



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

Rodrigo Pereira de Souza

**SISTEMA WEB PARA TRANSPORTADORA USANDO ADOBE FLEX
E API GOOGLE MAPS**

**Assis/SP
2011.**

Av. Getúlio Vargas, 1200 – Vila Nova Santana – Assis – SP – 19807-634
Fone/Fax: (0XX18) 3302 1055 homepage: www.fema.edu.br



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

Rodrigo Pereira de Souza

SISTEMA WEB PARA TRANSPORTADORA USANDO ADOBE FLEX E API GOOGLE MAPS

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

Orientador: Prof. Dr. Luiz Ricardo Begosso

Área de Concentração: Informática.

**Assis/SP
2011.**

FICHA CATALOGRÁFICA

SOUZA, Rodrigo Pereira

Sistema Web para transportadora usando Adobe Flex e API Google Maps / Rodrigo Pereira de Souza. Fundação Educacional do Município de Assis – FEMA – Assis, 2011.

50p.

Orientador: Prof. Dr. Luiz Ricardo Begosso.

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

1.Adobe Flex. 2.Google Maps. 3.Java.

CDD:001.6
Biblioteca da FEMA.



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

SISTEMA WEB PARA TRANSPORTADORA USANDO ADOBE FLEX E API GOOGLE MAPS

Rodrigo Pereira de Souza

Trabalho de Conclusão de Curso apresentado
ao Instituto Municipal de Ensino Superior de
Assis, como requisito de Curso de Bacharelado
em Ciência da Computação, analisado pela
seguinte comissão examinadora:

Orientador: Prof. Dr. Luiz Ricardo Begosso

Área de Concentração: Informática.

**Assis/SP
2011.**

DEDICATORIA

Dedico este trabalho, em primeiro lugar, à minha mãe Luzia de Fatima Pereira de Souza e ao meu pai Vilson de Souza, pelo grande incentivo, dedicação, compreensão, por acreditar em mim, sempre me ajudar e principalmente pelo amor dessa e de outras passagens da minha vida, e dedico também a minha família e amigos.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por sempre me dar forças, saúde e sabedoria para concluir mais essa etapa da minha vida.

À minha mãe Luzia de Fatima Pereira de Souza e ao meu pai Vilson de Souza, pela grande dedicação, compreensão, por sempre me ajudar, me dar forças, e incentivo para nunca desistir, mesmo nas horas mais difíceis, e principalmente pelo amor que sempre me deram.

Agradeço também ao meu orientador Luiz Ricardo Begosso, a quem tenho muita admiração, e que me forneceu conhecimento e ajudou a concluir este trabalho.

E também a todos os outros professores e amigos que conheci e que me ajudaram ao longo desses anos.

RESUMO

Este projeto tem por objetivo criar um sistema WEB para uma transportadora, simples, eficaz e de fácil utilização com ferramentas que facilitam a elaboração e melhoram o desempenho do sistema.

Nos módulos principais do sistema, o administrador terá acesso à parte de movimentação que calcula o valor do frete com os dados do cliente, e em seguida gera uma fatura a ser paga pelo cliente.

O administrador terá acesso também a uma parte do sistema para baixar as faturas pendentes além de ter a parte de cadastro, alteração, exclusão e pesquisa de clientes, funcionários, produtos, veículos, e tipo de veículos.

O gerente terá acesso apenas a relatórios de clientes, funcionários, produtos, veículos e tipo de veículos, além de relatórios de clientes com faturas pendentes.

Para o desenvolvimento foi utilizado a UML para a modelagem do sistema, a linguagem de programação Java e a IDE Eclipse, o Adobe Flex para desenvolvimento de uma interface agradável para o sistema, o framework Hibernate para a persistência de dados, a API Google Maps para a manipulação de mapas, o BlaseDS para comunicação entre o Flex e o Eclipse, e o Banco de dados Firebird.

Palavras-chave: Adobe Flex; Google Maps; Java.

ABSTRACT

This Project aims to create a carrier web-based system, simple, effective and easy to use with tools that facilitate the development and improve system performance.

In the main modules of the system, the administrator will have access to the movement that calculates the amount of freight to the customer data, and then generates an invoice to be paid by customer.

The administrator also has access to a part of the system to lower the outstanding invoices in addition to part of the registration, change, delete and search for customers, employees, products, vehicles, and type of vehicles.

The manager will have access only to reports of customers, employees, products, vehicles and type of vehicles, along with reports of customers with outstanding invoices.

For the development software, the UML was used to model the system, the Java programming language and Eclipse IDE, Adobe Flex to develop a nice interface to the system, the Hibernate framework for persistence of data, the API Google Maps for manipulation of maps, BlaseDS for communication between Flex and Eclipse, and database Firebird.

Keywords: Adobe Flex; Google Maps; Java.

Lista de Imagens

| | |
|---|----|
| Figura 1 - Arquitetura do Eclipse (S. Albuquerque; Eduardo, 2005, p.23)..... | 17 |
| Figura 2 - Forma em que o framework Flex é dividido (Martins; Fabio, 2011). | 18 |
| Figura 3 - Rota traçada no Google Maps. | 19 |
| Figura 4 - Ator Caixa associado ao Use Case Fechar Venda. | 22 |
| Figura 5 - Exemplo simples de uma classe Produto. | 23 |
| Figura 6 - Mapa mental das funções do sistema. | 24 |
| Figura 7 - Diagrama de Caso de Uso completo. | 25 |
| Figura 8 - Caso de Uso Manter Clientes. | 26 |
| Figura 9 - Caso de Uso Manter Funcionários..... | 28 |
| Figura 10 - Caso de Uso Manter Produtos..... | 30 |
| Figura 11 - Caso de Uso Manter Tipo Veículos..... | 32 |
| Figura 12 - Caso de Uso Manter Veículos. | 34 |
| Figura 13 - Caso de Uso Movimentar Transporte e Gerar Fatura. | 36 |
| Figura 14 - Caso de Uso Faturar Notas Abertas. | 38 |
| Figura 15 - Caso de Uso Analisar Faturas Abertas. | 39 |
| Figura 16 - Caso de Uso Analisar Relatórios. | 40 |
| Figura 17 - Diagrama de Classe do projeto..... | 41 |
| Figura 18 - Modelo Entidade Relacionamento..... | 42 |

Lista de tabelas

| | |
|---|----|
| Tabela 1 – Cronograma. | 15 |
| Tabela 2 – Dicionário de dados da tabela Adiantamento. | 43 |
| Tabela 3 - Dicionário de dados da tabela Carta_frete. | 43 |
| Tabela 4 - Relacionamentos da tabela Carta_frete. | 43 |
| Tabela 5 - Dicionário de dados da tabela Conhecimento_trans. | 45 |
| Tabela 6 - Relacionamentos da tabela Conhecimento_transp. | 45 |
| Tabela 7 - Dicionário de dados da tabela Cliente. | 45 |
| Tabela 8 - Dicionário de dados da tabela Fatura. | 46 |
| Tabela 9 - Relacionamentos da tabela Fatura. | 46 |
| Tabela 10 - Dicionário de dados da tabela Funcionário. | 46 |
| Tabela 11 - Dicionário de dados da tabela Produto. | 47 |
| Tabela 12 - Dicionário de dados da tabela Tipo_veiculo. | 47 |
| Tabela 13 - Dicionário de dados da tabela Veiculo. | 47 |
| Tabela 14 - Relacionamentos da tabela Veiculo. | 48 |

Sumário

| | |
|--|----|
| 1. INTRODUÇÃO | 12 |
| 1.1 OBJETIVO | 12 |
| 1.2 JUSTIFICATIVA | 13 |
| 1.3 ESTRUTURA DO TRABALHO..... | 13 |
| 1.4 CRONOGRAMA..... | 15 |
| 2. AMBIENTE DE DESENVOLVIMENTO | 16 |
| 2.1 ECLIPSE..... | 16 |
| 2.2 ADOBE FLEX..... | 17 |
| 2.3 API GOOGLE MAPS..... | 19 |
| 2.4 HIBERNATE..... | 19 |
| 2.5 FIREBIRD | 20 |
| 2.6 BLASEDS | 20 |
| 3. AMBIENTE DE MODELAGEM..... | 21 |
| 3.1 HISTÓRIA DA UML..... | 21 |
| 3.2 CONCEITOS..... | 21 |
| 4. MODELAGEM DO SOFTWARE | 24 |
| 4.1 DIAGRAMA DE CASO DE USO | 25 |
| 4.1.1 CASOS DE USO..... | 26 |
| 4.2 DIAGRAMA DE CLASSE | 41 |
| 4.3 MODELO ENTIDADE RELACIONAMENTO | 42 |
| 4.4 DICIONARIO DE DADOS | 43 |
| 5. CONCLUSÃO..... | 49 |
| 5.1 CONSIDERAÇÕES FINAIS | 49 |
| 5.2 TRABALHOS FUTUROS | 49 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 50 |

1. INTRODUÇÃO

Atualmente, todas as empresas necessitam de um sistema para gerenciar seus problemas e ajudar a ter um rendimento melhor sobre seus trabalhos. Essas necessidades fazem com que os profissionais de TI estejam muito bem capacitados para atender com muita eficiência o mercado de trabalho de hoje em dia.

Em uma transportadora, manter cadastros, movimentar fretes, gerenciar todos os transportes feitos e faturas que não foram pagas, podem se tornar tarefas muito difíceis para serem feitas sem ter algo para gravar seus dados, busca-los a hora que quiser com rapidez e podem acarretar prejuízo.

Este sistema é destinado a auxiliar e gerenciar de maneira fácil os serviços desenvolvidos dentro de uma transportadora utilizando e mostrando ferramentas que são muito eficientes para desenvolvimento de um sistema e fáceis de utilizá-las.

1.1 OBJETIVO

O objetivo desse trabalho é desenvolver um sistema WEB para uma transportadora que ajude a calcular o preço dos serviços prestados aos seus clientes e gerenciar de forma eficaz seus dados.

A finalidade do sistema é fazer com que ele tenha cadastros, pesquisas e relatórios de clientes, funcionários, veículos, entre outras informações necessárias para gerenciar os serviços a serem prestados.

O valor do frete será calculado com a ajuda do Google Maps que vai marcar em um mapa a rota do serviço que vai ser prestado a um determinado cliente cadastrado, pegando então as informações da quilometragem e adicionando outras informações adicionais como, porcentagem de lucro, preços e gerando o frete.

1.2 JUSTIFICATIVA

O intuito deste trabalho é criar um sistema que seja acessado pela internet para que o usuário possa utiliza-lo a partir de qualquer lugar em casos de ter que fazer alguma alteração em seus serviços com urgência.

Além de ser acessado pela internet, ele utilizará o Google Maps, que fornecerá um mapa para o usuário receber informação sobre a rota do serviço e sobre a localização dos pontos de origem e destino do transporte, além do sistema fornecer também o controle sobre cadastros, funcionários que estão em serviços e que estão disponíveis, entre outras funções para ajudar o usuário a ter um bom rendimento em sua empresa.

O Adobe Flex foi escolhido para auxiliar na construção da página, pois oferece plugin para o Eclipse, com isso pode-se trabalhar com a linguagem de programação Java e utilizar os vários componentes do Flex para desenvolver o layout da pagina, tudo muito simples e fácil de usar.

Outras ferramentas como Hibernate e banco de dados FireBird, serão utilizados pois, são muito uteis na parte de manipulação de dados do sistema. O FireBird porque é simples, fácil de usar e muito bom, e o Hibernate porque ajuda no trabalho com banco de dados fazendo mapeamento das tabelas do banco e facilitando a manipulação dos dados.

1.3 ESTRUTURA DO TRABALHO

Este trabalho esta dividido em quatro capítulos.

- O capítulo 1 é esta introdução, mostrando o que será feito neste projeto, o que será utilizado para desenvolver este projeto e o cronograma.
- O capítulo 2 apresenta o ambiente de desenvolvimento, descrevendo as ferramentas que serão utilizadas para desenvolver o sistema,

- O capítulo 3 apresenta o ambiente de modelagem, descrevendo a UML que será a linguagem de modelagem deste projeto, mostrando um pouco de história e conceitos dela como diagrama de caso de uso, diagrama de classe e etc.
- O capítulo 4 descreve a modelagem do software que é tudo que foi feito para a preparação do que o sistema terá para depois então começar a implementação, mostra os diagramas de caso de uso, o diagrama de classe, o modelo entidade relacionamento e o dicionário de dados.

1.4 CRONOGRAMA

| Datas | Atividades |
|-------------------------|--|
| 20.01.2011 – 10.02.2011 | Escolha do Orientador e do tema de TCC. |
| 15.02.2011 – 08.03.2011 | Desenvolvimento do Pré-Projeto para o orientador. |
| 08.03.2011 | Entrega do Pré-Projeto para o orientador. |
| 10.03.2011 – 20.04.2011 | Correção e ajustes do Pré-Projeto. |
| 20.04.2011 | Entrega do Pré-Projeto ao núcleo de monografia. |
| 21.04.2011 – 10.06.2011 | Desenvolvimento da documentação para o exame de qualificação. |
| 10.06.2011 | Entrega do documento do exame de qualificação para o orientador. |
| 11.06.2011 – 27.06.2011 | Correção e Ajustes do documento do exame de qualificação. |
| 27.06.2011 | Entrega da documentação do exame de qualificação no núcleo de monografia |
| 27.06.2011 – 20.07.2011 | Desenvolvimento do sistema e preparação para o exame de qualificação. |
| 25.07.2011 – 29.07.2011 | Bancas de qualificação. |
| 01.08.2011 – 03.10.2011 | Desenvolvimento sistema e da monografia. |
| 07.10.2011 | Entrega da monografia e do sistema para o orientador. |
| 08.10.2011 – 28.10.2011 | Entrega da monografia impressa. |
| 16.11.2011 – 30.11.2011 | Apresentação do TCC. |

Tabela 1 – Cronograma.

2. AMBIENTE DE DESENVOLVIMENTO

Para o desenvolvimento do projeto serão utilizadas as seguintes ferramentas:

2.1 Eclipse – O Eclipse é um IDE de código-fonte aberto, multiplataforma e extensível que foi construído em Java e sua plataforma é uma estrutura que fornece vários serviços que outros plug-ins podem aprimorar. Cada plug-in é desenvolvido para uma mesma plataforma que por fim acaba gerando um conjunto de ferramentas altamente integrado (Hemrajani; Anil, 2007, p.138).

Segundo Eduardo S. Albuquerque (2005, p. 22), o Eclipse é uma plataforma universal para a integração de ferramentas para diversos fins. Seu projeto foi lançado em 2001 quando a IBM doou o código fonte do WebsphereStudio Workbench no qual havia investido 40 milhões de dólares. Partindo daí foi criado o Eclipse Consortium com várias empresas importantes de tecnologia mundial para gerenciar e desenvolver a plataforma. Ela foi projetada para as seguintes funcionalidades:

- Suporte a criação de ferramentas para desenvolvimento de aplicações.
- Suporte a ferramentas para manipular tipos de conteúdo arbitrários (por exemplo, HTML, Java, C, JSP, EJB, XML, etc).
- Facilitar o uso de diferentes ferramentas para desenvolvimento num mesmo ambiente.
- Suporte GUI e ambientes de desenvolvimento de aplicação.
- Executar em vários sistemas operacionais.

A estrutura do Eclipse é definida em torno de pontos de extensão, que são locais no sistema onde outras ferramentas (plug-ins) trabalham juntas a ele. Ele conta com dois plug-ins principais que são o JDT e o PDE (S. Albuquerque; Eduardo, 2005, p.23).

A figura 1 mostra a arquitetura do Eclipse.

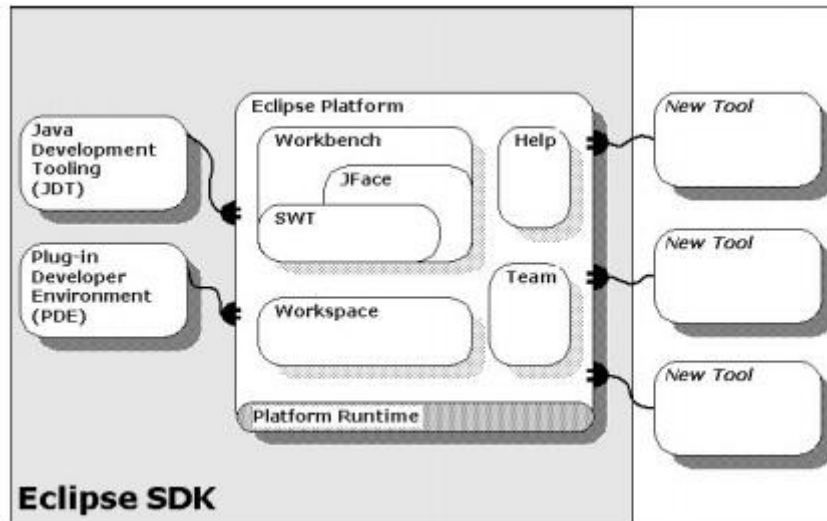


Figura 1 - Arquitetura do Eclipse (S. Albuquerque; Eduardo, 2005, p.23).

2.2 Adobe Flex – Criado pelo a Adobe o Flex possui uma estrutura altamente produtiva e é utilizado para desenvolvimento de aplicações RIA, com vários componentes que ajudam a desenvolver aplicativos de interfaces rica e eficaz. Ele utiliza também a linguagem de programação Action Script.

Com a compra da Macromedia em 2004, a Adobe Systems herdou uma estrutura utilizada para construir RIAs que era o Flex. A Adobe depois disso então resolveu tornar o Flex uma estrutura de software livre, enquanto a outra ferramenta baseada no eclipse permaneceu licenciável (Tarabal; Luiz, 2010).

Segundo Wellington César de Castro (2009), RIA é a abreviação de Rich Internet Applications ou Aplicações Ricas para Internet. São aplicações Web com características de uma aplicação desktop normal. Uma aplicação RIA transfere o processamento do cliente para o navegador, mas o processamento mais pesado mantém no servidor de aplicação.

O Flex possui funcionalidades para entrega eficientes de aplicações ricas e de alto desempenho. As aplicações em Flex rodam sobre o Adobe Flash Player 9, permitindo que os desenvolvedores estendam todas suas funcionalidades, criando aplicações mais robustas e integradas com arquitetura server-side. Além de deixar o Sistema com uma interface bonita, atrativa e segura (César de Castro; Wellington, 2009).

A figura 2 mostra a forma em que o Flex é dividido.

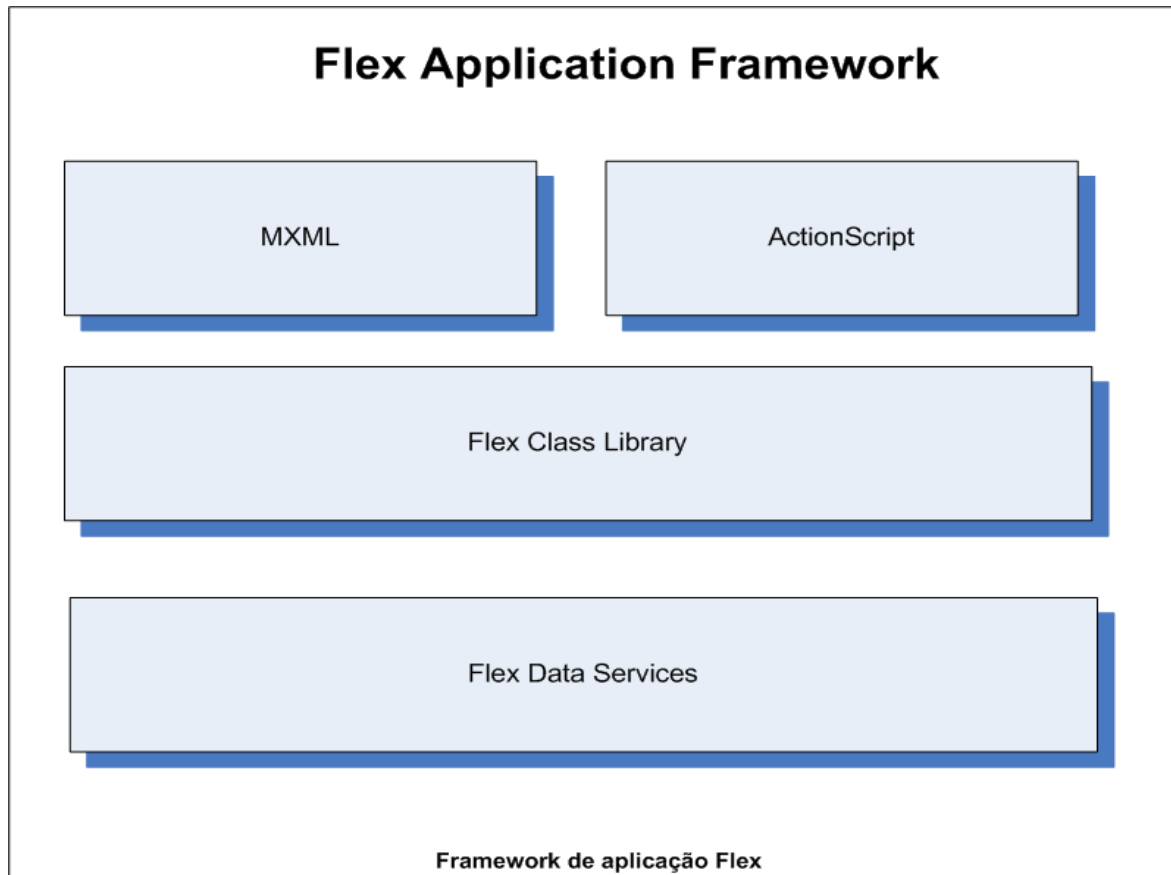


Figura 2 - Forma em que o framework Flex é dividido (Martins; Fabio, 2011).

MXML: Linguagem de marcação baseada em xml usada para definir a interface da aplicação. Pode ser usada também para definir aspectos não visuais da aplicação como acesso a dados no servidor. MXML inclui tags para componentes visuais como dataGrids, inputText, comboBox e menu. Inclui também tags para componentes não visuais como conexão com web services, data binding e efeitos de animação (Martins; Fabio).

ActionScript: Linguagem orientada a objetos semelhante ao Java e C#.

Flex Class Library: Bibliotecas de classes do Flex.

Flex Data Services: Oferece um conjunto de avançados recursos de gerenciamento de dados no lado do servidor que permitem ao Flex fazer um uso intenso desses dados.

2.3 API Google Maps – O Google Maps foi criado em Fevereiro de 2005 e em Junho de 2005 foi lançada a primeira versão da Google Maps API que vem sendo utilizada até os dias de hoje. A API consiste em um pequeno conjunto de classes em JavaScript que fornecem a interface necessária para que o usuário possa desenvolver suas aplicações exibindo mapas, pesquisando endereços, adicionando marcações no mapa entre outras funções (Azevedo; Carlos, 2008).

Com ajuda desta API no sistema apontará o endereço de clientes, traçará a rota de um determinado serviço a ser prestado no mapa, e a API também retornará a quilometragem desta rota.

Na figura 3, é mostrada uma rota traçada no Google Maps.



Figura 3 - Rota traçada no Google Maps.

2.4 Hibernate - É uma ferramenta de mapeamento objeto/relacional para Java. Ela ajuda a minimizar os códigos de acesso ao banco de dados e SQL utilizando sua própria HQL (Hibernate Query Language), que acelera o desenvolvimento e melhora o desempenho do sistema (Rosa Fragoso; Rodrigo, 2008).

Ele reduz muito o tempo de desenvolvimento de tarefas relacionadas à persistência de dados.

O Hibernate foi desenvolvido por vários programadores e liderado por Gavin King, tempos depois a empresa JBoss Inc contratou diversos desenvolvedores do programa para fazer suporte a ele (Alvez Cruz; Elifrano).

2.5 FireBird – Ele é derivado do código do Borland Interbase 6.0, possui o código fonte aberto e é totalmente gratuito.

Em 1984 Jim Starkey montou uma equipe com Don De Palma e Ann Harrison, e juntos criaram a empresa Groton Database Systems e começa o projeto Interbase, já em 1986 mudaram o nome para Interbase e lançaram a versão 2.0.

Depois de alguns anos, em 2000 a Borland que comercializava o Interbase anuncia que vai o tornar Open Source, mas no dia 25 de Julho de 2000 é lançado o código do Interbase 6, mas sem documentação, ferramentas de teste entre outras coisas, ou seja, ninguém podia alterar apenas sugerir alterações, um grupo de entusiastas então formão o grupo FirebirdTree com um espaço totalmente aberto a todos.

Depois de muitos imprevistos e problemas, em 2002 é lançada a versão 1.0 do Firebird que já nas primeiras semanas o numero de downloads atingiu dezenas de milhares (Trindade Anjos; Artur, 2002).

Para criar, incluir, excluir, atualizar, pesquisar, entre outras funções é utilizado a ferramenta FlameRobin, facilitando a administração do banco de dados.

2.6 BlaseDS - O BlaseDS é uma aplicação Java opensource da Adobe, que fornece serviços para que uma aplicação em Flex conecte-se ao Java, podendo então realizar chamadas de serviços remotos Java.

3. AMBIENTE DE MODELAGEM

A modelagem utilizada para desenvolvimento deste projeto será a UML, que será descrita neste capítulo.

3.1 História da UML

Os estudos sobre a tecnologia de objetos com ênfase em linguagens de programação se iniciaram na década de 1980, logo depois surgiram vários os métodos de análise e projetos, todos voltados para o mesmo conceito. Como todos os métodos eram similares, em 1994 James Rumbaugh e Grady Booch iniciaram o trabalho de unificação de todos os métodos e foi apresentado no ano seguinte (1995).

Em 1995 Ivar Jacobson se juntou a dupla e em 1996 começaram a trabalhar no método chamado Unified Modeling Language (UML) e a Object Management Group (OMG) iniciou o esforço de padronização na área de métodos.

Durante 1996 Rational Software Corporation contou a parceria de varias empresas que resultou na produção da versão 1.0 da UML, e em setembro de 1997 lançaram a versão 1.1 que foi aprovada como padrão pela OMG, com isso a UML acabou com a diferença entre que existam entre os métodos anteriores e unificaram as perspectivas entre os diversos tipos de sistemas, fases de desenvolvimento e conceitos internos (Carlos Macoratti; José).

3.2 Conceitos

Na UML existem quatro conceitos importantes para a modelagem de um sistema que é o diagrama de Caso de Uso, o diagrama de Classes, o diagrama de Estado e o de Sequencia.

Diagrama de Caso de Uso – É formado pelos atores, que são alguém ou alguma coisa que interage com o sistema, seu nome deve refletir com o seu papel que vai desempenhar no sistema.

Formado também pelo Use Case, que são sempre indicados pelo ator, cada Use Case especifica uma funcionalidade completa envolvendo os atores interessados (Carlos; João, 2005).

O ator e o Use Case se comunicam através de associações de comunicação como mostra na figura 4, o Ator Caixa se comunicando com o Use Case Fechar Venda.

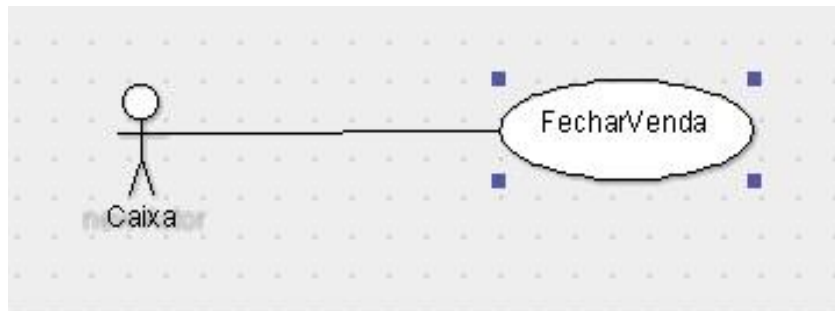


Figura 4 - Ator Caixa associado ao Use Case Fechar Venda.

Diagrama de Classes – Segundo João Carlos (2005) é a descrição de “alguma coisa” que no contexto do sistema modelado possui comportamento (métodos ou operações) e informações a serem armazenadas (atributos).

Existem três tipos de classe que são:

- **Classes de negócios (Informações lógicas)**- Que inclui a funcionalidade lógica e seus objetos utilizam dados que devem estar disponíveis em longo período de tempo, como por exemplo: Cliente, Veículos, Produtos.
- **Classes de Controle (Processos / Algoritmos)** – Que Incluem uma funcionalidade que não podem ser incluídas nos outros tipos de classes, como por exemplo: Cotação, Liquidação.
- **Classes de Interface (Objetos técnicos)** – Que são as classes que interagem com o Usuário, como por exemplo: Janelas, Menus.

A figura 5 mostra um exemplo de uma simples Classe Produto.

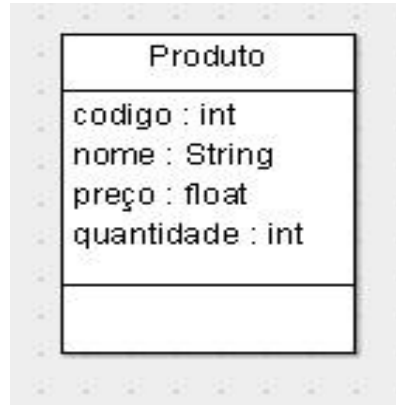


Figura 5 - Exemplo simples de uma classe Produto.

Diagrama de Estado – “Descrevem quais estados um objeto pode alcançar ao longo do seu ciclo de vida, e seu comportamento nestes estados. Exibem quais eventos provocam mudança de estado” (Carlos; João, 2005).

Diagrama de Sequência – “Descrevem como os objetos interagem e se comunicam. Foco é o tempo. Exibem a sequência de mensagens enviadas e recebidas entre um conjunto de objetos na execução de uma função” (Carlos; João, 2005).

4. MODELAGEM DO SOFTWARE

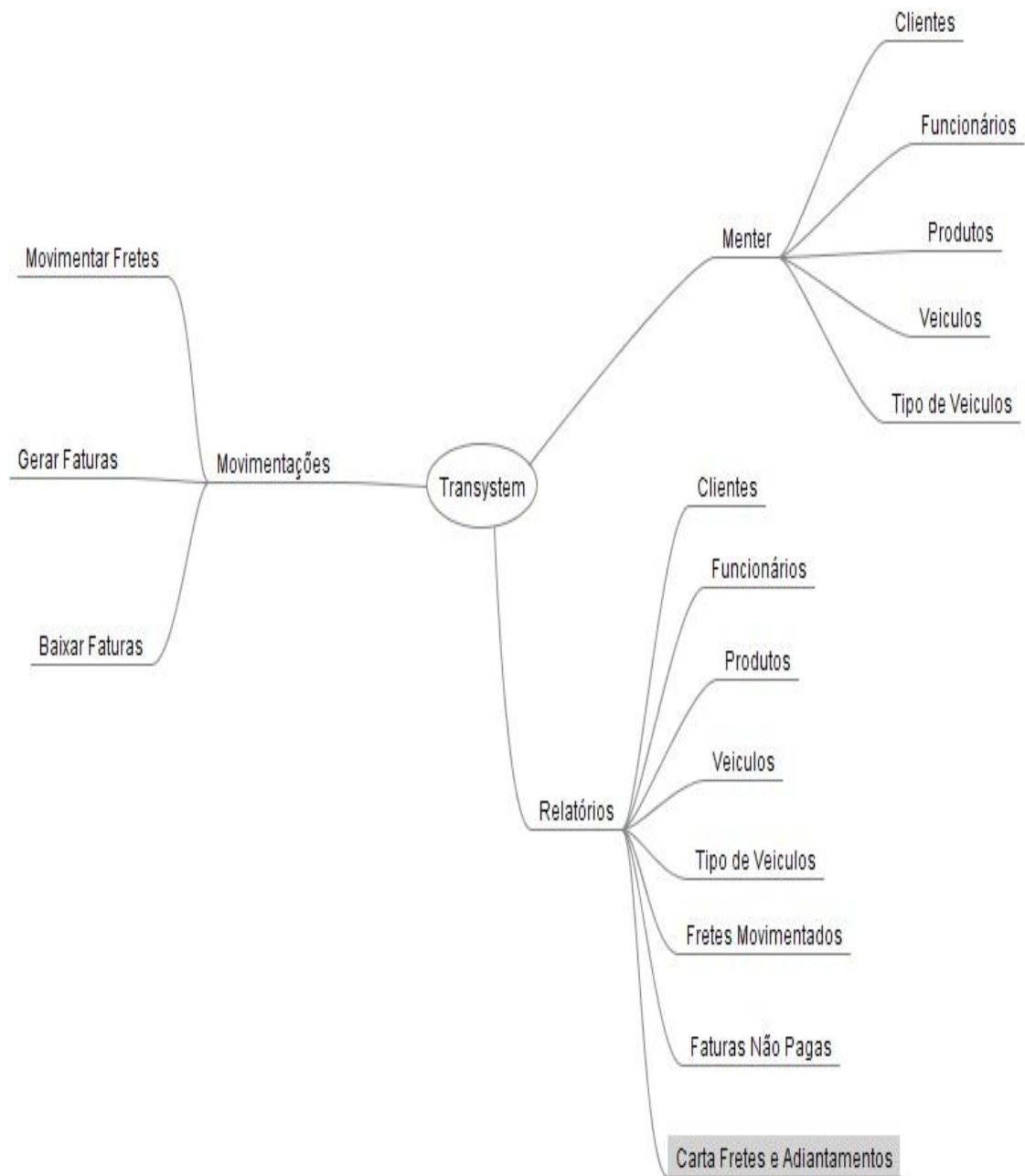


Figura 6 - Mapa mental das funções do sistema.

4.1 DIAGRAMA DE CASO DE USO

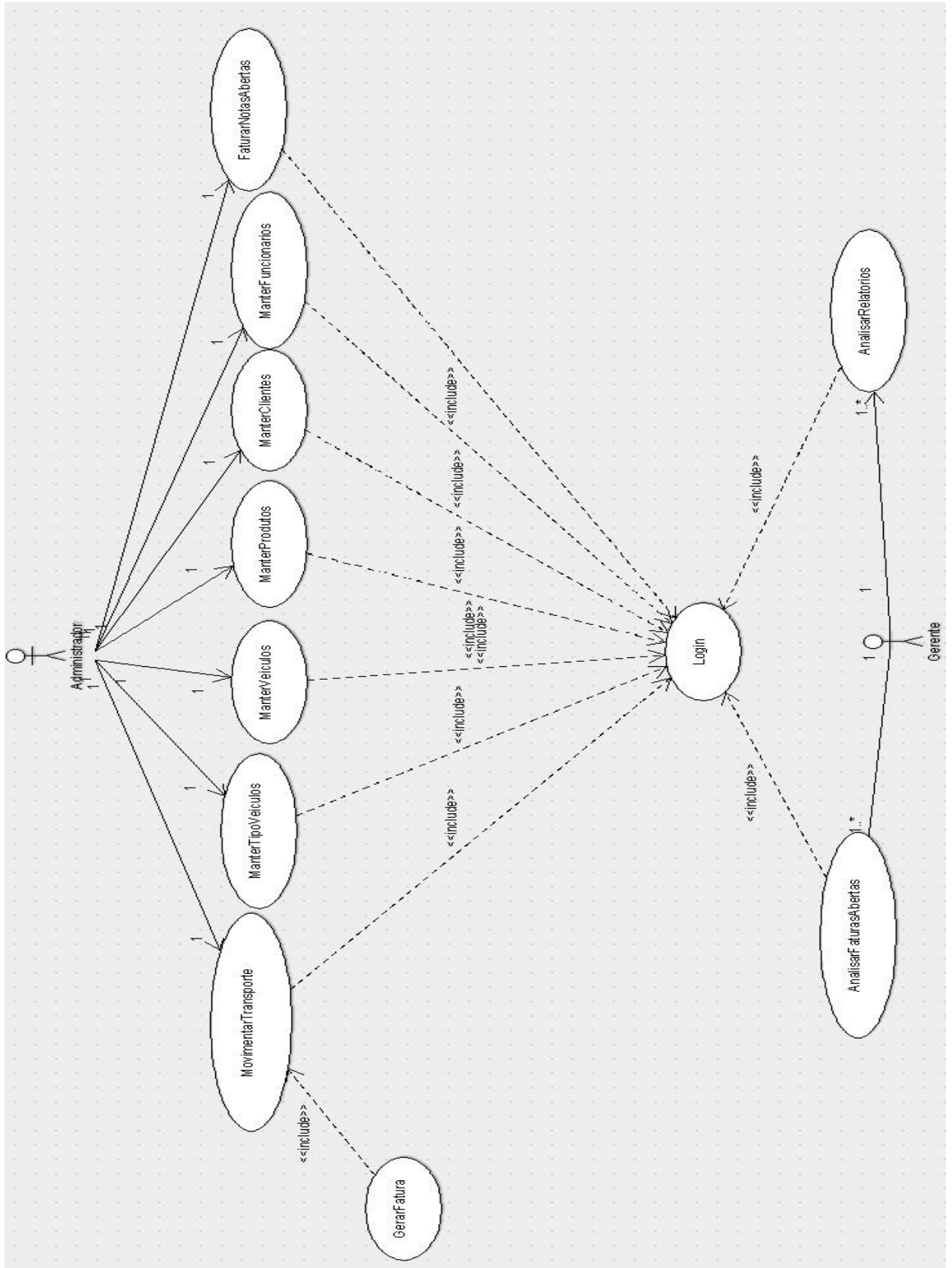


Figura 7 - Diagrama de Caso de Uso completo.

4.1.1 CASOS DE USO

Nome do Caso de Uso: Manter Clientes.

Ator: Administrador.

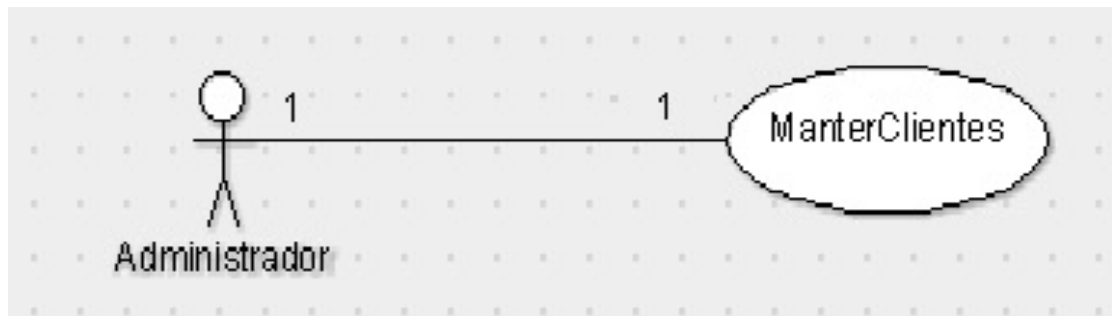


Figura 8 - Caso de Uso Manter Clientes.

O administrador da entrada no sistema e tem as opções de cadastrar, excluir, pesquisar e alterar clientes.

Somente o administrador pode ter acesso a esse tipo de recurso.

Especificação do Caso de Uso.

1. Administrador efetua login no sistema.
2. Sistema abre a pagina de menu com as opções de Cadastrar Clientes **[A1]** e Pesquisar Clientes **[A2]**.

[A1] Cadastrar Clientes.

1. Administrador solicita opção de Cadastrar Clientes.
2. Sistema abre tela de Cadastrar Clientes.
3. Administrador Preenche os campos com os dados do cliente e confirma.
4. Sistema fecha tela de Cadastrar Cliente e mostra mensagem de sucesso.

[A2] Pesquisar Clientes.

1. Administrador solicita opção de Pesquisar Clientes.
2. Sistema vai para pagina de Pesquisar Clientes.
3. Administrador escolhe se deseja pesquisar cliente pelo código, nome, cpf ou cnpj.
4. Administrador preenche o campo cliente de acordo com a opção de busca escolhida e confirma.
5. Sistema lista dados do cliente e aponta seu endereço no mapa do Google Maps.
6. Sistema mostra as opções de Alterar **[A3]** e Excluir **[A4]** ou voltar ao menu principal.

[A3] Alterar.

1. Administrador solicita editar campos para alteração.
2. Administrador preenche os campos que deseja modificar e confirma.
3. Sistema mostra mensagem de sucesso.

[A4] Excluir.

1. Administrador solicita opção de excluir.
2. Sistema apresenta mensagem de sucesso.

Nome do Caso de Uso: Manter Funcionários.

Ator: Administrador.

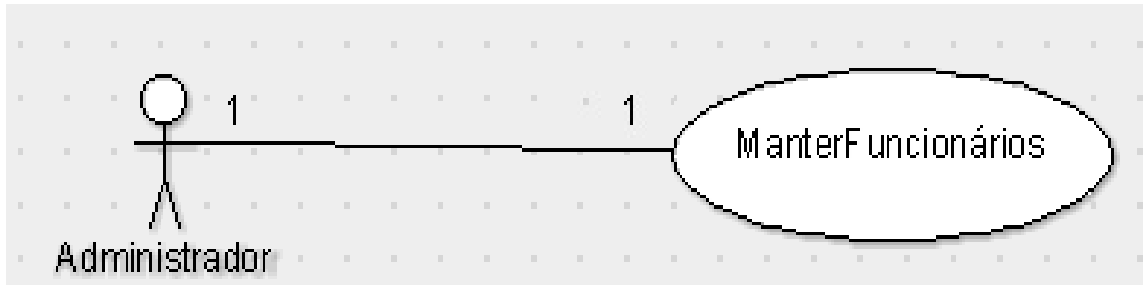


Figura 9 - Caso de Uso Manter Funcionários.

O administrador da entrada no sistema e tem as opções de cadastrar, excluir, pesquisar e alterar funcionários. Nesse cadastro existem as opções de cargo do funcionário, que são: Gerente, Administrador e Motorista, onde cada cargo tem seu nível de acesso do sistema.

Somente o administrador pode ter acesso a esse tipo de recurso.

Especificação do Caso de Uso.

1. Administrador efetua login no sistema.
2. Sistema abre a pagina de menu com as opções de Cadastrar Funcionários **[A1]** e Pesquisar Funcionários **[A2]**.

[A1] Cadastrar Funcionários.

1. Administrador solicita opção de Cadastrar Funcionários.
2. Sistema abre tela de Cadastrar Funcionários.
3. Administrador Preenche os campos com os dados do funcionário e confirma.
4. Sistema fecha tela de Cadastrar Funcionário e mostra mensagem de sucesso.

[A2] Pesquisar Funcionários.

1. Administrador solicita opção de Pesquisar Funcionários.
2. Sistema vai para pagina de Pesquisar Funcionários.
3. Administrador escolhe se deseja pesquisar funcionário pelo código, nome, cpf ou email.
4. Administrador preenche o campo funcionário de acordo com a opção de busca escolhida e confirma.
5. Sistema lista dados do funcionário.
6. Sistema mostra as opções de Alterar **[A3]** e Excluir **[A4]** ou voltar ao menu principal.

[A3] Alterar.

4. Administrador solicita editar campos para alteração.
5. Administrador preenche os campos que deseja modificar e confirma.
6. Sistema mostra mensagem de sucesso.

[A4] Excluir.

3. Administrador solicita opção de excluir.
4. Sistema apresenta mensagem de sucesso.

Nome do Caso de Uso: Manter Produto.

Ator: Administrador.

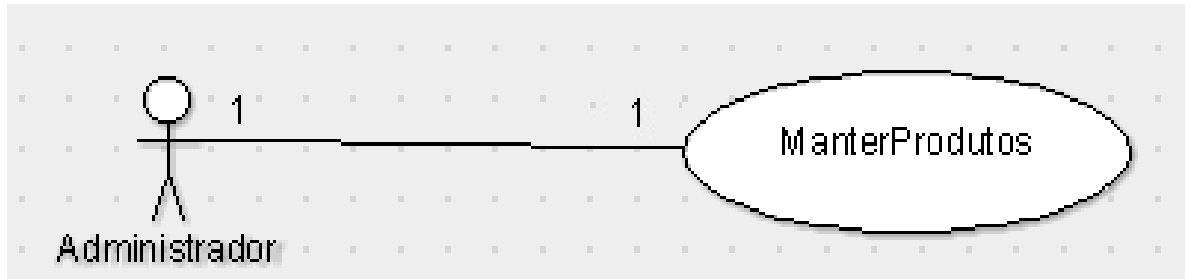


Figura 10 - Caso de Uso Manter Produtos.

O administrador da entrada no sistema e tem as opções de cadastrar, excluir, alterar e pesquisar os produtos.

Somente o administrador pode ter acesso a esse tipo de recurso.

Especificação do Caso de Uso.

1. Administrador efetua login no sistema.
2. Sistema abre a pagina de menu com as opções de Cadastrar Produtos **[A1]** e Pesquisar Produtos **[A2]**.

[A1] Cadastrar Produtos.

1. Administrador solicita opção de Cadastrar Produtos.
2. Sistema abre tela de Cadastrar Produtos.
3. Administrador Preenche os campos com os dados do produto e confirma.
4. Sistema fecha tela de Cadastrar Produtos e mostra mensagem de sucesso.

[A2] Pesquisar Produtos.

1. Administrador solicita opção de Pesquisar Produtos.
2. Sistema vai para pagina de Pesquisar Produtos.
3. Administrador escolhe se deseja pesquisar produto pelo código ou nome.
4. Administrador preenche o campo produto de acordo com a opção de busca escolhida e confirma.
5. Sistema lista dados do produto.
6. Sistema mostra as opções de Alterar **[A3]** e Excluir **[A4]** ou voltar ao menu principal.

[A3] Alterar.

1. Administrador solicita editar campos para alteração.
2. Administrador preenche os campos que deseja modificar e confirma.
3. Sistema mostra mensagem de sucesso.

[A4] Excluir.

1. Administrador solicita opção de excluir.
2. Sistema apresenta mensagem de sucesso.

Nome do Caso de Uso: Manter Tipo Veículos.

Ator: Administrador.

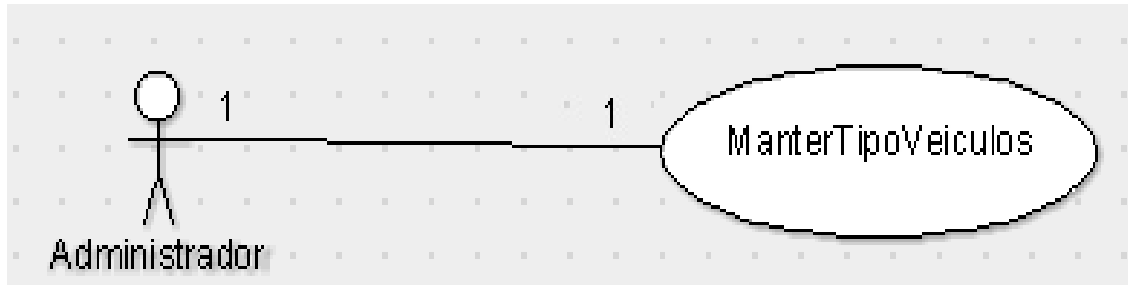


Figura 11 - Caso de Uso Manter Tipo Veículos.

O administrador da entrada no sistema e tem as opções de cadastrar, excluir, alterar e pesquisar os tipos de veículos da empresa.

Somente o administrador pode ter acesso a esse tipo de recurso.

Especificação do Caso de Uso.

1. Administrador efetua login no sistema.
2. Sistema abre a pagina de menu com as opções de Cadastrar Tipo Veículos **[A1]** e Pesquisar Tipo Veículos **[A2]**.

[A1] Cadastrar Tipo Veículos.

1. Administrador solicita opção de Cadastrar Tipo Veículos.
2. Sistema abre tela de Cadastrar Tipo Veículos.
3. Administrador Preenche os campos com os dados do tipo do veiculo e confirma.
4. Sistema fecha tela de Cadastrar Tipo Veículos e mostra mensagem de sucesso.

[A2] Pesquisar Tipo Veículos.

1. Administrador solicita opção de Pesquisar Tipo Veículos.
2. Sistema vai para pagina de Pesquisar Tipo Veículos.
3. Administrador escolhe se deseja pesquisar o tipo do veiculo pelo código ou tipo.
4. Administrador preenche o campo tipo veiculo de acordo com a opção de busca escolhida e confirma.
5. Sistema lista dados do tipo do veiculo.
6. Sistema mostra as opções de Alterar **[A3]** e Excluir **[A4]** ou voltar ao menu principal.

[A3] Alterar.

1. Administrador solicita editar campos para alteração.
2. Administrador preenche os campos que deseja modificar e confirma.
3. Sistema mostra mensagem de sucesso.

[A4] Excluir.

1. Administrador solicita opção de excluir.
2. Sistema apresenta mensagem de sucesso.

Nome do Caso de Uso: Manter Veículos.

Ator: Administrador.

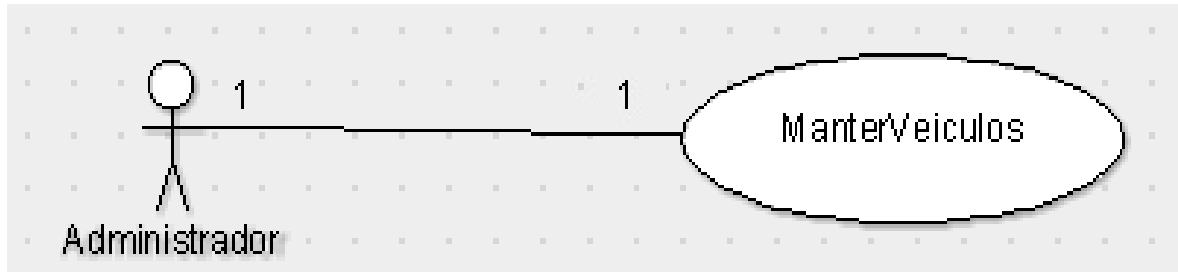


Figura 12 - Caso de Uso Manter Veículos.

O administrador da entrada no sistema e tem as opções de cadastrar, excluir, alterar e pesquisar os veículos.

Somente o administrador terá acesso a esse tipo de recurso.

Especificação do Caso de Uso.

1. Administrador efetua login no sistema.
2. Sistema abre a pagina de menu do sistema com as opções de Cadastrar Veículos **[A1]** e Pesquisar Veículos **[A2]**.

[A1] Cadastrar Veículos.

1. Administrador solicita opção de Cadastrar Veículos.
2. Sistema abre tela de Cadastrar Veículos.
3. Administrador Preenche os campos com os dados do veiculo e confirma.
4. Sistema fecha tela de Cadastrar Veículos e mostra mensagem de sucesso.

[A2] Pesquisar Veículos.

1. Administrador solicita opção de Pesquisar Veículos.
2. Sistema vai para pagina de Pesquisar Veículos.
3. Administrador seleciona a busca pelo código do veiculo.
4. Administrador preenche o campo veiculo e confirma.
5. Sistema lista dados do veiculo.
6. Sistema mostra as opções de Alterar **[A3]** e Excluir **[A4]** ou voltar ao menu principal.

[A3] Alterar.

1. Administrador solicita editar campos para alteração.
2. Administrador preenche os campos que deseja modificar e confirma.
3. Sistema mostra mensagem de sucesso.

[A4] Excluir.

1. Administrador solicita opção de excluir.
2. Sistema apresenta mensagem de sucesso.

Nome do Caso de Uso: Movimentar Transporte e Gerar Fatura.

Ator: Administrador.

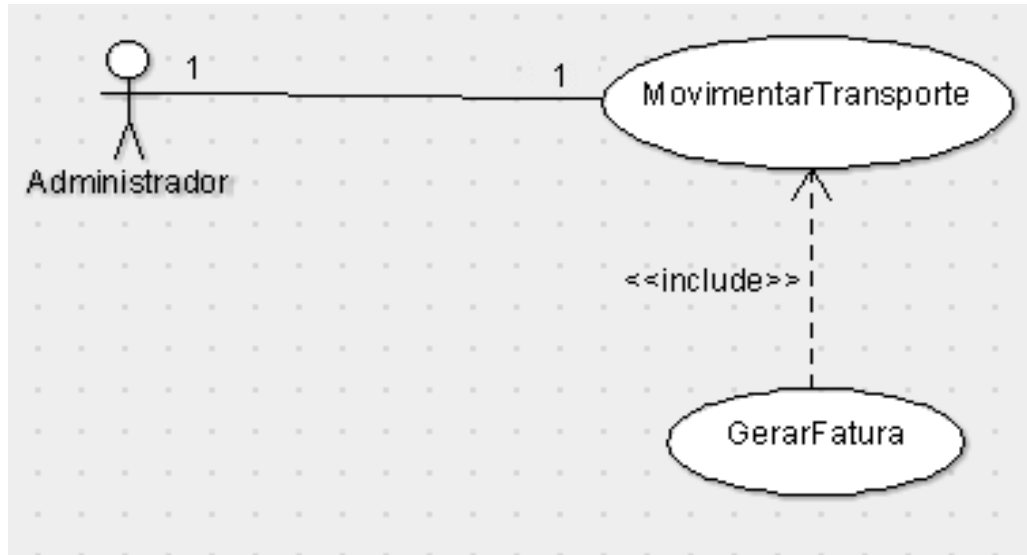


Figura 13 - Caso de Uso Movimentar Transporte e Gerar Fatura.

O administrador da entrada no sistema e tem a opção de movimentar transporte.

Nesse recurso o administrador movimenta o serviço a ser prestado para um cliente.

Ele informa ao sistema dados do cliente contratante, dados do cliente destinatário, o produto a ser transportado, o peso, o motorista que fara o transporte, o veiculo, e também o preço do quilometro por tonelada a ser transportado, o sistema então com a ajuda da API Google Maps calcula a distancia em quilômetros entre a origem e o destino com base nos dados dos clientes e calcula o preço do frete. O preço do frete é calculado da seguinte forma, $\text{peso} * \text{distancia} * \text{preço do Km/t}$.

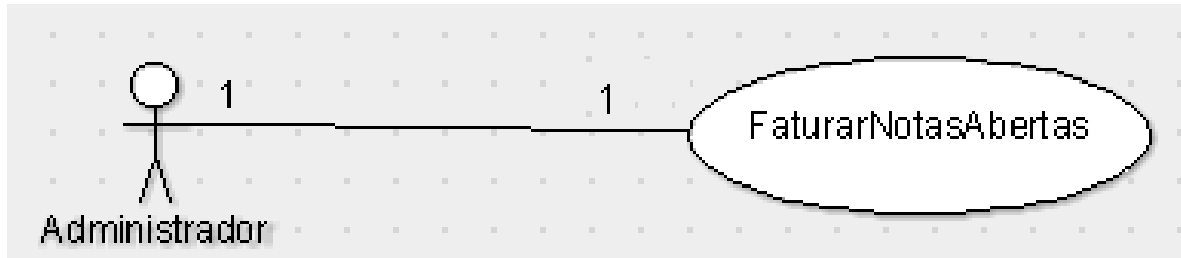
Depois que o administrador conclui o transporte, o sistema gera uma fatura com os dados do transporte, e uma data de vencimento.

Somente o administrador terá acesso a esse tipo de recurso.

Especificação do Caso de Uso.

1. Administrador efetua login no sistema.
2. Sistema vai para pagina de menu e mostra a opção de Movimentar Frete.

3. Administrador solicita a opção de Movimentar Frete.
4. Sistema vai para pagina de movimentar o frete com os dados iniciais do frete a serem preenchidos.
5. Administrador preenche os dados iniciais do frete e muda para a aba de dados de destino.
6. Administrador preenche os dados de destino do frete e muda para a aba dados finais.
7. Administrador preenche os campos com os valores do quilometro, combustível, pedágio e adicionais.
8. Sistema traça a rota do frete no Google Maps e calcula o valor.
9. Administrador muda para a aba de carta frete e adiantamento, preenche os campos com os valores e confirma.
10. Sistema gera a fatura do frete e cadastra.
11. Sistema imprime os documentos de Frete, Carta Frete, Adiantamento, Fatura e finaliza.

Nome do Caso de Uso: Faturar Notas Abertas.**Ator: Administrador.****Figura 14 - Caso de Uso Faturar Notas Abertas.**

O administrador da entrada no sistema e tem a opção de faturar as notas de transporte que ainda não foram pagas se caso o cliente contratante pagar. Enquanto o cliente não pagar a futura ficara em aberto.

Somente o administrador terá acesso a esse tipo de recurso.

Especificação do Caso de Uso.

1. Administrador efetua login no sistema.
2. Sistema vai para pagina de menu e mostra a opção de Baixar Faturas.
3. Administrador solicita a opção de Baixar Faturas.
4. Sistema vai para pagina de Baixar Faturas.
5. Administrador busca a fatura pelo código.
6. Sistema lista os dados da fatura.
7. Administrador confirma a baixa da fatura.
8. Sistema baixa a fatura do sistema.

Nome do Caso de Uso: Analisar Faturas Abertas.

Ator: Gerente.

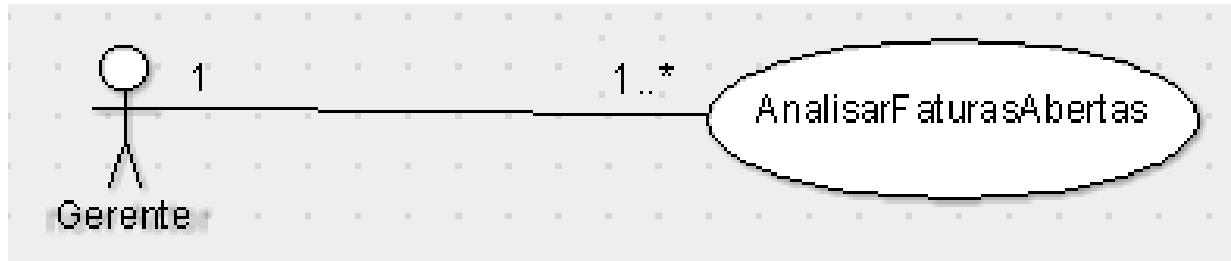


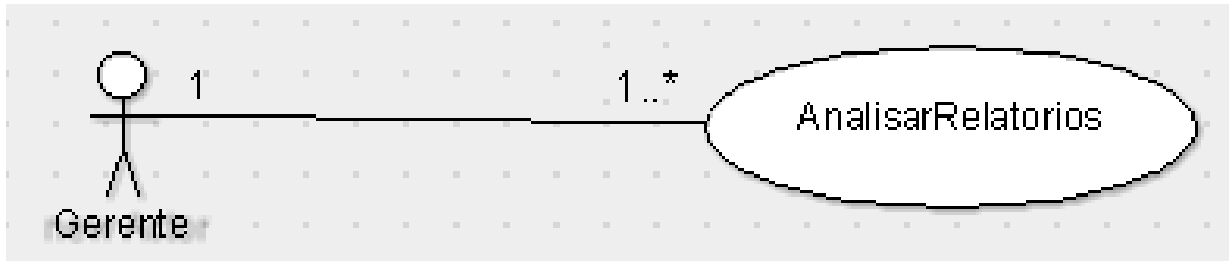
Figura 15 - Caso de Uso Analisar Faturas Abertas.

O gerente da entrada no sistema e tem a opção de analisar faturas que ainda não foram pagas pelos clientes. Essas faturas são geradas pela opção movimentar transporte.

Só o gerente terá acesso a esse tipo de recurso.

Especificação do Caso de Uso.

1. Gerente efetua login no sistema.
2. Sistema vai para pagina de menu e mostra a opção de Analisar Faturas Abertas.
3. Gerente solicita a opção Analisar Faturas Abertas.
4. Sistema vai para pagina de Analisar Faturas Abertas.
5. Sistema lista as faturas que ainda não foram pagas e baixadas do sistema.

Nome do Caso de Uso: Analisar Relatórios.**Ator: Gerente.****Figura 16 - Caso de Uso Analisar Relatórios.**

O gerente da entrada no sistema e pode além de analisar ou imprimir faturas não pagas, ele pode analisar relatórios de cliente, funcionários, produtos, veículos, tipo de veículos, fretes movimentados, faturas não pagas, carta fretes e adiantamentos.

Especificação do Caso de Uso.

1. Gerente efetua login no sistema.
2. Sistema vai para pagina de menu e mostra a opção de Relatórios de clientes, funcionários, produtos, veículos, tipos de veículos, fretes movimentados, faturas não pagas, carta fretes e adiantamentos.
3. Gerente solicita uma das opções.
4. Sistema faz a lista dos dados que o gerente solicitou.
5. Gerente solicita impressão dos documentos.
6. Sistema efetua a impressão.

4.2 DIAGRAMA DE CLASSE

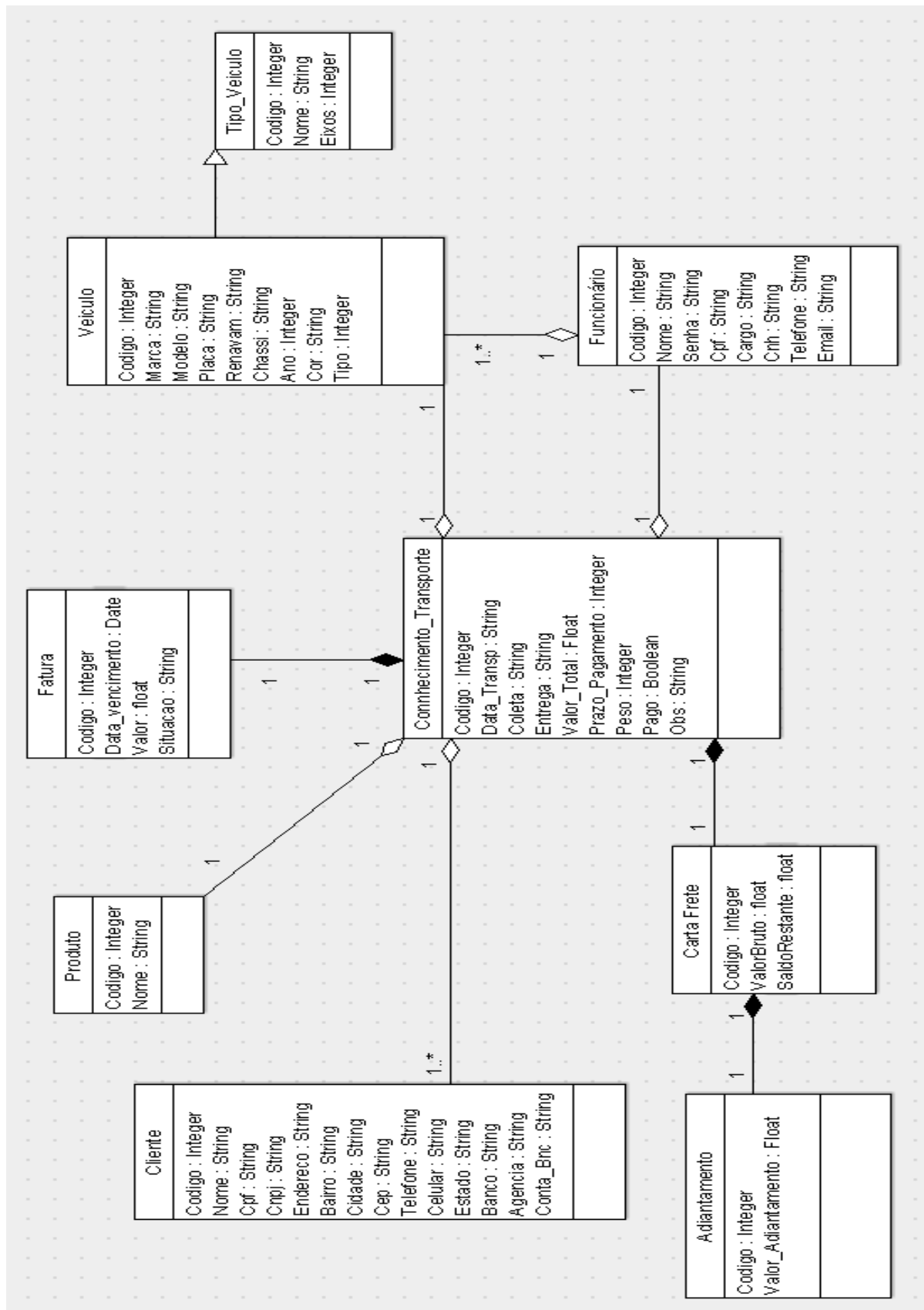


Figura 17 – Diagrama de classe do projeto.

4.3 MODELO ENTIDADE RELACIONAMENTO

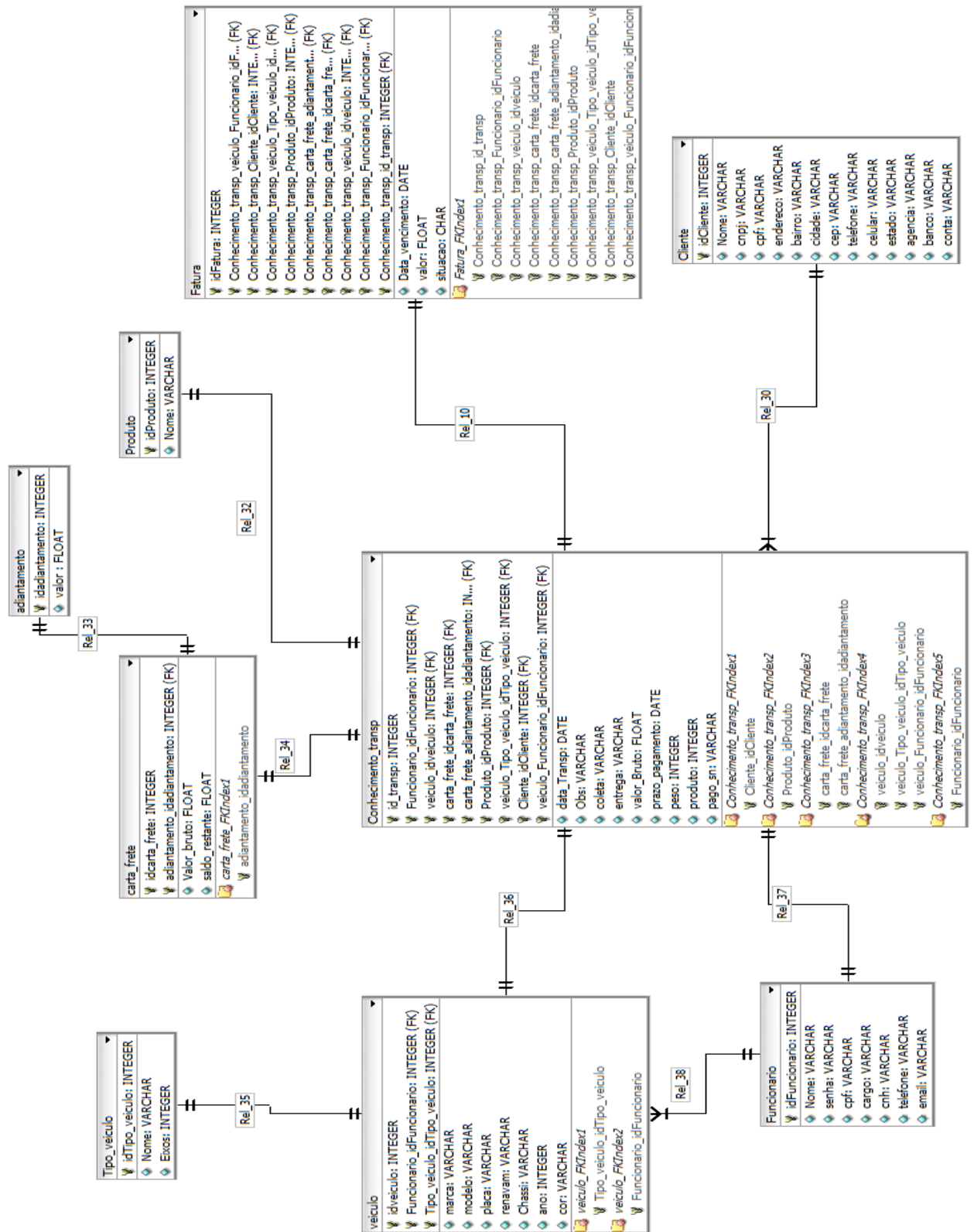


Figura 18 – Modelo Entidade Relacionamento.

4.4 DICIONARIO DE DADOS

| Entidade: Adiantamento | | | | | |
|-------------------------------|-----------------------|-------------|-----------|-----------|--|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| idAdiantamento | INTEGER | NOT NULL | X | | Código de identificação do adiantamento. |
| Valor | FLOAT | NULL | | | Valor que foi adiantado do carta frete ao motorista. |

Tabela 2 – Dicionário de dados da tabela Adiantamento.

| Entidade: Carta_frete | | | | | |
|---------------------------------|-----------------------|-------------|-----------|-----------|---|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| Adiantamento_id Adiantamento | INTEGER | NULL | | X | Código de identificação da tabela Adiantamento. |
| idCarta_frete | INTEGER | NOT NULL | X | | Código de identificação da carta frete. |
| Saldo_restante | FLOAT | NULL | | | Valor restante da carta frete devido ao adiantamento. |
| Valor_bruto | FLOAT | NULL | | | Valor bruto da carta frete. |

Tabela 3 - Dicionário de dados da tabela Carta_frete.

| Relacionamentos | | |
|------------------------|-----------------------------|-------------------------|
| Entidade | Campo origem | Campo de destino |
| Carta_frete | Adiantamento_idAdiantamento | <u>IdAdiantamento</u> |

Tabela 4 - Relacionamentos da tabela Carta_frete.

| Entidade: Conhecimento_transp | | | | | |
|--------------------------------------|-----------------------|-------------|-----------|-----------|--|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| Carta_frete_idCarta_frete | INTEGER | NULL | | X | Código de identificação da tabela Carta_frete. |
| Cliente_idCliente | INTEGER | NULL | | X | Código de identificação da tabela Cliente. |
| Coleta | VARCHAR (50) | NULL | | | Cidade onde será feita coleta do produto a ser transportado. |
| Data_transp | DATE | NULL | | | Data que vai ser feito o transporte. |
| Entrega | VARCHAR (50) | NULL | | | Cidade onde será feita entrega do produto transportado. |
| Funcionario_idFuncionario | INTEGER | NULL | | X | Código de identificação da tabela Funcionário. |
| idTransp | INTEGER | NOT NULL | X | | Código de identificação do conhecimento de transporte |
| Obs | VARCHAR (150) | NULL | | | Observações que podem ser adicionada sobre o transporte. |
| Pago_SN | VARCHAR (3) | NULL | | | Informa se transporte já foi pago (Sim ou Não). |
| Peso | FLOAT | NULL | | | Peso total da carga que será transportada. |
| Prazo_pagamento | DATE | NULL | | | Prazo para o cliente efetuar o pagamento depois do serviço prestado. |
| Produto_idProduto | INTEGER | NULL | | X | Código de identificação da tabela Produto. |
| Valor_bruto | FLOAT | NULL | | | Valor bruto do transporte. |
| Valor_liquido | FLOAT | NULL | | | Valor do transporte já com descontos de pedágios e etc. |

| | | | | | |
|-------------------|---------|------|--|---|--|
| Veiculo_idVeiculo | INTEGER | NULL | | X | Código de identificação da tabela Veiculo. |
|-------------------|---------|------|--|---|--|

Tabela 5 - Dicionário de dados da tabela Conhecimento_trans.

| Relacionamentos | | |
|---------------------|---------------------------|----------------------|
| Entidade | Campo origem | Campo de destino |
| Conhecimento_transp | Cliente_idCliente | <u>idCliente</u> |
| Conhecimento_transp | Carta_frete_idCarta_frete | <u>idCarta_frete</u> |
| Conhecimento_transp | Funcionário_idFuncionario | <u>idFuncionario</u> |
| Conhecimento_transp | Produto_idProduto | <u>idProduto</u> |
| Conhecimento_transp | Veiculo_idVeiculo | <u>idVeiculo</u> |

Tabela 6 - Relacionamentos da tabela Conhecimento_transp.

| Entidade: Cliente | | | | | |
|-------------------|----------------|----------|----|----|-------------------------------------|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| Agencia | VARCHAR (10) | NULL | | | Agencia bancaria do cliente. |
| Bairro | VARCHAR (50) | NULL | | | Bairro do cliente. |
| Banco | VARCHAR (30) | NULL | | | Nome do banco do cliente. |
| Cidade | VARCHAR (50) | NULL | | | Cidade do cliente. |
| CNPJ | VARCHAR(18) | NULL | | | CNPJ do cliente. |
| Conta | VARCHAR (15) | NULL | | | Numero de conta do cliente. |
| CPF | VARCHAR (14) | NULL | | | CPF do cliente. |
| Endereço | VARCHAR (150) | NULL | | | Endereço do cliente. |
| Estado | VARCHAR (2) | NULL | | | Sigla do estado do cliente. |
| idCliente | INTEGER | NOT NULL | X | | Código de identificação do Cliente. |
| Nome | VARCHAR (150) | NULL | | | Nome do cliente. |
| Telefone | VARCHAR (14) | NULL | | | Telefone do cliente. |

Tabela 7 - Dicionário de dados da tabela Cliente.

| Entidade: Fatura | | | | | |
|-------------------------------|-----------------------|-------------|-----------|-----------|--|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| Conhecimento_transp_id_transp | INTEGER | NULL | | X | Código de identificação da tabela Conhecimento_transp. |
| Data_vencimento | DATE | NULL | | | Data de vencimento da fatura. |
| idFatura | INTEGER | NOT NULL | X | | Código de identificação da Fatura. |
| Situação | VARCHAR (3) | NULL | | | Se a fatura foi paga (Sim ou Não). |
| Valor | FLOAT | NULL | | | Valor da fatura. |

Tabela 8 - Dicionário de dados da tabela Fatura.

| Relacionamentos | | |
|------------------------|-------------------------------|-------------------------|
| Entidade | Campo origem | Campo de destino |
| Conhecimento_transp | Conhecimento_transp_id_transp | <u>id_transp</u> |

Tabela 9 - Relacionamentos da tabela Fatura.

| Entidade: Funcionário | | | | | |
|------------------------------|-----------------------|-------------|-----------|-----------|---|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| Cargo | VARCHAR (10) | NULL | | | Cargo do funcionário. |
| Email | VARCHAR (60) | NULL | | | Email de contato do funcionário. |
| idFuncionario | INTEGER | NOT NULL | X | | Código de identificação do funcionário. |
| Nome | VARCHAR (150) | NOT NULL | | | Nome do funcionário. |
| Senha | VARCHAR (14) | NOT NULL | | | Senha de login do funcionário. |

Tabela 10 - Dicionário de dados da tabela Funcionário.

| Entidade: Produto | | | | | |
|--------------------------|-----------------------|-------------|-----------|-----------|-------------------------------------|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| idProduto | INTEGER | NOT NULL | X | | Código de identificação do produto. |
| Nome | VARCHAR (100) | NULL | | | Nome do produto. |

Tabela 11 - Dicionário de dados da tabela Produto.

| Entidade: Tipo_veiculo | | | | | |
|-------------------------------|-----------------------|-------------|-----------|-----------|---|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| Eixos | INTEGER | NULL | | | Quantidade de eixos. |
| IdTipo_veiculo | INTEGER | NOT NULL | X | | Código de identificação do tipo do veiculo. |
| Nome | VARCHAR (150) | NULL | | | Nome do tipo do veiculo. |

Tabela 12 - Dicionário de dados da tabela Tipo_veiculo.

| Entidade: Veiculo | | | | | |
|-----------------------------|-----------------------|-------------|-----------|-----------|---|
| Campo | Tipos de dados | Nulo | PK | FK | Descrição |
| Ano | INTEGER | NULL | | | Ano de fabricação do veiculo. |
| Chassi | VARCHAR (25) | NULL | | | Código do chassi do veiculo. |
| Cor | VARCHAR (20) | NULL | | | Cor do veiculo. |
| Funcionario_idFuncionario | INTEGER | NULL | | X | Código de identificação do funcionário. |
| idVeiculo | INTEGER | NOT NULL | X | | Código de identificação do veiculo. |
| Marca | VARCHAR (50) | NULL | | | Marca do veiculo. |
| Modelo | VARCHAR (50) | NULL | | | Modelo do veiculo. |
| Placa | VARCHAR (8) | NULL | | | Placa do veiculo. |
| Renavam | VARCHAR (18) | NULL | | | Renavam do veiculo. |
| Tipo_veiculo_idTipo_veiculo | INTEGER | NULL | | X | Código de identificação do tipo do veiculo. |

Tabela 13 - Dicionário de dados da tabela Veiculo.

| Relacionamentos | | |
|------------------------|-----------------------------|-------------------------|
| Entidade | Campo origem | Campo de destino |
| Veiculo | Tipo_veiculo_idTipo_veiculo | idTipo_veiculo |
| Veiculo | Funcionario_idFuncionario | idFuncionario |

Tabela 14 - Relacionamentos da tabela Veiculo.

5. CONCLUSÃO

5.1 Considerações finais

Este trabalho de conclusão de curso foi motivado pelo interesse em sistemas webs que vem crescendo cada vez mais e vem se tornando cada vez mais acessível para as pessoas, e também para mostrar que a ferramenta API Google Maps é utilizada para manipular mapas de forma muito eficaz.

Um dos grandes desafios foi aprender como é o funcionamento do Adobe Flex, com seus componentes e funções, e principalmente como é feita a conexão entre o Adobe Flex e o Eclipse, no qual envolvia outra ferramenta chamada BlaseDS, que fazia as conexões em xml das classes do Eclipse no Flex.

Outro desafio foi aprender como fazer o Google Maps trabalhar com a aplicação no Flex, e também como conseguir traçar as rotas no mapa e pegar a quilometragem desta rota para depois calcular o valor do frete.

5.2 Trabalhos futuros

Uma sugestão para a extensão deste trabalho pode-se estudar mais sobre a API Google Maps e conseguir retirar também valores de pedágios em determinadas rotas que forem traçadas ou colher outras informações que possam ser uteis.

Outra sugestão é aprimorar o uso da API Google Maps para funcionar como um GPS e o usuário poderá saber em que posição do mapa seus motoristas estão, se já terminaram o frete ou se ainda estão em viagem.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, Eduardo Soares. Detecção de padrões de código Java na plataforma Eclipse. 2005. 88p. Dissertação (mestrado). Instituto Militar de Engenharia. Rio de Janeiro. 2005.

ANJOS, Arthur Trindade. Firebird e Interbase, o passado, o presente e o futuro. Disponível em: <http://www.arsoft.pt/artur/firebird/files/cl_001.pdf>. Acesso em: 20. Mai. 2011.

AZEVEDO, Carlos. Google Maps API. Disponível em: <<http://fabio-martins.com/artigos-ti/27-artigos-var-ti/80-flex.html>>. Acesso em: 20. Mai. 2011.

CARLOS, João. Modelagem Orientada a Objetos e UML. Disponível em: <http://imasters.com.br/artigo/2934/uml/modelagem_orientada_a_objetos_e_uml/>. Acesso em: 04. Jun. 2011.

CÉSAR DE CASTRO, Wellington. O que é ADOBE FLEX? Disponível em: <http://www.oficinadanet.com.br/artigo/1482/o_que_e_adobe_flex>. Acesso em: 17. Out. 2011.

CRUZ, Elifranio Alves. Hibernate conceitos. Disponível em: <<http://www.slideshare.net/elifranio/hibernate-conceitos>>. Acesso em: 01. Jun. 2011.

FRAGOSO, Rodrigo Rosa. O que é Hibernate. Disponível em: <http://www.dicas-l.com.br/arquivo/o_que_e_hibernate.php>. Acesso em: 01. Jun. 2011.

GOOGLE. Guia do desenvolvedor Google Maps API for Flash. Disponível em: <<http://code.google.com/intl/pt-BR/apis/maps/documentation/flash/>>. Acesso em: 14. Mar. 2011.

HEMRAJANI, Arial. Desenvolvimento ágil em Java com Spring, Hibernate e Eclipse. 290p. 2006.

MACORRATI, José Carlos. UML – Conceitos Básicos II. Disponível em: <http://www.macoratti.net/vb_uml2.htm>. Acesso em: 03. Jun. 2011.

MARTINS, Fabio. Adobe Flex. Disponível em: <<http://fabio-martins.com/artigos-ti/27-artigos-var-ti/80-flex.html>>. Acesso em: 17. Out. 2011.

WILLIAMS, Mike. Google Maps API Tutorial. Disponível em: <<http://econym.org.uk/gmap/>>. Acesso em: 14. Mar. 2011.