-style-type: none"> • O usuário autorizado poderá cadastrar, excluir, alterar, consultar e definir papel que cada funcionário terá dentro do sistema de acordo com o gerenciamento de usuários do sistema. 	<i>Essencial</i>
Controle de instituições de ensino. <ul style="list-style-type: none"> • O sistema permitirá ao usuário autorizado efetuar o cadastro, exclusão, alteração e consulta das instituições de ensino. 	<i>Essencial</i>
Controle de clientes. <ul style="list-style-type: none"> • Disponibilizar ao usuário autorizado, efetuar cadastros, alterações, exclusões e consultas de seus clientes. 	<i>Essencial</i>
Controle de contas a receber. <ul style="list-style-type: none"> • Disponibilizar ao usuário autorizado, efetuar cadastros, alterações, exclusões e consultas das contas a receber. 	<i>Essencial</i>
Gerar carteira de identificação. <ul style="list-style-type: none"> • Após cadastro de um novo cliente, o sistema irá criar para impressão uma carteira de identificação do usuário, onde constará seu nome, data de validade, instituição de ensino e código com barras. 	<i>Essencial</i>
Gerar histórico de uso dos clientes. <ul style="list-style-type: none"> • Após a apresentação do cartão de identificação para leitura, o sistema vai gerar e armazenar automaticamente um histórico, 	<i>Essencial</i>

constando data e hora em que cada cliente entrou no ônibus.	
<p>Proporcionar módulo administrativo, para que se possa alterar conteúdo da página web.</p> <ul style="list-style-type: none"> • O usuário autorizado pelo sistema poderá alterar endereços, promoções, rotas e serviços que a empresa disponibilizará ao público. 	<i>Essencial</i>
<p>Emissão de relatórios.</p> <ul style="list-style-type: none"> • Relatório de clientes • Relatório de funcionários • Relatório de contratos em vigor • Relatório de contratos cancelados • Relatório de histórico de uso (frequência) • Relatório de fechamento mensal • Relatório de contas a receber por período • Relatório de parcelas atrasadas 	<i>Essencial</i>
<p>Registro de Instituições de Ensino</p> <ul style="list-style-type: none"> • Disponibilizar ao usuário autorizado, efetuar cadastros, alterações, exclusões e consultas das instituições de ensino. 	<i>Essencial</i>
<p>Registro de carros</p> <ul style="list-style-type: none"> • Disponibilizar ao usuário autorizado, efetuar cadastros, alterações, exclusões e consultas dos carros da empresa. 	<i>Essencial</i>
<p>Controle de rotas</p> <ul style="list-style-type: none"> • Disponibilizar ao usuário autorizado, efetuar cadastros, alterações, exclusões e consultas das rotas que são seguidas de acordo com cada carro. 	<i>Essencial</i>
<p>Controle de contratos</p> <ul style="list-style-type: none"> • Disponibilizar ao usuário autorizado, efetuar cadastros, alterações, exclusões e consultas dos contratos assinados. 	<i>Essencial</i>
<p>Emissão de e-mails aos clientes.</p> <ul style="list-style-type: none"> • Envio de e-mails, sobre promoções, mudanças de rotas, excursões, etc., para clientes cadastrados. 	<i>Desejável</i>

Tabela 2 – Objetivos Específicos

5 LIMITES E RESTRIÇÕES DA SOLUÇÃO

O sistema proposto não fará controle financeiro da empresa, será feito apenas o controle das contas a receber de seus clientes. Tratando-se de um sistema web, será necessário terem instalados navegadores nas máquinas e as mesmas estarem conectadas à internet.

A integridade dos dados será controlada pelo servidor do site em que o sistema será hospedado.

O sistema será construído levando em consideração que a infraestrutura de rede da empresa citada esteja pronta e configurada.

6 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA

6.1 IDENTIFICAÇÃO DOS USUÁRIOS

Usuários Internos: Administrador e Atendente

Usuários Externos: Funcionários Cadastrados

Usuários do Negócio: Administrador e Atendente

Usuários do não-negócio: Visitantes

6.2 HIERARQUIA ENTRE OS USUÁRIOS

1. **Administrador** - Somente o administrador terá total acesso ao sistema, podendo assim gerenciar todos os usuários, clientes, contas, relatórios e também efetuar a manutenção de alguns módulos da aplicação.
2. **Atendente** – Poderá gerenciar os cadastros de clientes e contas a receber.
3. **Motoristas Cadastrados** – Terá acesso somente nos módulos clientes obtendo apenas o retorno da situação financeira do cliente junto à empresa e atualizar histórico.
4. **Visitantes** – Apenas poderão verificar promoções, rotas, serviços, etc.

Nas próximas quatro tabelas serão apresentadas as descrições e os objetos acessados no modo de leitura e de gerenciamento de cada tipo de usuário do sistema proposto.

Nome do Grupo	Visitante
Descrição	Visitantes serão os usuários que acessam o sistema para consulta das informações dispostas na página web.
Objetos acessados modo leitura	Apenas poderão verificar o conteúdo do site, como: rotas, preços, horários, serviços, endereços, etc.
Objetos acessados no	Nenhum. Gerenciamento de conteúdo não é permitido para

modo de gerenciamento	de	esse grupo de usuários.
-----------------------	----	-------------------------

Tabela 3 – Descrição do usuário do sistema – Visitante

Nome do Grupo		Atendente
Descrição		Atendentes poderão efetuar os cadastros dos clientes e novos funcionários, verificar dados cadastrais dos clientes e funcionários e gerenciar as contas a receber.
Objetos acessados no modo leitura	no	Verificar dados dos clientes e contas a receber.
Objetos acessados no modo de gerenciamento	no de	Gerenciar cadastros de clientes e funcionários.

Tabela 4 – Descrição do usuário do sistema – Atendente

Nome do Grupo		Administrador
Descrição		Gerenciar todo o sistema.
Objetos acessados no modo leitura	no	Relatórios e Históricos de uso.
Objetos acessados no modo de gerenciamento	no de	Clientes, funcionários, usuários do sistema, contas a receber da empresa.

Tabela 5 – Descrição do usuário do sistema – Administrador

Nome do Grupo		Funcionários Cadastrados
Descrição		Visualizar status dos clientes e atualizar histórico de uso.
Objetos acessados no modo leitura	no	Retorno da situação do cliente, ou seja, se ele está pendente ou em dia com suas contas junto a empresa.
Objetos acessados no modo de gerenciamento	no de	Nenhum. Gerenciamento de conteúdo não será permitido para esse grupo de usuário.

Tabela 6 – Descrição do usuário do sistema – Motoristas Cadastrados

7 TECNOLOGIAS UTILIZADAS

As tecnologias que serão utilizadas no desenvolvimento do sistema e as ferramentas que serão necessárias estão listadas abaixo:

7.1 WINDOWS FORMS (C#)

Para o desenvolvimento da interface gráfica, foi escolhido o componente *smart-client* do *.NET Framework*, por ser totalmente compatível com as versões XP, Vista e Seven do sistema operacional Windows.

7.2 VISUAL STUDIO 2010

Como ferramenta de desenvolvimento será utilizada o *Visual Studio 2010*, conjunto de ferramentas unificadas, ferramenta que simplifica a criação, depuração e implantação de aplicações.

7.3 LINGUAGEM DE PROGRAMAÇÃO C#

É uma linguagem de programação orientada a objetos, fortemente tipada, desenvolvida pela Microsoft como parte da plataforma *.NET*.

7.4 NHIBERNATE

NHibernate é uma biblioteca (*Framework*) baseada em *.NET* para persistir os objetos para bases de dados relacionais.

7.5 MICROSOFT SQL SERVER 2008

O *Microsoft SQL Server 2008* é um Sistema Gerenciador de Banco de Dados (SGBD) completo e integrado de gerenciamento e análise de dados, permitindo que organizações gerenciem informações críticas e confidenciais de forma confiável.

7.6 MICROSOFT ASP.NET 4.0

ASP.NET é a plataforma da Microsoft para o desenvolvimento de aplicações *web* e é o sucessor da tecnologia *ASP*.

7.7 CSS - CASCADING STYLE SHEETS

Nada mais é, que um documento onde são definidas regras de formatação ou de estilos, a serem aplicadas aos elementos estruturais de marcação. A finalidade das CSS é a de retirar do HTML toda e qualquer declaração que vise a formatação, a apresentação do documento.

7.8 HTML

HTML é uma linguagem com a qual se definem as páginas *web*.

7.9 VISUAL PARADIGM FOR UML

A ferramenta *Visual Paradigm* é uma ferramenta CASE voltada para UML (*Unified Modeling Language*), sendo possível trabalhar com os mais diversos diagramas da UML, como diagramas de caso de uso, diagramas de classes, diagramas de seqüência, dentre outros.

8 ARQUITETURA DO SISTEMA

O sistema será desenvolvido em uma arquitetura em três camadas, sendo:

1. Camada de apresentação: responsável por exibir os dados ao usuário.
2. Camada de negócio: responsável por toda a regra de negócio.
3. Camada de dados: responsável pelo acesso ao banco de dados.

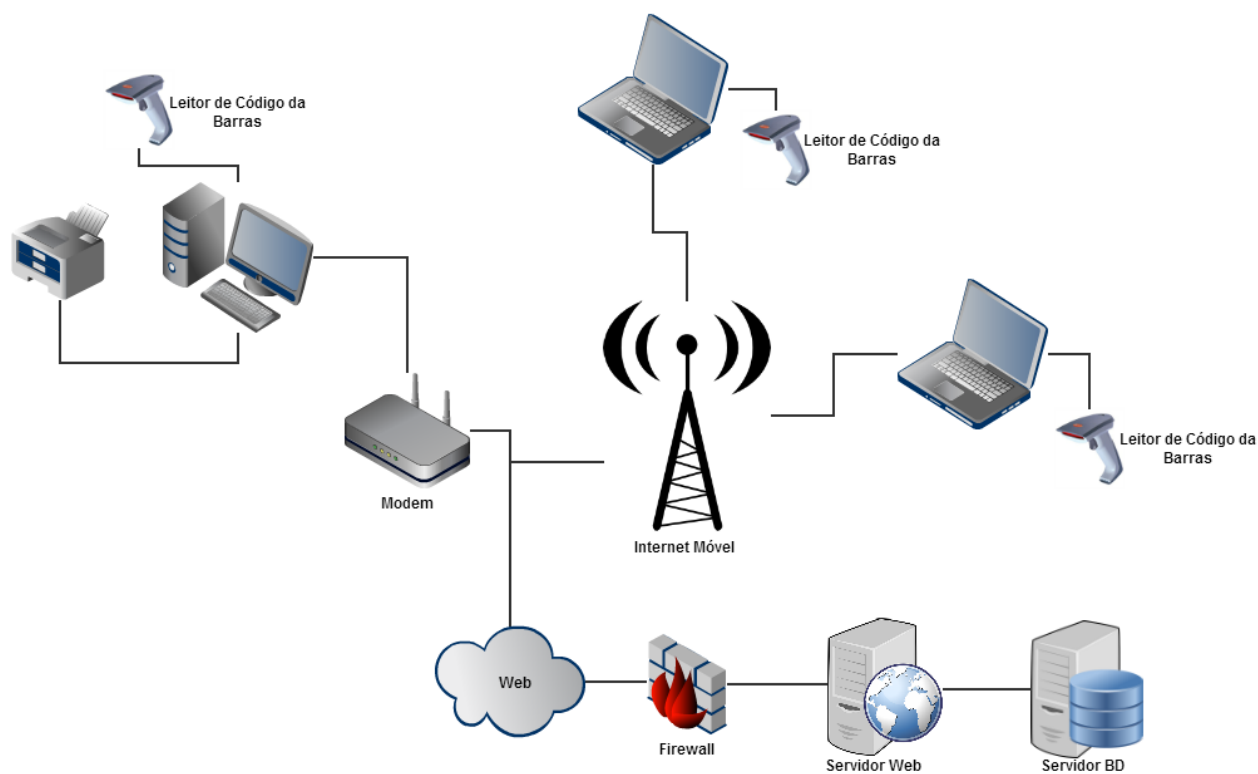


Figura 1 – Arquitetura de Hardware do Sistema

- Servidor BD: É o servidor de banco de dados, tem a função de armazenar os dados manipulados pelos sistemas hospedados no servidor web. Para o *website* este servidor deve ter instalado o *Microsoft SQL Server 2008*;

- Servidor Web: Tem a função de processar e armazenar o código dos sistemas web hospedados nele. Para rodar a aplicação deve possuir o servidor *Internet Information Services (IIS) 6.0*, com *Microsoft .Net Framework 3.5* instalado;
- *Firewall*: A hospedagem deve garantir a segurança ao acesso e a manipulação das informações armazenadas em seus servidores, isso é dada com um dispositivo de uma rede de computadores que tem por objetivo aplicar uma política de segurança a um determinado ponto da rede.
- Modem Roteador: Conexão com a internet.
- Internet 3G: Responsável pelo acesso a internet utilizando a tecnologia 3G oferecida pelas operadoras de telefonia móvel.
- Leitores de código de barras: Periférico responsável pela leitura de código de barras.

9 METODOLOGIA DE DESENVOLVIMENTO

No desenvolvimento do sistema será utilizado um processo denominado eXtreme Programming ou simplesmente XP, que é uma das mais conhecidas metodologias de desenvolvimento de software que segue os princípios do Manifesto Ágil. De acordo com (ASTEELS, 2002), o processo XP, é um método bastante flexível e oferece a possibilidade da comunicação com o cliente ser mais intensa que nos métodos tradicionais, devendo o cliente estar sempre disponível para tirar as dúvidas, rever requisitos e atribuir propriedades, utilizando-se de sistema de nomes, em vez de termos técnicos para facilitar a comunicação.

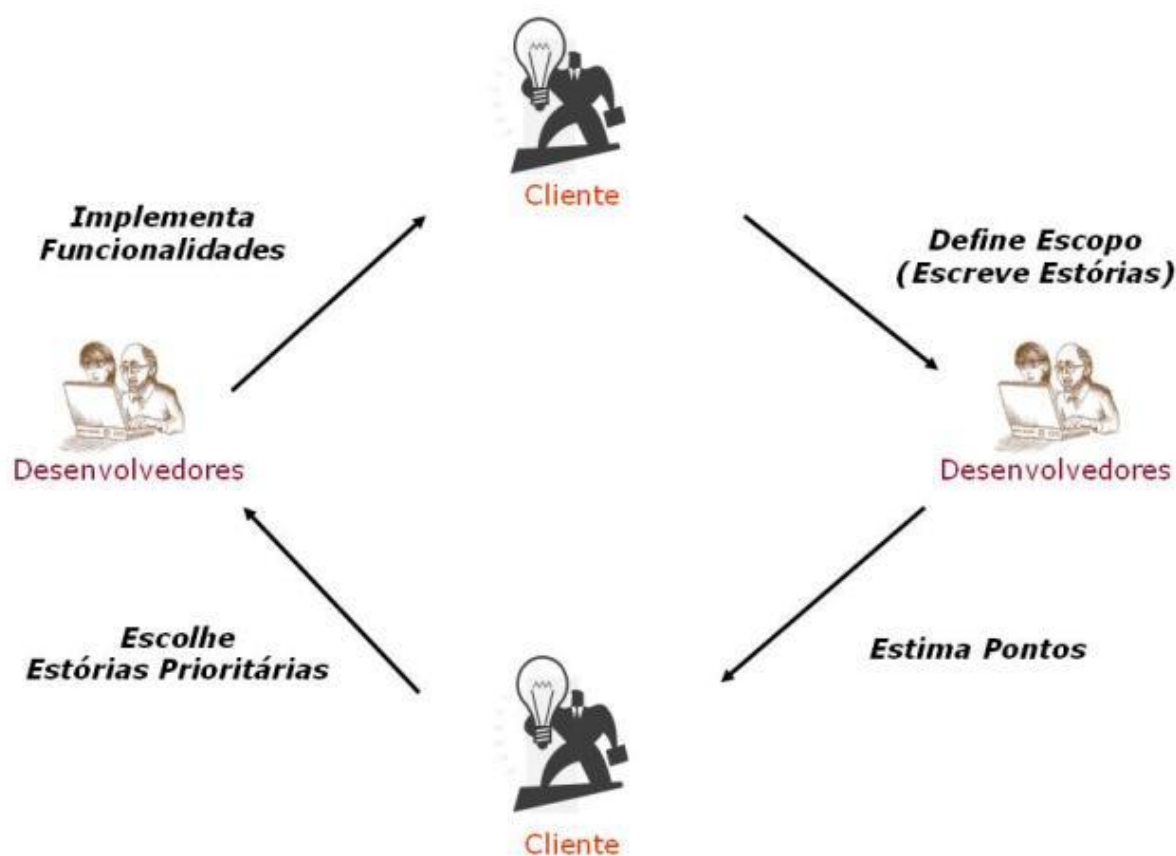


Figura 2 – Ciclo de Vida (XP)

Por ser uma metodologia bastante flexível e adaptável, houve uma adaptação de um modelo do BECK alterando algumas das etapas das fases de desenvolvimento, a fim de proporcionar um melhor rendimento, controle e planejamento do desenvolvimento deste projeto.

A definição da metodologia sugere que o processo de Análise de Requisitos possui como principal objetivo o estudo de viabilidade do produto, a identificação e análise dos riscos do negócio e do projeto. Define uma arquitetura que ofereça suporte estável à implementação, correspondendo a atividades de concepção e elaboração. Para melhor aplicação deste processo três papéis foram identificados, cada um contendo atividades e prazos para sua execução.

A documentação do projeto possui alguns diagramas da UML, como os diagramas de casos de uso, diagramas de classes e diagramas de atividades. Essa documentação somente tem início quando é definido o conjunto de requisitos a serem atendidos na versão em desenvolvimento.

Toda a documentação elaborada é discutida em reuniões. A implementação das funcionalidades tem início somente após a aprovação dos documentos. Estas atividades de entendimento e preparação do projeto possuem o prazo de uma semana para seu término.

Na fase de Monitoria de Projetos, é realizada a criação do código-fonte e a execução de testes no sistema. Durante esta fase os diagramas de classe e de banco de dados são constantemente atualizados, com isto pretende-se deixar a documentação condizente com a implementação do sistema. Este processo requer o acompanhamento diário ou semanal dos gerentes de requisitos e do projeto, para que desvios na implementação dos requisitos possam ser identificados e corrigidos a tempo sem acarretar problemas ao cronograma.

Os testes estão sob responsabilidade de todos os envolvidos no projeto. Durante o processo serão gerados vários artefatos que apresentarão um feedback sobre o andamento do projeto para a empresa.

10 CRONOGRAMA

Esta seção apresenta um cronograma inicial resumido para o projeto, destacando quais serão os principais marcos do projeto, e quando eles ocorrerão.

	20 11					20 12						
	Novembro		Dezembro			Janeiro		Fevereiro		Março	Abril	
Concepção	█	█	█	█	█							
Elaboração					█	█	█	█	█			
Construção								█	█	█	█	█
Iteração 1								█	█	█	█	
Iteração 2										█	█	█
Transição												█
Monografia						█	█	█			█	█

Figura 3 – Cronograma Inicial Resumido

11 RECURSOS NECESSÁRIOS

Para o desenvolvimento do sistema proposto, será necessário um microcomputador com o *Microsoft Windows 7* instalado como sistema operacional, plataforma *.Net Framework 4.0* e demais ferramentas citadas em Tecnologias Utilizadas deste documento instaladas e uma máquina leitora de código de barras.

12 REFERÊNCIAS

ASTELS, D. Extreme programming : Guia Prático / Dave Astels, Granville Miller, Miroslav Novak ; Tradução de Kátia Roque. – Rio de Janeiro : Ed. Campus, 2002.

BECK, K. Programação Extrema (XP) Explicada: Acolha as mudanças. Porto Alegre: Bookman, 2004.

BRASIL. Código de Defesa do Consumidor – Nova ed. ver., atual. e ampli. com o Decreto nº 2.181, de 20 de Março de 1.997- Brasília, : Ministério da Justiça, 2010. 124p.

13 BIBLIOGRAFIA

Comissão de Normalização de trabalhos Acadêmicos. **Normas para elaboração de trabalhos acadêmicos**. Sistema de Bibliotecas, Universidade Tecnológica Federal do Paraná. Curitiba, 2008.

14 ANEXO A - TABELA DE PONTOS POR FUNÇÃO

A tabela a seguir mostra a contagem de pontos por função não ajustada dos requisitos do projeto proposto.

<u>Legenda</u> ALI = Arquivo Lógico Interno AIE = Arquivo de Interface Externa TER = Tipo de Elemento de Registro TED = Tipo de Elemento de Dados	ALI Baixa (ALlb) 16 112 Média (ALIm) 0 0 Alta (ALla) 0 0
<u>Legenda</u> IE = Input Externo TAR = Tipo de Arquivos Referenciados	AIE Baixa (AIEb) 0 0 Média (AIEm) 0 0 Alta (AIEa) 0 0
<u>Legenda</u> OE = OutPut Externo	IE Baixa (IEb) 13 39 Média (IEm) 2 8 Alta (IEa) 1 6
<u>Legenda</u> OE = OutPut Externo	OE Baixa (OEb) 5 20 Média (OEm) 4 20 Alta (OEa) 8 56
<u>Legenda</u> Peso: representado em pontos por função	CE Baixa (OEb) 0 0 Média (OEm) 0 0 Alta (OEa) 0 0
	Total 261

Tabela 7 – Pontos por Função

15 ANEXO B - MODELO ENTIDADE-RELACIONAMENTO

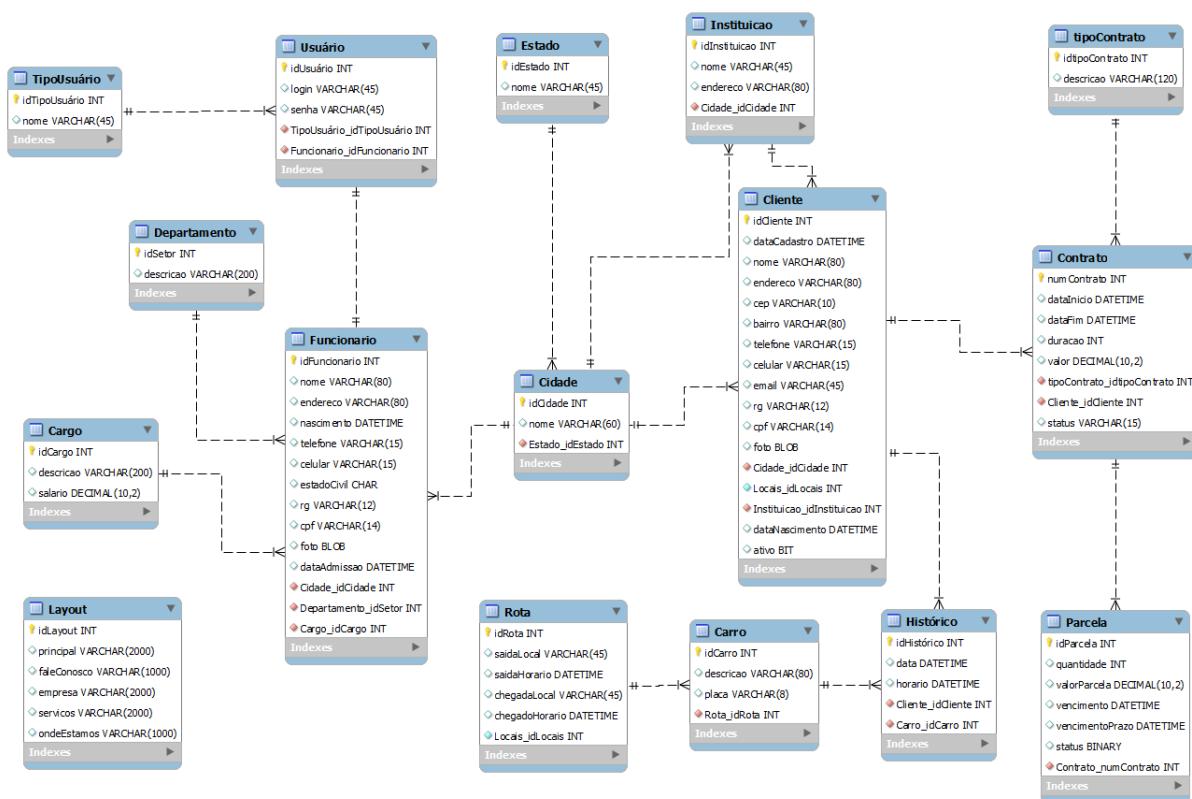


Imagem B.1 - MER - Modelo Entidade-Relacionamento