

UNIVERSIDADE DO OESTE DE SANTA CATARINA – UNOESC

CURSO DE SISTEMAS DE INFORMAÇÃO

RAPHAEL SILVEIRA

**MONITORAMENTO DE INTEGRAÇÃO DE DADOS DE TECNOLOGIAS DE
RASTREAMENTO**

CHAPECÓ (SC)

2010

RAPHAEL SILVEIRA

**MONITORAMENTO DE INTEGRAÇÃO DE DADOS DE TECNOLOGIAS DE
RASTREAMENTO**

**Trabalho de Conclusão de Curso apresentado
ao Curso de Sistemas de Informação, Área
das Ciências Exatas e da Terra, da
Universidade do Oeste de Santa Catarina
como requisito à obtenção do título de
Bacharel em Sistemas de Informação.**

Orientador: Prof.º Rafael Leite

CHAPECÓ (SC)

2010

RAPHAEL SILVEIRA

**MONITORAMENTO DE INTEGRAÇÃO DE DADOS DE TECNOLOGIAS DE
RASTREAMENTO**

**Trabalho de Conclusão de Curso apresentado
ao Curso de Sistemas de Informação, Área das
Ciências Exatas e da Terra, da Universidade do
Oeste de Santa Catarina como requisito à
obtenção do título de Bacharel em Sistemas de
Informação.**

Aprovada em

BANCA EXAMINADORA

Profº Rafael Leite, Esp.
Universidade do Oeste de Santa Catarina
Nota Atribuída

Profº Raquel Aparecida Pegoraro, M.Sc.
Universidade do Oeste de Santa Catarina
Nota Atribuída

Profº Tiago Zonta, M.Sc.
Universidade do Oeste de Santa Catarina
Nota Atribuída

RESUMO

O presente estudo objetiva a efetuação de monitoramento de integração de dados entre uma empresa de gerenciamento de risco no transporte rodoviário e empresas de tecnologia de rastreamento de veículos. O que motiva a pesquisa é o aumento exponencial na procura dos serviços oferecidos pela gerenciadora de risco e por consequência o aumento no fluxo dos dados integrados. A gerenciadora reúne em um único aplicativo o processo de integração de dados provenientes de cinco diferentes tecnologias que repassam seus dados de maneiras distintas. O *software* desenvolvido aqui irá monitorar as integrações realizadas pelo aplicativo que a gerenciadora utiliza auxiliando na visualização e interpretação individual dos processos realizados na integração de dados, permitindo que ações preventivas e corretivas ocorram em um intervalo de tempo menor, de maneira planejada e padronizada, através de armazenamento de informações, gerando um histórico de problemas encontrados no processo de integração e da atuação da equipe de suporte nestes problemas.

Palavras chave: *software*, monitoramento, integração.

ABSTRACT

The following study objectives the accomplishment in monitoring data integration between a risk management in cargo transportation and the companies that provide technology in vehicle tracking. What motives the research is the exponential increase in the search for the services offered by the risk management company and consequentially an increase in the integrated data flux. The risk management company puts in one single applicative the integration data process of five different technologies that offer data in distinct ways. The software here developed will monitor the integrations made by the applicative used by the risk management company, helping the single visualizing and single interpretation on the data processes made, allowing preventive and corrective actions to occur in a shorter period of time, in an organized and planed way, through information storage, generating a history list of the problems found in the integration and the support team actions toward these problems.

Key words: software, monitoring, integration.

LISTA DE FIGURAS

Figura 1: Layout de recebimento de evento da tecnologia Omnilink	17
Figura 2: Layout de requisição da tecnologia Rodosis	19
Figura 3: Layout de retorno da tecnologia Rodosis	19
Figura 4: Layout de requisição da tecnologia OnixSat	21
Figura 5: Layout de retorno da tecnologia OnixSat.....	21
Figura 6: Software de integração em atividade.	35
Figura 7: Processo de integração visão geral.....	36
Figura 8: Processo de integração tecnologia Omnilink.	37
Figura 9: Processo de integração tecnologia OnixSat.	38
Figura 10: Processo de integração tecnologia Rodosis.....	38
Figura 11: Processo de integração tecnologia ControlLoc.	39
Figura 12: Processo de integração tecnologia Sascar.	39
Figura 13: Software verificador de integrações em atividade.	41
Figura 14: Diagrama de casos de uso.	47
Figura 15: Diagrama de Classes do Sistema.....	59
Figura 16: Diagrama do Banco de dados.....	60
Figura 17: Diagrama de Seqüência Efetuar Login.	61
Figura 18: Diagrama de Seqüência Manter Usuário.....	61
Figura 19: Diagrama de Seqüência Manter Tecnologia.....	62
Figura 20: Diagrama de Seqüência Manter Exceções.....	62
Figura 21: Diagrama de Seqüência Parametrizar Integração.....	63
Figura 22: Diagrama de Seqüência Ativar Monitoramento.....	63
Figura 23: Diagrama de Seqüência Desativar Monitoramento.....	64
Figura 24: Diagrama de Seqüência Gráficos de Desempenho.....	64
Figura 25: Diagrama de Seqüência Situação das Integrações.....	65
Figura 26: Diagrama de Seqüência Verificar Alertas Novos.....	66
Figura 27: Diagrama de Seqüência Verificar Alertas em Tratamento.....	67
Figura 28: Diagrama de Seqüência Verificar Alertas Estabilizados.....	68
Figura 29: Diagrama de Seqüência Gerar Relatórios de Alertas.....	69
Figura 30: Tela de Login.....	69
Figura 31: Tela Principal.....	70
Figura 32: Tela de Cadastro de Usuários.....	70
Figura 33: Tela de Cadastro de Tecnologia.....	70
Figura 34: Tela de Cadastro de Parâmetros.....	71
Figura 35: Tela de Cadastro de Exceções.....	71
Figura 36: Tela de Tecnologias Monitoradas.....	72
Figura 37: Tela de Monitoramento.....	72
Figura 38: Tela de Alertas em Tratamento.....	73
Figura 39: Tela de Alertas Estabilizados.....	73
Figura 40: Tela de Gráfico de Registros.....	74
Figura 41: Tela de Gráfico de Velocidade.....	74
Figura 42: Tela de Relatório Gráfico de Alertas.....	75
Figura 43: Tela de Relatório Gráfico de Alertas por Tecnologia.....	75
Figura 44: Tela de Relatório Gráfico de Alertas Verificados x Não Verificados.....	76
Figura 45: Tela de Relatório de Alertas não Verificados.....	76

Figura 46: Tela de Relatório de Alertas Verificados. 77
Figura 47: Tela de Detalhes do Alerta..... 77

LISTA DE TABELAS

Tabela 1: Identificação de tags do documento XML da tecnologia Omnilink	18
Tabela 2: Identificação de Tags do documento XML de requisição da tecnologia Rodosis.....	19
Tabela 3: Identificação de Tags do documento XML de retorno da tecnologia Rodosis	20
Tabela 4: Identificação de tags do documento XML de requisição da tecnologia OnixSat.....	21
Tabela 5: Identificação de tags do documento de retorno da tecnologia OnixSat.....	22
Tabela 6: Caso de Uso Efetuar Login.	48
Tabela 7: Caso de Uso Manter Usuário.....	49
Tabela 8: Caso de Uso Manter Tecnologias	50
Tabela 9: Caso de Uso Manter Exceções	51
Tabela 10: Caso de Uso Parametrizar Integração	52
Tabela 11: Caso de Uso Ativar Monitoramento.....	53
Tabela 12: Caso de Uso Desativar Monitoramento	53
Tabela 13: Caso de Uso Visualizar Situação das Integrações	54
Tabela 14: Caso de Uso Visualizar Gráficos de Desempenho	55
Tabela 15: Caso de Uso Verificar e Gerenciar Alertas.....	56
Tabela 16: Caso de Uso Tratar Alertas.....	57
Tabela 17: Caso de Uso Finalizar Alertas	57
Tabela 18: Caso de Uso Gerar Relatórios de Alertas	58

SUMÁRIO

1 INTRODUÇÃO.....	10
1.1 HISTÓRICO	11
1.2 OBJETIVOS GERAIS.....	11
1.3 OBJETIVOS ESPECÍFICOS	12
2 FUNDAMENTAÇÃO TEÓRICA.....	12
2.1 GERENCIAMENTO DE RISCO NO TRANSPORTE DE CARGAS	13
2.1.1 Gerenciamento de Risco e Sistemas de Informação.....	13
2.2 SISTEMAS INTEGRADOS DE RASTREAMENTO.....	14
2.3 RECEBIMENTOS DE DADOS	15
2.3.1 Tecnologia Sascar®	16
2.3.2 Tecnologia ControlLoc®	16
2.3.3 Tecnologia Omnilink®	16
2.3.4 Tecnologia Rodosis®	18
2.3.5 Tecnologia OnixSat®	20
2.4 XML (Extensible Modeling Language).....	22
2.4.1 Objetivos básicos da Xml	23
2.4.2 Tipos de Aplicações	24
2.4.3 Composição de um documento XML	25
2.5 WEB SERVICES	26
2.5.1 Tecnologias Utilizadas na Criação de Web Services	26
2.5.1.1 Soap	27
2.5.1.2 Wsdl.....	27
2.5.1.3 Uddi.....	28
2.6 THREADS	28
2.6.1 O Ciclo de Vida dos Threads	29
2.6.2 Gerenciamento dos Threads	30
2.6.3 Multithreading	30
2.6.4 Threads no Delphi®	31
2.7 ENGENHARIA DE SOFTWARE.....	32
2.7.1 Requisitos	32
2.7.1.1 Requisitos Funcionais e não-funcionais	33
2.7.2 Projeto do Software com UML.....	33
3 ANÁLISE DO SISTEMA	34
3.1 ESTUDO DO SISTEMA EXISTENTE.....	34
3.1.1 Integração de Dados	34
3.1.1.1 Integração Omnilink®	36
3.1.1.2 Integrações OnixSat® e Rodosis®.....	37
3.1.1.3 Integrações ControlLoc® e Sascar®.....	38
3.1.2 Problemas Encontrados	42
3.2 PROJETO DO NOVO SISTEMA.....	44
3.2.1 Requisitos Funcionais	44
3.2.2 Requisitos não-funcionais	46
3.2.3 Diagrama de casos de Uso.....	47
3.2.4 Descrição dos Casos de Uso.....	48

3.2.5 Diagrama de Classes do Sistema.....	59
3.2.6 Diagrama de Entidade-Relacionamento do Banco de Dados.....	60
3.2.7 Diagramas de Seqüência do Sistema.....	61
3.2.8 Interface do Sistema.....	69
4 CONCLUSÃO	78
REFERÊNCIAS	79

1 INTRODUÇÃO

A Angellira Rastreamento Satelital obtém informações das tecnologias ControlLoc®, Omnilink®, OnixSat®, Rodosis® e Sascar®, provedoras dos sinais de seus equipamentos de rastreamento de veículos de cargas utilizando um *software* desenvolvido pela própria empresa que será denominado neste trabalho como *software* de integração. É através deste *software* que os dados de geoprocessamento e comunicação são recebidos das tecnologias de maneiras distintas e inseridos em um banco de dados gerando informações homogêneas

Com o crescimento na procura pelo ramo de rastreamento de veículos e conseqüentemente o aumento do número de clientes adquiridos, a quantia de dados trafegados entre as tecnologias e a empresa alvo deste estudo se elevou consideravelmente. Surgiu então a necessidade de ter um controle maior sobre estas trocas de informações e criar uma ferramenta que desempenhe o papel de monitoramento do recebimento de pacotes de dados provenientes destas tecnologias. O *software* de integração é uma importante ferramenta de integração de dados, que auxilia no desenvolvimento e utilização de outras ferramentas voltadas para o gerenciamento de risco e logístico de veículos de cargas.

Este trabalho propõe o desenvolvimento de projeto e implementação de um *software* que efetue o monitoramento das integrações de dados realizadas pelo *software* de integração para atuar em eventuais exceções ocorridas durante este processo e armazenar informações das exceções ocorridas. A ferramenta desenvolvida nesse estudo tem por objetivo auxiliar na otimização dos processos de integração, necessidade que surgiu com o aumento da carta de clientes e por conseqüência o número de tecnologias aplicadas no processo de gerenciamento e o tráfego de informações.

1.1 HISTÓRICO

A Angellira Rastreamento Satelital é uma empresa de gerenciamento de risco e logística que também desenvolve sistemas logísticos para transportes de cargas desde fevereiro de 2001. Os serviços oferecidos pela Angellira têm como objetivo principal garantir a segurança do veículo e do condutor. Com esse objetivo utiliza alguns dispositivos de rastreamento, aplicação e desenvolvimento de programas, entre outras medidas de segurança. (Angellira, 2009).

Sua matriz é situada na cidade de Chapecó, estado de Santa Catarina e, a exemplo do setor onde atua, tem expandido exponencialmente sua atuação em todo território nacional. A estrutura física da empresa conta com redundância completa de dados, geradores, *nobreaks*, *links*, *firewalls* e canais de dados próprios, instrutores para total informação do funcionamento de procedimentos de risco e logística para os motoristas e clientes usuários de seu site, técnicos especializados na manutenção preventiva de quaisquer equipamentos de rastreamento assegurando seu completo funcionamento. (Angellira, 2009).

A Angellira além do gerenciamento de risco realizado, disponibiliza gratuitamente aos clientes um exclusivo site onde o transportador/embarcador dispõem de vários recursos logísticos como: envio e recebimento de mensagens, históricos, posições, rotas utilizadas para tráfego, quilometragens percorridas, distâncias, exclusivo mapa de visualização rápida de posições, gerenciador de usuário, e vários outros controles como velocidade, temperatura, RPM, controle de custos integrados. Cada cliente possui uma necessidade diferente, e por isso precisa de uma ferramenta exclusiva, para tanto a empresa conta com uma *software-house* própria que desenvolve projetos e *softwares* que auxiliam na gestão do transporte. (Angellira, 2009).

1.2 OBJETIVOS GERAIS

O projeto visa desenvolver um sistema que proporcione o monitoramento das integrações de recebimento de dados entre a empresa Angellira® e as tecnologias

ControlLoc®, Omnilink®, OnixSat®, Rodosis® e Sascar® em tempo real, avaliando possíveis exceções na integração e possibilitando a ação imediata na prevenção e correção de falhas no processo.

1.3 OBJETIVOS ESPECÍFICOS

O projeto tem o intuito de atender os seguintes objetivos específicos listados abaixo:

- a) Armazenar resultados provenientes do processo de integração de pacotes de dados;
- b) Disponibilizar ao usuário a informação dos últimos eventos gerados pela tecnologia;
- c) Apresentar o desempenho do processo de integração quantificando os pacotes recebidos e o tempo percorrido, possibilitando o cálculo da velocidade média de inserção destas informações no banco de dados;
- d) Auxiliar através das informações apresentadas na identificação de exceções;
- e) Fornecer ao usuário consultas de históricos de alertas de exceções;
- f) Auxiliar o suporte gerando instruções de ação através de procedimentos pré-determinados de acordo com a exceção gerada.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentadas informações que são frutos de pesquisas que auxiliaram no entendimento do ramo em que a empresa estudada atua, bem como o funcionamento de seus processos automatizados e ferramentas utilizadas para a realização destes processos como também para o desenvolvimento deste trabalho.

2.1 GERENCIAMENTO DE RISCO NO TRANSPORTE DE CARGAS

O gerenciamento de risco é aplicado nas empresas de transportes rodoviários de cargas com o intuito de proteger o transporte de mercadorias desde a origem até o destino da carga. Ele emprega um conjunto de técnicas e medidas preventivas com a finalidade de identificar, avaliar e evitar possíveis danos no transporte. A empresa que adota esta estratégia precisa de planejamento, investimento, tecnologia e execução competentes. Dentre as ferramentas de gerenciamento de risco de frotas adotadas está o rastreamento e monitoramento de veículos via satélite. (Anefalos,1999).

Uma alternativa economicamente viável para empresas do setor de transporte de cargas é terceirizar os serviços de rastreamento de veículos. Estimavam-se em 2005 cerca de 230 empresas de gerenciamento de risco no Brasil, algumas com atuação em todo o território nacional, outras com atuação regional ou local e mais de 150 mil o número de caminhões monitorados por sistemas de rastreamento. (Neves, 2005).

2.1.1 Gerenciamento de Risco e Sistemas de Informação

Os sistemas de informação são os suportes que dispomos para armazenar uma informação qualquer e um meio de enviar esta informação a um destinatário, sendo fundamental para gerenciar risco em empresas de transporte de cargas através de ferramentas que proporcionam o rastreamento de veículos. Para tanto utilizam-se *hardwares* (equipamento rastreador instalado no veículo de carga, e servidores de armazenamento de dados nas empresas fornecedoras do equipamento rastreador) e *softwares* (ferramentas que possibilitam a transmissão de dados entre o veículo e a central de dados das empresas fornecedoras do equipamento rastreador). (Anefalos, 1999).

A visualização das trocas de informações entre o equipamento rastreador no veículo e a central de dados é obtida através de um *software* que a empresa fornecedora da tecnologia disponibiliza à transportadora proprietária do veículo. No

caso de gerenciadoras de risco, que terceirizam seu serviço de rastreamento para transportadoras e por trabalharem com variadas tecnologias do segmento necessitam receber estes dados através de outro canal de comunicação, possibilitando a integração destes dados em um único *software*.

A empresa observada durante este estudo rastreia e monitora veículos de carga que utilizam equipamentos rastreadores de cinco diferentes tecnologias. Ela desenvolveu um *software* de integração de dados que se comunica com as centrais destas tecnologias através deste canal diferenciado e disponibiliza os dados obtidos de forma homogênea para os usuários do seu sistema de rastreamento.

2.2 SISTEMAS INTEGRADOS DE RASTREAMENTO

Sistemas Integrados de Rastreamento são ferramentas utilizadas para controlar o fluxo de veículos no transporte de cargas, de modo a aumentar a segurança e a eficiência na utilização da frota. Os sistemas integrados de rastreamento provaram ser um método flexível e poderoso, resolvendo vários tipos de problemas logísticos e de segurança, além da obtenção de soluções para problemas das áreas de logística e segurança em tempo reduzido. Em geral, cada veículo é equipado com um módulo eletrônico que inclui um receptor de GPS (*Global Positioning System*) e um dispositivo de comunicação, que permite a troca de mensagens entre os veículos e uma Central de Controle. Este conjunto de é instalado no veículo e é denominado equipamento rastreador. (Moura, 2005).

Os sistemas disponíveis para esta função são: localização por direcionamento, triangulação de antenas e a utilização da constelação GPS. A localização por direcionamento tem por função emitir um sinal silencioso, criptografado a um satélite que retransmite esta informação a uma central que passa a visualizar a informação quando há um aviso de roubo ou furto. A triangulação de antenas é um sistema que segue o mesmo conceito aplicado aos satélites, porém utilizando antenas em terra. Oferece uma localização muito precisa, um custo de transmissão muito baixo, opera tanto *indoor* quanto *outdoor*, porém com uma área de abrangência limitada. (Moura, 2005).

O GPS é a mais conhecida de todas as constelações de satélites utilizados para localização. O sistema GPS só fornece as coordenadas de latitude e longitude, além de temporização. O uso do GPS é atualmente o mais comercializado no mercado de transportadores e operadores logísticos, com ele se tem três opções para rastreamento:

- a) GPS + Celular;
- b) GPS + Rádio;
- c) GPS + Satélite.

(Moura, 2005).

O sistema GPS é baseado em 24 satélites colocados em órbita do planeta Terra pelo governo dos Estados Unidos, os quais ficam permanentemente transmitindo sinais de controle para a Terra, de modo gratuito (Letham, 1996).

Dependendo do sistema escolhido pode-se ter uma solução simples, com apenas o rastreamento propriamente dito, até uma solução mais complexa que possui todas as funcionalidades integradas (rastreamento, troca de informações entre a central e o veículo, acionamento de sensores a longa distância).

2.3 RECEBIMENTOS DE DADOS

Cabe à gerenciadora de risco monitorar e rastrear veículos de transportadoras que contratam seus serviços independente da tecnologia que equipa a frota rastreada. Para atender estas peculiaridades a gerenciadora desenvolve um sistema de integração de dados.

As tecnologias ControlLoc®, Omnilink®, OnixSat®, Rodosis® e Sascar® disponibilizam uma forma de comunicação de dados particular a cada uma delas para a empresa foco deste estudo. Três delas necessitam de equipamentos servidores exclusivos, onde há seus respectivos bancos de dados recebendo informações provenientes de suas centrais e *softwares* específicos que realizam esta comunicação. As duas tecnologias restantes disponibilizam seus dados via *web service*.

2.3.1 Tecnologia Sascar®

Para se ter uma interface de integração com a tecnologia Sascar® primeiramente a gerenciadora de risco deve possuir um equipamento servidor com sistema operacional Linux®, banco de dados PostgreSQL® Server e acesso via SSH liberado. (Sascar, 2008).

O tráfego de informações é realizado diretamente entre o servidor e a central da Sascar® através do acesso SSH. Todos os dados provenientes da central são armazenados no banco de dados configurado no servidor Linux®. Para que a gerenciadora tenha acesso a estas informações a Sascar® disponibiliza acesso limitado à determinadas tabelas do banco de dados existente no servidor com a finalidade de possibilitar à gerenciadora implementar uma aplicação que efetue a busca destas informações. (Sascar, 2008).

2.3.2 Tecnologia ControlLoc®

A tecnologia Controlloc® libera o acesso ao seu banco de dados MySQL® para que seja implementada uma aplicação pela gerenciadora que busca estas informações. Também é necessário um equipamento de caráter servidor, porém neste caso utiliza-se o sistema operacional Windows® e o processo de atualização das informações enviadas e recebidas são realizadas através de uma aplicação exclusiva da tecnologia instalada no servidor que se comunica com sua central. (ControlLoc, 2008).

2.3.3 Tecnologia Omnilink®

A tecnologia Omnilink® necessita de um equipamento exclusivo do tipo servidor, nele é instalada uma aplicação denominada Interface de Acesso à Central Saver Turbo® (IAS), que realiza troca de informações com a central da Omnilink®

chamada Central Saver Turbo®. Para efetuar a integração de dados é criada pela gerenciadora de risco uma aplicação que se comunicará com o IAS via *socket* informando o IP do equipamento que possui o IAS instalado, assim como a porta pela qual se escutam as conexões (padrão 12000). (Omnilink, 2009)

A troca de informações entre a aplicação cliente implementada pela gerenciadora de risco e o servidor IAS será realizada pelo *socket* bidirecional trocando textos formatados em XML (*Extensible Modeling Language*) e conterà todas as informações enviadas pelo rastreador assim como as informações enviadas ao rastreador. Neste processo convencionam-se Tele comandos como sendo mensagens enviadas aos rastreadores e tele eventos como sendo mensagens recebidas dos rastreadores. (Omnilink, 2009)

Exemplificando um envio de mensagem (tele comando) através da interface de integração a aplicação cliente envia os dados do comando em formato XML para o servidor IAS, este transmitirá o comando para a Central Saver Turbo® que por sua vez se comunicará com o veículo enviando o comando. Após a atuação do comando no veículo é gerado um tele evento, em que o equipamento retorna o status de atuação do comando para a Central Saver Turbo®, que por sua vez re-transmite a informação ao servidor IAS o qual se comunica com a aplicação cliente retornando a atuação do comando.

Por questões de segurança será apresentado no Figura 1 apenas partes do documento XML de recebimento de um evento gerado pelo equipamento rastreador sendo omitido algumas *tags* e os valores constados nas *tags* apresentadas:

```

<TeleEvento>
  <UltimoTeleEventoProcessado>?</UltimoTeleEventoProcessado>
  <IdSeqMsg>?</IdSeqMsg>
  <Origem>?</Origem>
  <Destino>?</Destino>
  <TipoMsg>?</TipoMsg>
  <CodMsg>?<CodMsg>
  <TipoEvento>?</TipoEvento>
  <DataHoraEmissao>?</DataHoraEmissao>
  <IdTerminal>?</IdTerminal>
  <Latitude>?</Latitude>
  <Longitude>?</Longitude>
  <DataHoraEvento>?</DataHoraEvento>
  <Statusveic>?</Statusveic>
  <Intervalo>?</Intervalo>
  <Ignicao>?</Ignicao>
  <Temperatura>?</Temperatura>
  <IdSeqveiculo>?</IdSeqveiculo>
  <Localizacao>?</Localizacao>
  <ModeloRastreador>?</ModeloRastreador>
  <TamanhoMensagem>?</TamanhoMensagem>
</TeleEvento>

```

Figura 1: Layout de recebimento de evento da tecnologia Omnilink
Fonte: Omnilink (2009).

Tabela 1: Identificação de tags do documento XML da tecnologia Omnilink	
Tag	Descrição
<TeleEvento>	Recebimento de mensagem
<UltimoTeleEventoProcessado>	Último evento processado no IAS
<IdSeqMsg>	Id de sequencia gerado pelo sistema
<Origem>	Modulo de rastreamento que originou o evento
<Destino>	Base de comunicação que recebeu o evento
<TipoMsg>	Tipo da mensagem
<CodMsg>	Identificação da mensagem
<TipoEvento>	Identificação do tipo de evento recebido
<DataHoraEmissao>	Data e hora da emissão do evento
<IdTerminal>	Identificação do equipamento rastreador
<Latitude>	Latitude da posição do veículo
<Longitude>	Longitude da posição do veículo
<DataHoraEvento>	Data e hora da geração do evento
<Intervalo>	Intervalo de envio de posição pelo rastreador
<Ignicao>	Estado da ignição do veículo
<Temperatura>	Valor indicado pelo sensor de temperatura
<IdSeqVeiculo>	Id seqüencial de quando a comunicação foi gerada
<Localizacao>	Distância até o ponto de referência mais próximo
<ModeloRastreador>	Modelo do equipamento rastreador
<TamanhoMensagem>	Tamanho da mensagem em bytes

Fonte: Omnilink (2009).

2.3.4 Tecnologia Rodosis®

A tecnologia Rodosis® oferece acesso autenticado a seus serviços de *web service* através do protocolo HTTP (*HyperText Transfer Protocol*) /HTTPS (*HyperText Transfer Protocol Secure*) utilizando o método *POST*. Através deste serviço toda informação enviada ou recebida trafega em formato XML. Para ter acesso aos dados provenientes da central Rodosis® a gerenciadora necessita desenvolver uma aplicação que seja responsável por enviar e receber dados e que efetue a gravação de suas *ID's* (código de identificação da mensagem) de forma a controlar o seu fluxo e evitar duplicidade na coleta e armazenamento das informações. (Rodosis, 2008).

Para conhecimento seguem na Figura 2 e Figura 3 o formato de requisição e recebimento apenas de mensagens da tecnologia. Por questões de segurança, será apresentando parte dos seus respectivos documentos XML omitindo algumas *tags* e os valores constados nas *tags* visualizadas:

```

<RequestMensagensReport>
  <Autenticação login = "?" senha = "?"></Autenticação>
  <ForwardMessages nid = "?"></ForwardMessages>
  <ReturnMessages rid = "?"></ReturnMessages>
</RequestMensagensReport>

```

Figura 2: Layout de requisição da tecnologia Rodosis
Fonte: Rodosis (2008).

Tabela 2: Identificação de Tags do documento XML de requisição da tecnologia Rodosis

Tags	Descrição
<RequestMensagensReport>	Solicitação de mensagens
<Autenticação login = "?" senha = "?">	Autenticação do cliente informando login e senha de integração
<ForwardMessages nid = "?">	Solicitação de transmissão onde "nid" é a identificação da última transmissão recebida
<ReturnMessages rid = "?">	Solicitação de recebimento onde "rid" é a identificação do último evento de rastreamento recebido

Fonte: Rodosis (2008).

```

<MensagensReport>
  <ForwardMessages>
    <ForwardMessage nid = "?">
      <veiculo?></veiculo>
      <Fid?></Fid>
      <Status tipo = "?" timestamp = "?"></Status>
    </ForwardMessage>
  </ForwardMessages>
  <ReturnMessages>
    <ReturnMessage rid = "?">
      <veiculo?></veiculo>
      <DataHoraEvento?></DataHoraEvento>
      <TipoMensagem?></TipoMensagem>
      <TipoEvento?></TipoEvento>
      <Latitude?></Latitude>
      <Longitude?></Longitude>
      <Ignicao?></Ignição>
    </ReturnMessage>
  </ReturnMessages>
</MensagensReport>

```

Figura 3: Layout de retorno da tecnologia Rodosis
Fonte: Rodosis (2008).

Tabela 3: Identificação de Tags do documento XML de retorno da tecnologia Rodosis

Tags	Descrição
<MensagensReport>	Retorno de mensagens
<Forwardmessages>	Lista de notificação de status de transmissão
<ForwardMessages nid="?">	Notificação de status de transmissão onde "nid" é a identificação da transmissão
<Veiculo>	Placa do veículo
<Fid>	Identificação da mensagem enviada gerado no servidor
<Status tipo="?" timestamp="?">	Status da transmissão onde "tipo" é a identificação do status e "timestamp" a data e hora da notificação
<ReturnMessages>	Lista de eventos de rastreamento
<ReturnMessages rid="?">	Identificação do evento
<DataHoraEvento>	Data e hora do evento gerado no rastreador
<TipoMensagem>	Tipo da mensagem
<TipoEvento>	Tipo de evento recebido
<Latitude>	Latitude do posicionamento do veículo
<Longitude>	Longitude do posicionamento do veículo
<Ignicao>	Identificação do status da ignição do veículo

Fonte: Rodosis (2008).

2.3.5 Tecnologia OnixSat®

As informações provenientes da central da OnixSat® são enviadas à empresa através de um serviço de *web service*. Este *web service* foi desenvolvido pela OnixSat® para trabalhar apenas com requisição/resposta (*httprequest/httpresponse*), sendo somente XML sem a existência do WSDL (*Web Services Description Language*).. (OnixSat, 2009)

Este sistema de integração fornece informações dos veículos à gerenciadora e permite que a gerenciadora envie comandos e informações aos veículos. Ele consegue recuperar apenas as mensagens das duas últimas horas, ou seja, em caso do sistema ficar parado por mais de duas horas as mensagens fora deste intervalo serão perdidas. Para cada solicitação realizada o número máximo de mensagens recebidas é de mil, sendo que para cada solicitação deve-se respeitar um intervalo de tempo não permitindo que a mesma requisição seja repetida durante o mesmo período. (OnixSat, 2009).

Para conhecimento são apresentadas na Figura 4 e Figura 5 o formato de requisição e recebimento apenas de mensagens da tecnologia. Por questões de segurança será apresentando parte dos seus respectivos documentos XML omitindo algumas *tags* e os valores constados nas *tags* visualizadas:

```
<RequestMensagemCBDebug>
  <Login?></Login>
  <senha?></senha>
  <mId?></mId>
</RequestMensagemCBDebug>
```

Figura 4: Layout de requisição da tecnologia OnixSat
Fonte: OnixSat (2009).

Tabela 4: Identificação de tags do documento XML de requisição da tecnologia OnixSat

Tag	Descrição
<RequestMensagemCBDebug>	Solicitação de mensagem
<Login>	Usuário de acesso ao Webservice
<Senha>	Senha de acesso ao Webservice
<MId>	Id da última mensagem recebida

Fonte: OnixSat (2009).

No exemplo acima é enviada a requisição ao *web service* informando o *login* e a senha de acesso ao serviço e o valor da última mensagem recebida dos veículos no campo *<mId>* que sempre deverá ser preenchido. As próximas mensagens a serem recebidas virão a partir deste *Id* enviado.

```
<ResponseMensagemCB>
  <MensagemCB>
    <mId?></mId>
    <veId?></veId>
    <dt?></dt>
    <lat?></lat>
    <lon?></lon>
    <dMac?></dMac>
    <tfrID?></tfrID>
    <st1?></st1>
    <ori?></ori>
    <tpmsg?></tpmsg>
    <dtinc?></dtinc>
  </MensagemCB>
</ResponseMensagemCB>
```

Figura 5: Layout de retorno da tecnologia OnixSat
Fonte: OnixSat (2009).

Tabela 5: Identificação de tags do documento de retorno da tecnologia OnixSat	
Tag	Descrição
<ResponseMensagemCB>	Retorno de mensagem
<Mensagem>	Conteúdo da mensagem
<MId>	Id da Mensagem
<VeId>	Id do veículo
<Dt>	Data e hora da mensagem
<Lat>	Latitude de posicionamento do veículo
<Lon>	Longitude de posicionamento do veículo
<DMac>	Descrição de macro recebida
<TfrID>	Código da macro
<St1>	Valor do sensor de temperature 1
<Ori>	Origem da Mensagem
<Tpmsg>	Tipo da mensagem
<Dtinc>	Data de gravação no servidor da OnixSat

Fonte: Omnilink (2009).

2.4 XML (Extensible Modeling Language)

No final da década de 60 através de pesquisadores da IBM® foi criado o SGML (*Standard Generalized Markup Language*). Diferente de outras linguagens de programação, uma linguagem SGML não é voltada para a execução de ações em uma ordem designada e sim para um sistema de marcação capaz de transmitir e manipular informações independentemente de sistema operacional e formatos de arquivos. Ela é uma meta-linguagem voltada para especificação de linguagens de marcação sem possuir uma pré-definição de *tags* e elementos, porém necessita de um documento denominado DTD (*Document Type Definition*) que serve como uma gramática para a linguagem de marcação de texto. (Guimarães, 2005).

A linguagem HTML (*HiperText Markup Language*) criada por Tim Bernes Lee derivada do SGML e é especificamente voltada para a estruturação de documentos e sua apresentação visual na *web*. Através de marcações (*tags*) específicas é possível identificar os elementos que compõe este documento e apresentá-los na tela do computador como *websites* através de navegadores de internet. (Guimarães, 2005).

Com o surgimento de novas necessidades e as limitações existentes na HTML, foi proposta uma extensão desta linguagem, nascia a linguagem XML. Uma

linguagem baseada em marcadores, e apesar de possuir muitas semelhanças com a HTML em suas sintaxes básicas é voltada para outra finalidade. (Guimarães, 2005).

A XML herda várias características da sua antecessora, porém acrescenta melhorias e objetiva preencher lacunas que a HTML não preenchia. Tanto a XML, como a HTML são formas derivadas da SGML. A HTML deriva da SGML objetivando a confecção de páginas para a web, enquanto a XML é uma versão mais simplificada por possuir opções reduzidas de sintaxe. (Guimarães, 2005).

Esta linguagem de marcação separa a descrição da estrutura dos documentos de detalhes da apresentação, assim como em SGML, as *tags* na XML tem como prioridade descrever a estrutura dos documentos sem descrever a forma como eles serão apresentados, também possibilita a busca estruturada de informações baseando-se nelas. (Deitel, 2003).

Ela é considerada flexível, pois diferentemente do SGML não necessita de um DTD para cada documento. Além de flexível é uma linguagem simples contendo um conjunto de estruturas muito amplo que permite transferência e apresentação de conteúdos. Por ser no formato de texto a XML serve como meio de integração entre aplicações de diferentes plataformas, como por exemplo, na criação de *web services*. (Silva Filho, 2004).

2.4.1 Objetivos básicos da Xml

A linguagem XML é uma linguagem extensível, quando a comparamos com outros formatos observamos simplicidade característica. Tal característica foi possibilitada pelo projeto de metas na segunda edição recomendadas pelo W3C (*Word Wide Web Consortium*) para XML listadas a seguir:

- a) A XML deve ser utilizável diretamente da internet;
- b) A XML deve prover suporte à ampla variedade de aplicações;
- c) A XML deve ser compatível com a SGML;
- d) Deve facilitar a escrita de programas que processem documentos XML;
- e) A quantidade de recursos adicionais deve ser mantida ao mínimo, o ideal é zero;

- f) Documentos XML devem ser legíveis para pessoas e razoavelmente claros;
- g) O projeto XML deve ser reparado rapidamente;
- h) O projeto XML deve ser formal e conciso;
- i) Documentos em XML devem ser fáceis de criar;
- j) Concisão na marcação XML tem importância mínima.

(W3C, 2009).

Estas metas de projeto de XML objetivam facilitar a implementação mantendo a simplicidade no desenvolvimento. O documento XML possui uma estruturação em árvore podendo descrever a maioria das estruturas de dados necessários às aplicações. Por prover suporte à SGML e HTML tem sido incrementado com a adição de novas *tags* e rótulos à linguagem HTML em documentos XML. Esta flexibilidade é possível porque a XML permite que o programador defina e trabalhe com o conjunto de *tags* pertinente a sua ação. (Deitel, 2003).

2.4.2 Tipos de Aplicações

A XML é uma tecnologia em evolução, sendo utilizada nas mais diversas aplicações. Observa-se que por satisfazer as mais variadas demandas o emprego da XML resulta em uma melhor solução, como listado abaixo:

- a) *Resource description framework* (RDF) – *Framework* para descrição e troca de metadados que utilize XML como sintaxe comum tanto para troca como para processamento;
- b) *Wireless Markup Language* (WML) – Linguagem baseada em XML com a qual é possível visualizar em celulares o conteúdo e interface de usuário de página;
- c) *Chemical Markup Language* (CML) – Uma das primeiras aplicações que surgiu utilizando a XML, ela possibilita realizar a descrição das informações de estruturas moleculares;
- d) *Open Financial Exchange* (OFX) – Aplicação XML que permite transações financeiras;

- e) *Mathematical Markup Language* (MATHML) – Aplicação XML utilizada para descrever equações matemáticas. É uma ferramenta para web que oferece suporte à comunicação e à pesquisa científica;
- f) *Synchronized Multimedia Integration Language* (SMIL) – Aplicação XML que permite elaborar apresentações multimídia. O SMIL não especifica o conteúdo a ser apresentado, ele descreve quando e onde será executado um vídeo ou áudio;
- g) *Scalable Vector Graphics* (SVG) – Utilizado para descrever imagens bidimensionais usando linhas e curvas, dentre outros elementos para especificar a informação gráfica;
- h) *Vector Markup Language* (VML) – A aplicação XML direcionada a gráficos vetoriais que utiliza a descrição de um caminho de linhas e curvas para construir uma figura;
- i) *Voice XML* – Aplicação XML que permite interação de forma falada entre usuário e computador;
- j) *Talk ML* – Aplicação XML que oferece diálogo com interação por voz, tendo por objetivo ser aplicada em centros de atendimento;
- k) *Channel Definition Format* (CDF) – Tecnologia que permite o envio automático de conteúdo para usuários utilizando XML.

(W3C, 2009).

2.4.3 Composição de um documento XML

O documento XML é composto por dados, que são os blocos de textos, e marcações que formatam a estrutura destes blocos induzindo a interpretação de seu conteúdo. Uma marcação é sempre declarada entre os sinais “<” e “>” podendo ser composta por marcas de início e fim de um elemento, referência à entidade, instrução de processamento, declaração de tipo de documento ou comentário.

- a) Elementos;
- b) Atributos;
- c) Comentários;
- d) Entidades;

e) Instruções para Processamento.
(W3C, 2009).

2.5 WEB SERVICES

Novas Tecnologias de desenvolvimento estão surgindo após a popularização da internet. Aplicativos que permitem uma ampla integração entre as mais variadas plataformas de *softwares* e serviços que se encontram disponibilizadas na rede mundial de computadores. Ao contrário período que antecede o auge da internet, quando as tecnologias disponíveis permitiam apenas a apresentação de informações, hoje é possível realizar troca de informações na grande rede através de aplicações distribuídas. Este modelo em crescimento trata tarefas complexas, como o gerenciamento de transações, através de serviços distribuídos que utilizem interfaces de acesso simples e bem definidas. (Cunha, 2002).

Os *web services* são serviços distribuídos muito utilizados na integração entre plataformas diferentes permitindo que novas aplicações possam se comunicar com aquelas já existentes. Seu principal objetivo é realizar a comunicação de aplicações de forma direta com outras aplicações fazendo o uso da internet. Os *web services* são ideais para comunicação entre sistemas. Pois a comunicação entre os serviços são padronizadas possibilitando esta independência de plataforma e de linguagem de programação. (Reckziegel, 2006).

2.5.1 Tecnologias Utilizadas na Criação de Web Services

A base dos *web services* é a linguagem XML, pois ela é utilizada no formato de transmissão, na descrição e na forma de armazenamento deste componente. A troca de informações é realizada pelo protocolo SOAP (*Simple Object Access Protocol*) através do transporte de dados pelo protocolo HTTP. Os seus serviços são descritos através da linguagem WSDL (*Web Services Description Language*) e o

processo de publicação, pesquisa e descoberta é realizado pelo protocolo UDDI (*Universal, Description, Discovery and Integration*). (Sampaio, 2006).

2.5.1.1 Soap

O SOAP (*Simple Object Access Protocol*) baseia-se numa invocação remota de um método e para isto precisa especificar o endereço do componente, o nome do método e os argumentos para esse método. Em relação à Web o SOAP é um protocolo de RPC (*Remote Procedure Call*) que funciona sobre HTTP de forma a ultrapassar as restrições de segurança/*firewalls* normalmente impostas aos sistemas clássicos de RPC como RMI (*Remote Method Invocation*), DCOM (*Distributed Component Object Model*), CORBA (*Common Object Request Broker Architecture*) e IIOP (*Internet Inter-Orb Protocol*). Em vez de usar HTTP para solicitar a uma página HTML sua visualização em um browser o SOAP envia uma mensagem de XML através do pedido HTTP e recebe uma resposta, se existir, através da resposta do HTTP. (Sampaio, 2006).

2.5.1.2 Wsdl

O WSDL (*Web Services Description Language*) é linguagem baseada em XML para descrever o serviço dos métodos do *web service* desenvolvida pelo W3C. Ele descreve os serviços disponibilizados à rede através de uma semântica XML, este providencia a documentação necessária para se chamar um sistema distribuído e o procedimento necessário para que esta comunicação se estabeleça. Enquanto que o SOAP especifica a comunicação entre um cliente e um servidor, o WSDL descreve os serviços oferecidos. (Sampaio, 2006).

2.5.1.3 Uddi

O UDDI (*Universal Description Discovery and Integration*) é um protocolo desenvolvido para organizar e registrar os *web services* com o objetivo de acelerar a sua interoperabilidade. Um registro UDDI contém três tipos de informação:

- a) Informações gerais de cada organização;
- b) Informações de organizações e serviços por categorias de negócios;
- c) Informações técnicas sobre os serviços providenciados pelas organizações.

(Sampaio, 2006).

O UDDI providencia três funções principais, conhecidas como publicação, descoberta e ligação:

- a) Publicação: permite que uma organização divulgue os seus serviços;
- b) Descoberta: permite que o cliente do serviço, procure e encontre um determinado serviço;
- c) Ligação (*bind*): permite que o cliente do serviço, possa estabelecer a ligação e interagir com o serviço.

(Sampaio, 2006).

2.6 THREADS

Na escala nano métrica, providencial na era atual, frações de segundo são extremamente importantes para os programas. É aí que os *threads* se encaixam, é por causa deles que muitos mecanismos e computadores alcançaram maior velocidade na execução de tarefas que antes eram muito demoradas, daí sua importância para a computação atual. (Scheffer, 2007).

Um *thread* é uma unidade singular dentro de um processo, ele é leve não exigindo muito do sistema operacional, é fácil de criar, de se manter e de se gerenciar. O número de threads dentro de um processo vai depender da

complexidade do sistema que compõe. Nos *threads* o fluxo de instruções executadas permite que múltiplas execuções ocorram num mesmo ambiente com um grande grau de independência, permitindo que a execução de uma tarefa não comprometa a execução de outra. (Scheffer, 2007).

Os *threads* possuem identidade própria, são escalonadas no processador e compartilham os mesmos recursos do processo pai. Todos os *threads* de um processo compartilham a memória e os recursos desse processo podem se comunicar sem a intermediação do núcleo. Permitem explorar plenamente o paralelismo real das máquinas com multiprocessadores aumentando o número de atividades executadas por unidade de tempo e podem associar mais *threads* à dispositivos de entrada e saída. (Scheffer, 2007).

2.6.1 O Ciclo de Vida dos Threads

Num determinado momento durante a execução da tarefa, o *thread* pode estar em um de seus vários estados. Ele inicia seu ciclo de vida no estado “novo”, e irá permanecer nesse estado até que o sistema operacional lhe atribua ao processador. Neste momento o *thread* entra no estado executável, quando recebe um quantum (uma pequena fração de tempo que o processador determina para a execução do thread) e executa a tarefa que lhe foi destinada. Ela passa então ao estado de espera, quando o processador determina novamente um quantum para a execução de outro *thread*. Enquanto o *thread* está no estado de espera ele está aguardando para retornar ao estado executável. (Scheffer, 2007).

O estado de espera pode ser sincronizado, ou seja, é estabelecido um período para que este *thread* permaneça no estado de espera, quando esse tempo expirar ele vai retornar ao estado executável. Ainda que com o processador disponível, enquanto esse período não acabar o *thread* não será executado. Outra modalidade de *thread* sincronizado é o *thread* adormecido, ele fica nesse estado quando não tem que realizar tarefa alguma por um determinado intervalo de tempo, este intervalo é denominado “intervalo de adormecimento”. Quando o *thread* termina de executar a sua tarefa ele entra no estado “terminado”, esse estado também pode ser alcançado quando ocorrer um erro. (Scheffer, 2007).

2.6.2 Gerenciamento dos Threads

O gerenciamento dos *threads* pode ser implementado em dois níveis, de usuário e de sistema. Pode-se dizer que no nível de usuário o sistema operacional não “enxerga” o *thread* e no nível de sistema o sistema operacional “enxerga” o *thread*. (Scheffer, 2007).

No gerenciamento do nível de usuário todas as tarefas de gerenciamento dos *threads* são realizadas pela aplicação. Os *threads* são implementados por uma biblioteca ligada à aplicação API (*Application Programming Interface*) que realiza as funções relacionadas pelos threads. Neste nível não se executa os *threads* de maneira privilegiada, o chaveamento não requer a intervenção do núcleo e o escalonamento é específico da aplicação sem passar pelo *kernel*, entretanto é normal que chamadas do sistema bloqueiem esses processos pois os *threads* não utilizam o processador. (Scheffer, 2007).

No gerenciamento do nível de sistema o núcleo é quem gerencia os *threads*, pois o chaveamento destas requer a sua intervenção. Neste nível de gerenciamento os *threads* fazem uso do processador, e se por ventura, ocorrer um bloqueio de um *thread* o núcleo poderá executar outro *thread* deste mesmo processo. O núcleo pode escalonar vários *threads* de um mesmo processo em processadores distintos, explorando assim o paralelismo de máquinas multiprocessadas. (Scheffer, 2007).

2.6.3 Multithreading

Multithreading é a capacidade que o sistema operacional tem em suportar múltiplos *threads* de execução dentro de um único processo e executar vários processos dentro de um programa. (Tenenbaum, 1999).

Quando um mecanismo possui mais de um processo sendo executado em uma seqüência rápida, e cada processo possuir mais de um *thread*, este mecanismo é denominado *multithread*. O mecanismo *Multithread* permite que o espaço de endereçamento dos processos seja compartilhado com os threads, alocando menos memória e trabalhando em conjunto. Neste mecanismo cada um dos processos

poderá responder várias solicitações de forma concorrente ou simultânea em caso de existência de mais de um processador. (Tenenbaum, 1999).

Os modos de implementação de *threads* podem variar de um sistema operacional para outro como, por exemplo:

- a) O MS-DOS® suporta um único *thread* (*single-thread*)
- b) O UNIX® suporta vários processos mas somente um *thread* por processo (*single-thread*);
- c) O Windows® NT suporta vários *threads* por processo (*multithread*);
- d) O Windows® 2000, Solaris®, Linux®, Mach® e OS/2®, suportam vários processos com múltiplos *threads* (*multithread*).

2.6.4 Threads no Delphi®

Para criar um *thread* no Delphi® basta criar uma classe descendente da classe *TThread*. A classe *TThread* é pré definida e comporta métodos e eventos já implementados pelo Delphi® que auxiliam na criação do *thread*. Esta classe é abstrata, não podendo ser utilizada diretamente, para isto é necessário o método *Execute*. Este método é responsável por todo o funcionamento do *thread*. (Marotta, 2007).

Para que o *thread* realize suas tarefas existem outros métodos que deverão ser implementados, cada um com a sua devida função:

- a) *Resume* – executa o método *execute*, retomando o funcionamento da *thread*;
- b) *Suspend* – pausa a execução do *thread*, ou seja, suspende a sua execução do ponto onde estiver para que se possa dar continuidade posteriormente;
- c) *Terminate* – passa o valor *true* para a propriedade *terminated* do *thread*. Esta propriedade deverá sempre ser verificada no método *execute* para que a execução do *thread* seja encerrada;
- d) *Synchronize* – permite acesso a partes da VCL (*Visual Component Library*) sem que se perca o controle da aplicação. Efetua um procedimento do

thread principal sendo necessário definir qual o procedimento a ser realizado.

(Pauli, 2005).

2.7 ENGENHARIA DE SOFTWARE

Em meados da década de 70 surgiu a Engenharia de *Software* para contornar a *crise do software* e assim tornar o desenvolvimento de sistemas de *softwares* complexos mais controlados e sistemáticos. A Engenharia de *Software* é a metodologia de desenvolvimento e manutenção de sistemas modulares tendo como características um processo dinâmico, integrado e inteligente de soluções tecnológicas e adequação dos requisitos funcionais do negócio do cliente. (Rezende, 2005).

A Engenharia de *Software* avalia e garante as qualidades do *software*, pois utiliza modelos abstratos que permitem especificar, projetar, implementar e mantê-lo oferecendo mecanismos para se planejar e gerenciar o processo de desenvolvimento. (Rezende, 2005).

2.7.1 Requisitos

Os requisitos devem ser elaborados no início de um projeto de *software*, onde são fundamentais para o desenvolvimento, pois definem as atividades que o *software* fará quando estiver pronto. Requisitos bem definidos evitam alta manutenção e facilitam o desenvolvimento e implementação do projeto, de forma que atenda as expectativas do usuário ou cliente. Nesta fase deve-se alocar tempo suficiente para o esclarecimento dos requisitos funcionais e não funcionais do *software*. (Rezende, 2005).

2.7.1.1 Requisitos Funcionais e não-funcionais

Requisitos são divididos em requisitos funcionais e não-funcionais. Requisitos funcionais descrevem as funções que geralmente os usuários necessitam que o *software* ofereça, ou seja, definem a funcionalidade desejada do *software*. A especificação de um requisito funcional deve determinar o que se espera que o *software* faça, sem a preocupação de como ele faz. (Sommerville, 2007).

Os requisitos não-funcionais são qualidades que englobam o funcionamento do *software*, como por exemplo, usabilidade e desempenho. A descrição dos requisitos não-funcionais normalmente é realizada de maneira informal. (Sommerville, 2007).

Os exemplos de requisitos funcionais e não funcionais podem ser visualizados nos itens 3.2.1.1 e 3.2.1.2 da análise do sistema apresentado neste trabalho.

2.7.2 Projeto do Software com UML

A UML (*Unified Modeling Language*) é uma das linguagens mais utilizadas para especificar e projetar *softwares* abordando a orientação à objetos. Ela é utilizada para traduzir os requerimentos do sistema, em um nível mais próximo do usuário, para codificação, em um nível mais aproximado da aplicação. Por se encontrar entre estas duas camadas é que a UML pretende ser de fácil entendimento se tornando um meio de comunicação. (OMG, 2010).

Através de seus diagramas gráficos a UML facilita visualizar e discutir soluções entre a equipe de desenvolvimento e o usuário. A UML possui seu padrão administrado pela *Object Management Group* (OMG). A função da OMG é de divulgar e organizar as especificações da UML, controlando o que deve ou não ser incluído em sua arquitetura. (OMG, 2010).

3 ANÁLISE DO SISTEMA

Neste capítulo será realizado um breve estudo sobre o sistema existente na empresa e posteriormente apresentado o projeto do novo sistema que é objeto deste trabalho.

3.1 ESTUDO DO SISTEMA EXISTENTE

A empresa alvo deste estudo desenvolveu um *software* que tem por capacidade reunir em apenas um aplicativo a integração de dados provenientes de diferentes tecnologias de rastreamento. Para acompanhar o funcionamento deste *software* a empresa dispõe de colaboradores da área de suporte técnico que acompanham os recebimentos destas informações através de outra aplicação criada por ela que consulta dados provenientes destas inserções de registros no seu banco de dados.

Para um melhor entendimento será descrita a forma como são integradas estas informações, como são verificadas as exceções que ocorrem no decorrer deste processo e os problemas encontrados nesta forma de verificação.

3.1.1 Integração de Dados

A inserção de informações no banco de dados da empresa é realizada pelo *software* de integração desenvolvido através da tecnologia Delphi® 2007. Este *software* de integração reúne em um único aplicativo o processo de integração de dados provenientes de cinco diferentes tecnologias, apresentados no item 2.1 da fundamentação teórica, entre outros processos automatizados que auxiliam o sistema de monitoramento de veículos. Ele é formado por uma tela que contém a lista das integrações com uma opção para habilitar ou desabilitar cada uma delas e

um campo que informa o intervalo de execução da integração selecionada, conforme a Figura 6.

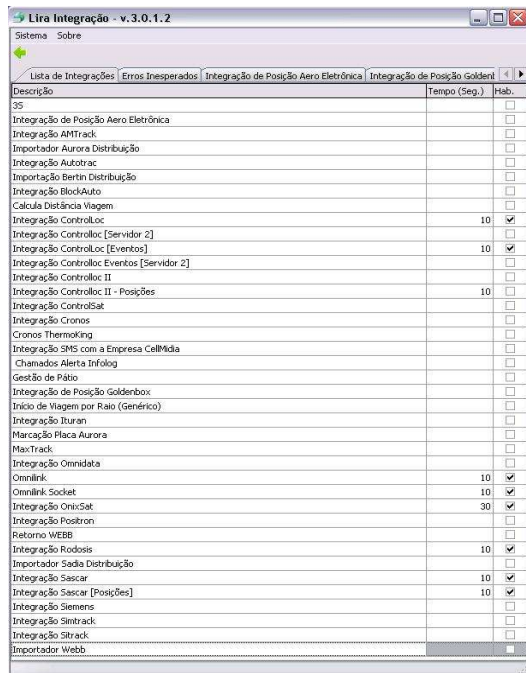


Figura 6: Software de integração em atividade.

O *software* de integração utiliza como tecnologia processamento paralelo, *threads*, onde se processa as informações de posicionamento, mensagens e alertas recebidas por cada tecnologia, de acordo com a forma que as mesmas disponibilizam os dados para integração. Estas tecnologias disponibilizam seus dados de maneiras distintas e a função do *software* de integração é coletar estes dados atendendo cada particularidade e inserir no banco de dados da empresa, como ilustrado na Figura 7.

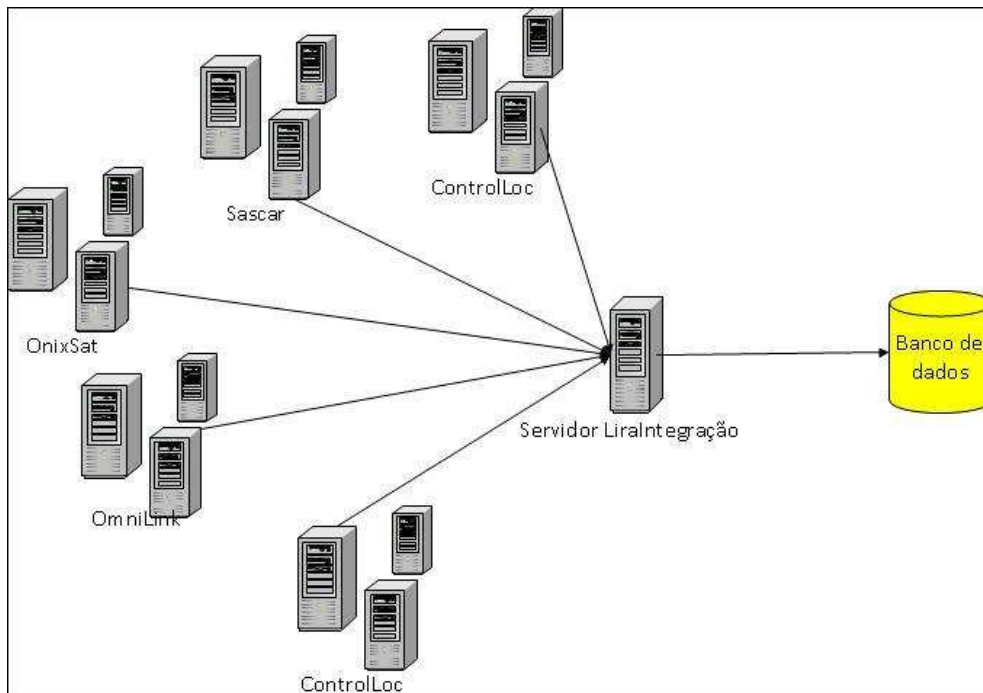


Figura 7: Processo de integração visão geral.

Em resumo o *software* de integração processa a unificação em um único banco de dados informações de tecnologias distintas em um formato padronizado permitindo assim que o sistema de rastreamento de veículos da empresa utilize dados homogêneos para a realização de suas tarefas.

3.1.1.1 Integração Omnilink®

A tecnologia possui um aplicativo que realiza a disponibilização das informações atuais de seu banco de dados no formato XML, descrito no item 2.3.3 da fundamentação teórica. O *software* de integração se conecta ao aplicativo IAS, troca dados através de documentos XML. O retorno dos dados emitidos pelo IAS são lidos pelo *software* de integração que extrai as informações provenientes da central da tecnologia e as insere no banco de dados da empresa, como ilustrado na Figura 8.

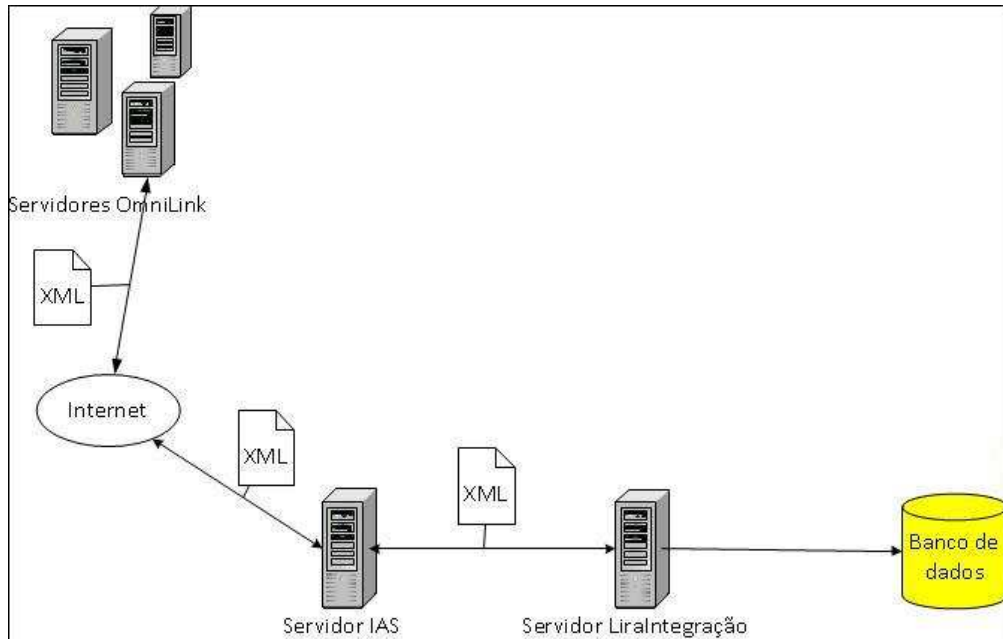


Figura 8: Processo de integração tecnologia Omnilink.

3.1.1.2 Integrações OnixSat® e Rodosis®

Estas tecnologias disponibilizam suas informações para integração de dados por meio de Web Services, citado nos itens 2.3.4 e 2.3.5 da fundamentação teórica. Nestes casos o que ocorre na execução do *software* de integração em busca de dados recentes é a conexão a estes *web services*, e através de requisições buscar os dados em formato XML. Cada tecnologia adota uma formatação para seu documento XML e conforme estes dados são lidos ocorre a inserção no banco de dados da empresa como ilustrados na Figura 9 e Figura 10.

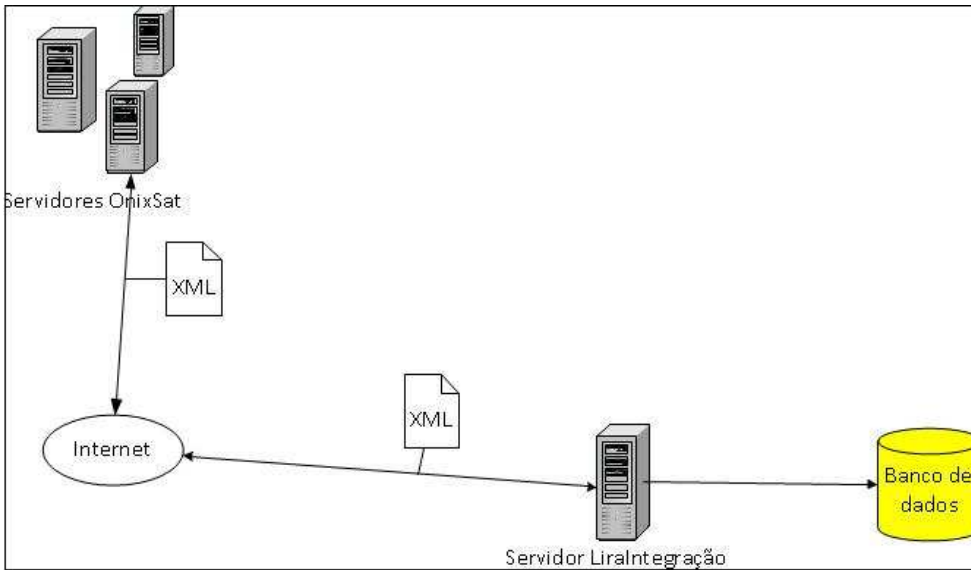


Figura 9: Processo de integração tecnologia OnixSat.

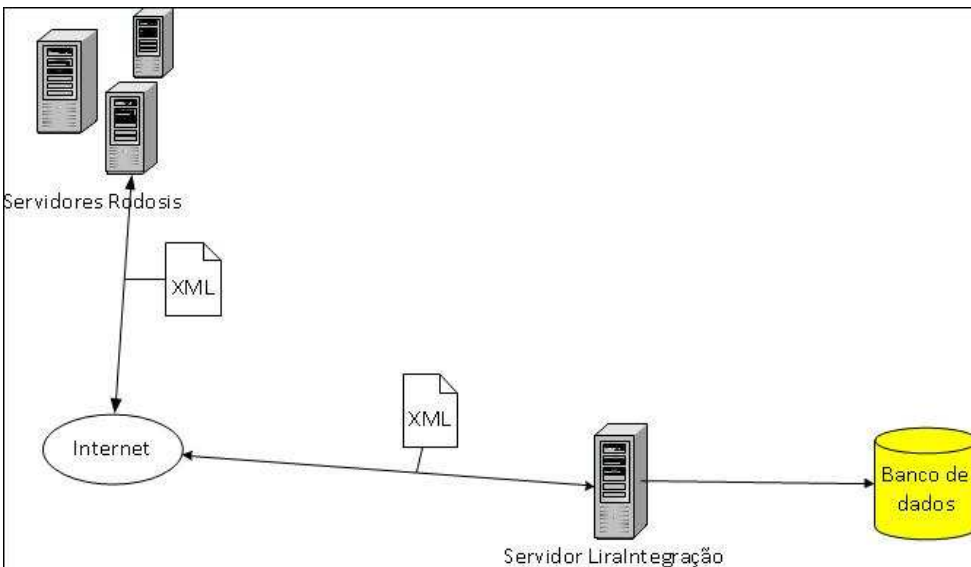


Figura 10: Processo de integração tecnologia Rodosis.

3.1.1.3 Integrações ControlLoc® e Sascar®

As integrações Controlloc® e Sascar®, processos já descritos nos itens 2.3.1 e 2.3.2 da fundamentação teórica necessitam de um equipamento servidor exclusivo para seus bancos de dados. O processo de integração destas tecnologias é realizado através de acesso direto aos seus respectivos bancos de dados por intermédio do *software* de integração. A cada execução de busca realizada pelo *software* de integração são efetuadas consultas SQL nestas bases e os resultados

provenientes destas consultas, quando houver novas informações, são inseridos no banco de dados da empresa, como ilustrados na Figura 11 e Figura 12.

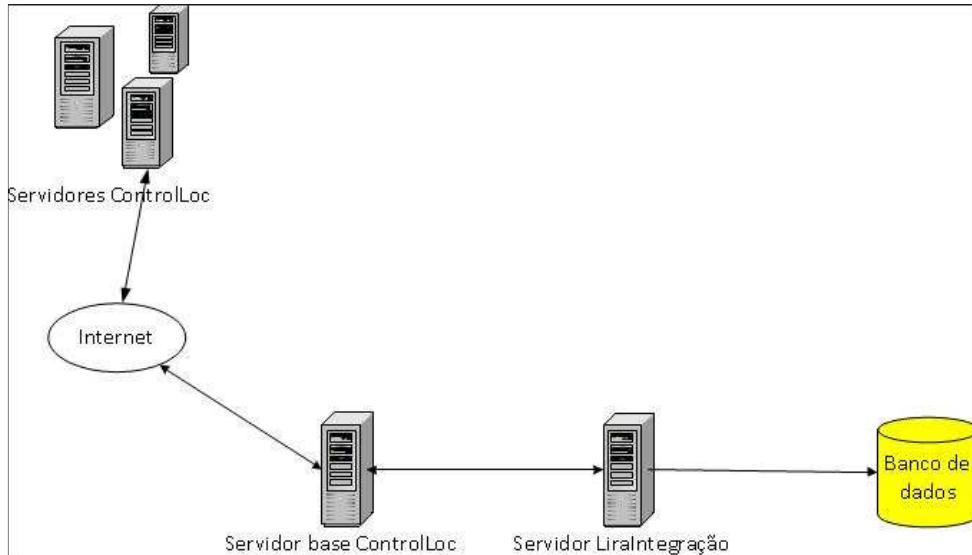


Figura 11: Processo de integração tecnologia ControlLoc.

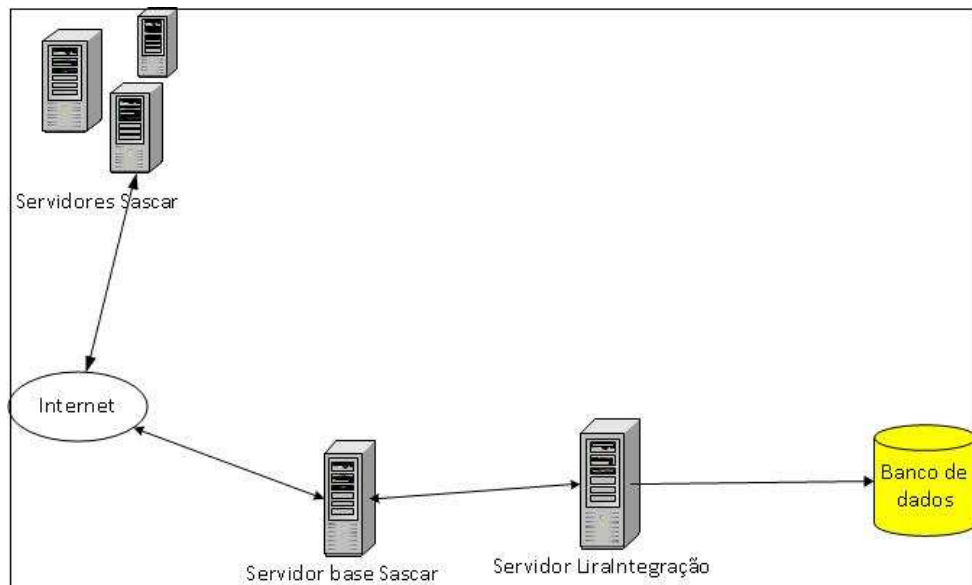


Figura 12: Processo de integração tecnologia Sascar.

3.1.2 Verificação das integrações

O *software* de integração integra o recebimento de dados de posicionamento, mensagens e alertas emitidos pelas tecnologias referente a seus equipamentos instalados nos veículos de cargas. As informações de posicionamento são emitidas

pelos equipamentos constantemente enquanto as mensagens e alertas são enviados ao passo que um evento é gerado, como por exemplo, uma solicitação de parada para pernoite enviada pelo motorista através do teclado do computador de bordo ou um alerta de violação de carga emitida pelas travas de segurança quando estas são forçadas. Como as mensagens e alertas são provenientes da ocorrência de uma ação, toma-se como referência para a funcionalidade de recebimento de dados as informações de posicionamento, pois estes são emitidos automaticamente de acordo com o período configurado por cada tecnologia.

Ao passo que o *software* de integração é executado ele efetua a inserção de dados de posições recentes em uma tabela específica de posicionamento no banco de dados. No fim de cada execução o *software* de integração armazena o horário de início e fim do processo recém efetuado e o valor total de registros de posicionamento integrados neste período em uma tabela de *log* no banco de dados.

Quando estas informações são inseridas em suas devidas tabelas é realizado um processo por meio de *triggers* no banco de dados que realiza o *update* de informações contidas em uma tabela de atualização constante. Nesta tabela há informações como nome da tecnologia, horário da última execução da integração e horário do último posicionamento gerado pela tecnologia.

As informações contidas nesta tabela são visualizadas através um *software*, desenvolvido pela empresa alvo deste trabalho em Delphi® 7, que a cada sete minutos que realiza pesquisa através de comandos em SQL apresentando os resultados na tela para o usuário. Este *software* será mencionado neste trabalho como verificador de integração e é apresentado na Figura 13.

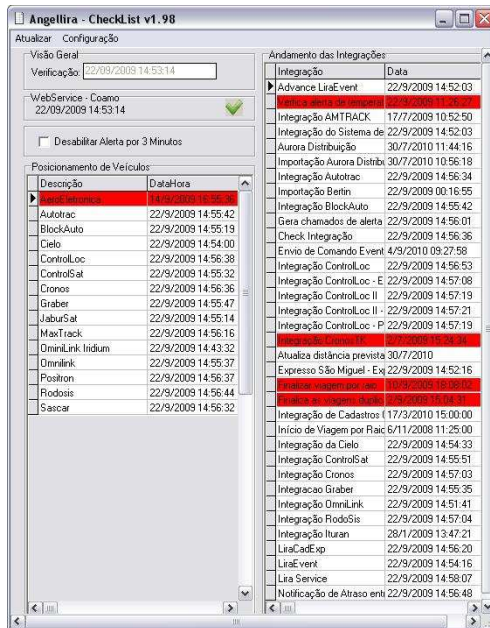


Figura 13: Software verificador de integrações em atividade.

O *software* verificador tem em sua tela principal duas grades, onde na grade da direita são listadas as tecnologias, o horário do posicionamento mais recente dentre os já recebidos por seus equipamentos e a tolerância estipulada em minutos que faz relação entre o horário do posicionamento apresentado e o horário atual. Na grade da esquerda são apresentadas as integrações com as tecnologias efetuadas pelo *software* de integração e também os demais processos automatizados realizados por ele com suas respectivas informações de tolerância, que neste caso é utilizada como parâmetro para a relação entre o horário de finalização do último processo de integração e o horário atual.

A finalidade da primeira grade citada é identificar o não recebimento de dados sendo que na segunda é a identificação de uma não-conformidade no processo de integração. Quando a diferença entre os horários apresentados nas grades do *software* verificador e a hora atual são maiores do que os valores de suas respectivas tolerâncias a linha correspondente a tecnologia em determinada grade é pintada na cor vermelha e um bip é acionado como forma de alerta. Com base nisto o funcionário de suporte ao visualizar o alerta toma as devidas providências para que o processo volte a ser realizado de acordo com o esperado.

3.1.2 Problemas Encontrados

A empresa encontra grande dificuldade em identificar algumas instabilidades que ocorrem, muito por falta de informações. O *software* verificador apresenta somente informações momentâneas da última integração executada e do posicionamento mais recente emitido por uma tecnologia, não oferecendo ambiente para pesquisas e relatórios, em determinado período, que referenciam anomalias ocorridas no recebimento de informações.

Sabe-se que um histórico é de grande valia para a apresentação de informações mais elaboradas que por sua vez permitem uma análise mais proveitosa da exceção no recebimento de dados que está ocorrendo e encurtando o caminho em uma série de atuações corretivas. Hoje são visualizados os alertas apenas no momento em que se ocorre, sendo que tais anomalias se repetem e não há um procedimento padrão para tais correções, o retrabalho para o colaborador responsável a dar o devido suporte quando necessário é inevitável.

Ao utilizar o horário de posicionamento como parâmetro para avaliar o recebimento de dados de determinada tecnologia o *software* verificador se torna suscetível a falhas, pois este horário “atual” não é a data de cadastro da informação no banco de dados e sim a data emitida pelo equipamento rastreador instalado no veículo e nem sempre é adquirido no processo de integração mais recente camuflando uma possível falta de dados durante um período.

Supondo que dentre os registros recebidos de uma tecnologia em um pacote integrado as 21h00min pelo *software* integrador tenha-se recebido um registro de posicionamento que traz o horário da posição de um veículo igual as 21h15min. Logo que esta integração ocorre é realizada a pesquisa pelo *software* verificador que apresenta na grade de posições este horário como o posicionamento mais recente. Sabendo que o *software* verificador realiza suas pesquisas a cada sete minutos, a tolerância estipulada para a tecnologia neste caso seja de dez minutos e não ocorra mais o recebimento de dados provenientes desta tecnologia ocorrerá o seguinte:

- a) As 21h07min o *software* verificador realizará nova verificação de recebimento de registros, como está sendo apresentado última posição igual a 21h:15min o horário este dado estará 8 minutos a frente do horário de execução da verificação;

- b) As 21h14min o *software* verificador realizará outra vez a pesquisa e o horário da posição estará um minuto adiantado em relação ao horário de verificação e não será desenhada a linha vermelha de alerta na grade de posições, quando deveria pois já se ultrapassou o limite estipulado como tolerância em quatro minutos;
- c) As 21h21min será efetuada nova verificação, o horário de posicionamento apresentado estará atrasado 6 minutos em relação à tolerância estipulada, quando na verdade já ultrapassou em onze minutos;
- d) As 21h28min ao ser novamente realizada a pesquisa pelo *software* verificador a linha correspondente à tecnologia será desenhada em vermelho alertando o suporte, pois de acordo com o que está sendo apresentado em tela o então horário do posicionamento estará atrasado 13 minutos e fora do período de 10 minutos de tolerância, quando já se está sem receber informações a 28 minutos após o último recebimento de dados.

A informação de ultima posição gerada é de grande importância para o processo de rastreamento de veículos, onde o operador de teleprocessamento se foca neste dado especificamente para acompanhar determinado veículo em sua viagem, mas não como informação segura para constatar que está ou não recebendo informações.

3.2 PROJETO DO NOVO SISTEMA

Visto que há o armazenamento de dados em uma tabela que relatam o total de registros inseridos, quando iniciou e quando finalizou um processo de integração tem-se como finalidade criar um *software* que realize o monitoramento das integrações efetuadas pelo *software* de integração utilizando como base esta tabela e excluindo a verificação de recebimento de informações através do horário de posicionamento dos veículos.

Aplicar cálculos matemáticos aos dados de execução da integração extraíndo assim informações que vão além do que se encontra armazenado, como por exemplo, quanto tempo durou este recebimento, utilizando as informações de início e fim do processo e relacionar a quantia de dados recebidos com o tempo percorrido para inseri-los no banco de dados com a finalidade obter a velocidade da integração, além de calcular o intervalo entre as execuções efetuadas pelo *software* de integração.

Definir parâmetros para o bom funcionamento da integração e quando o processo estiver abaixo do que estiver estipulado, além de gerar o alerta ao usuário e armazenar a informação referente a exceção que aconteceu para que se possa ter um histórico do ocorrido.

3.2.1 Requisitos Funcionais

- a) O *software* deve permitir a manutenção (cadastro, alteração ou exclusão) de tecnologias que disponibilizem interface de integração com a empresa como, nome, email e telefone;
- b) O *software* deve permitir a manutenção (cadastro, alteração ou exclusão) dos parâmetros abaixo que definem o funcionamento ideal para a execução de busca e inserção de pacotes obtidos pela integração da empresa com as tecnologias;
 - Faixa máxima de registros por pacote,
 - Faixa mínima de registros por pacote,

- Velocidade máxima de inserção de registros,
 - Velocidade mínima de inserção de registros,
 - Tempo de execução para integração do pacote,
 - Intervalo entre as execuções de integração,
 - Número de tolerância para integrações de pacotes com registros abaixo da faixa mínima,
 - Número de tolerância para integrações de pacotes com velocidade de inserção de registros abaixo da velocidade mínima,
 - Número de tolerância para integrações de pacotes sem registros,
 - Número de tolerância para integrações com atraso (tempo de execução acima do estipulado),
- c) O *software* deve permitir a manutenção (cadastro, alteração ou exclusão) de informações de usuários permitindo acesso ao sistema como nome completo do usuário, login e senha para acesso;
- d) O *software* deve permitir a manutenção (cadastro, alteração ou exclusão) de informações exceções que venham ocorrer durante a integração de pacotes de dados que correspondem à descrição da exceção e procedimentos a serem seguidos quando ela ocorrer;
- e) O *software* deve permitir a visualização dos dados abaixo referentes às integrações das tecnologias cadastradas em sua execução mais recente;
- Horário da execução obtido através da data de fim da integração armazenada no banco de dados,
 - Quantidade de registros obtidos através do armazenamento no banco de dados do número de informações adquiridas na integração,
 - Tempo percorrido obtido através da diferença entre a data de fim e data de início da integração,
 - Velocidade de inserção obtida através do calculo de divisão da quantidade de registros pelo tempo percorrido na integração,
- f) O *software* deve permitir a visualização dos gráficos abaixo efetuando o comparativo entre os parâmetros cadastrados e os resultados das últimas execuções da integração de determinada tecnologia escolhida pelo usuário;

- Gráfico de velocidade: gráfico em linha contendo faixa máxima e mínima de inserção de registros e a velocidade de cada integração ocorrida nos últimos trinta minutos,
- Gráfico de registros: gráfico em linha contendo faixa máxima e mínima de registros por pacote e a quantidade de registros a cada integração ocorrida nos últimos trinta minutos;
- g) O *software* deve emitir alerta em tela quando ocorrer alguma exceção informando qual exceção ocorreu e os passos a serem realizados para estabilizar a integração;
- h) O usuário deve poder inserir informações no alerta que correspondem à sua atuação para estabilizar a integração que está abaixo dos parâmetros;
- i) O usuário deve poder finalizar o alerta depois de ser identificada a estabilidade da integração, pelo sistema, que gerou o alerta;
- j) O usuário administrados deve poder ativar ou desativar o monitoramento de integração de uma tecnologia por ele escolhida;
- k) O usuário administrador deve poder obter através de relatório o desempenho da atuação dos usuários suporte na ação para estabilizar os processos de integração em um período determinado por ele;
- l) O usuário administrador deve poder obter através de relatório os alertas ocorridos durante o processo de execução das integrações num todo, ou de uma tecnologia em específico, através de gráfico em um período determinado por ele;

3.2.2 Requisitos não-funcionais

- a) O tempo de resposta do *software* não deve ultrapassar um minuto;
- b) O *software* deve ser implementado em Delphi® 2009;
- c) O *software* deve ser operacionalizado no sistema operacional Windows®;
- d) O banco de dados deve ser Microsoft Sql Server® 2005;
- e) O banco de dados deve ser protegido e separado da base de produção;

- f) O banco de dados deve manter uma rotina de *backup* para garantir histórico de informações;
- g) O *software* deve apresentar informações ao usuário através de *feedback* gráfico auxiliando na compreensão dos dados.

3.2.3 Diagrama de casos de Uso

O diagrama de casos de uso do sistema é a visão do usuário em relação ao que pode ser executado pelo sistema, a Figura 14 apresenta o diagrama de casos de uso do *software* de monitoramento nas visões dos usuários Administrador e Suporte.

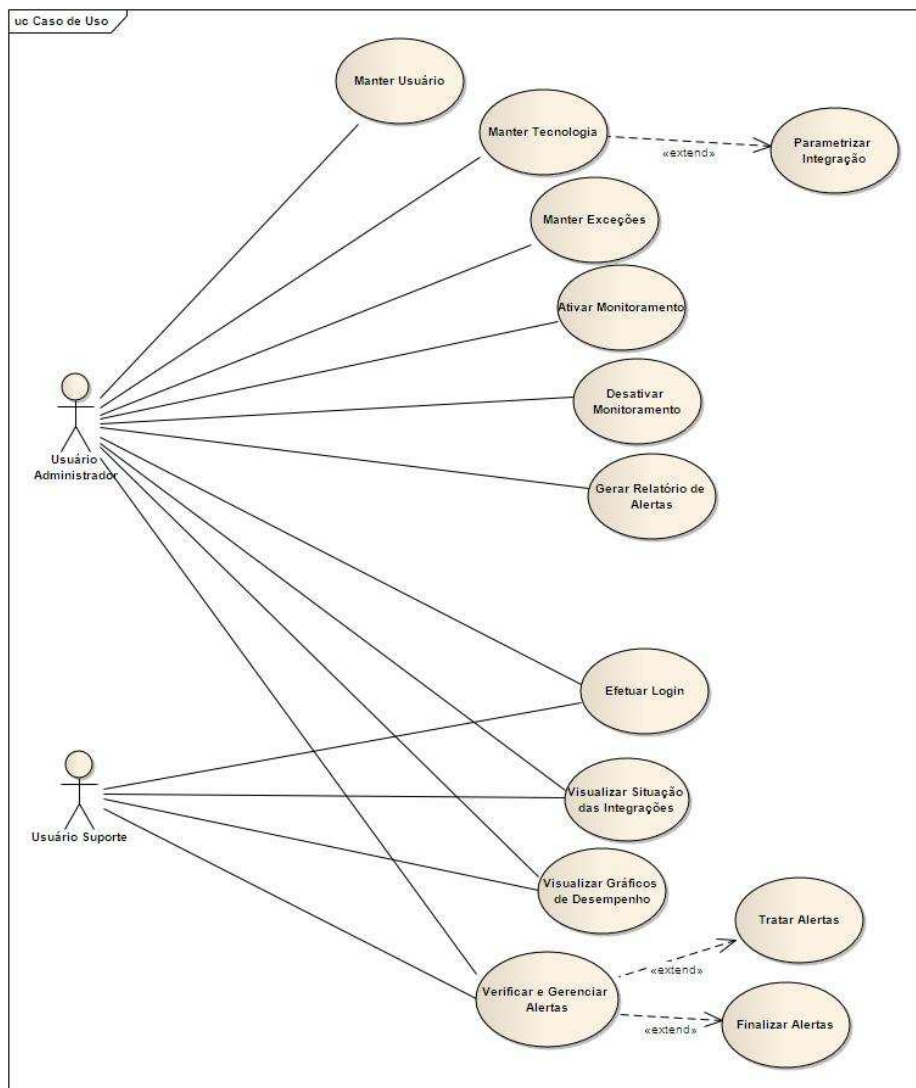


Figura 14: Diagrama de casos de uso.

3.2.4 Descrição dos Casos de Uso

As tabelas listadas a seguir descrevem os casos de uso do diagrama apresentado anteriormente.

Tabela 6: Caso de Uso Efetuar Login.
<p>Caso de Uso: Efetuar <i>Login</i> Ator: Usuário Administrador e Suporte Pré-condição: N/A Descrição: Ao executar o Monitor de Integrações o usuário informa seus dados de <i>login</i> e senha para que este tenha acesso às ferramentas disponíveis no sistema. Fluxo Base: 1. O Sistema apresenta a tela de controle de acesso; 2. O usuário preenche seus dados e confirma; 3. O sistema valida a conta e a senha fornecidas. Pós-condição: Usuário autenticado e autorizado no sistema.</p>
<p>Fluxo Alternativo: Se, no passo 2 do fluxo base, o usuário optar por cancelar a opção então: 1. Termina o processo. Pós-Condição: O caso de uso foi encerrado sem que o usuário seja autenticado.</p>
<p>Fluxo de Exceção: Se, no passo 3 do fluxo base, o sistema não validar os dados de conta e senha fornecidos pelo usuário então: 1. O sistema apresenta a mensagem “Usuário ou senha incorretos”; 2. Retorna ao passo 1 do fluxo base. Pós-condição: N/A.</p>

Tabela 7: Caso de Uso Manter Usuário
<p>Caso de Uso: Manter Usuário Ator: Usuário Administrador Pré-condição: Usuário conectado no sistema. Descrição: O usuário poderá cadastrar, editar ou excluir usuários, sendo estes com permissão de administrador do sistema ou apenas com o perfil de usuário suporte. Fluxo Base:</p> <ol style="list-style-type: none"> 1. No menu “cadastros” da tela principal o usuário seleciona “usuário”; 2. O sistema apresenta em tela o formulário de cadastro de usuários; 3. O usuário seleciona a opção “novo”; 4. O usuário preenche os dados obrigatórios que se referem ao usuário que será cadastrado e confirma; 5. O sistema verifica e insere as informações no banco de dados. <p>Pós-Condição: Caso de uso encerrado com usuário cadastrado conforme perfil selecionado.</p>
<p>Fluxo Alternativo: Se, no passo 2 do fluxo base, o usuário optar por cancelar a opção então:</p> <ol style="list-style-type: none"> 1. Termina o processo. <p>Pós-Condição: O caso de uso é encerrado sem que o usuário esteja cadastrado.</p>
<p>Fluxo de Exceção: Se, no passo 4 do fluxo base, o sistema identificar que o usuário já se encontra cadastrado então:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a mensagem “Já existe este usuário cadastrado”; 2. Retorna ao passo 2 do fluxo base. <p>Pós-condição: N/A. Se, no passo 4 do fluxo base, o sistema identificar a falta de informações obrigatórias para o cadastro do usuário então:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a mensagem “É necessário preencher todos os campos obrigatórios para que o cadastro do usuário seja efetuado.”; 2. Retorna ao passo 2 do fluxo base. <p>Pós-condição: N/A.</p>

Tabela 8: Caso de Uso Manter Tecnologias
<p>Caso de Uso: Manter Tecnologia Ator: Usuário Administrador Pré-condição: Usuário conectado no sistema. O usuário poderá cadastrar, editar ou excluir tecnologias que atualmente oferecem uma interface para integração com o banco de dados da empresa assim como as que futuramente disponibilizarão este serviço. Fluxo Base:</p> <ol style="list-style-type: none"> 1. No menu “cadastros” da tela principal o usuário seleciona “Tecnologias”; 2. O sistema apresenta em tela o formulário de cadastro de tecnologias; 3. O usuário seleciona a opção “novo”; 4. O usuário preenche os dados obrigatórios que se referem à tecnologia que será cadastrada e confirma; 5. O sistema verifica e insere as informações no banco de dados. <p>Pós-condição: Tecnologia cadastrada no banco de dados.</p>
<p>Fluxo Alternativo: Se, no passo 2 do fluxo base, o usuário optar por cancelar a opção então:</p> <ol style="list-style-type: none"> 1. Termina o processo. <p>Pós-Condição: O caso de uso é encerrado sem que a tecnologia fosse cadastrada.</p>
<p>Fluxo de Exceção: Se, no passo 5 do fluxo base, o sistema identificar que a tecnologia já se encontra cadastrada então:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a mensagem “Já existe esta tecnologia cadastrada”; 2. Retorna ao passo 2 do fluxo base. <p>Pós-condição: N/A. Se, no passo 5 do fluxo base, o sistema identificar a falta de informações obrigatórias para o cadastro da tecnologia então:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a mensagem “É necessário preencher todos os campos obrigatórios para que o cadastro da tecnologia seja efetuado.”; 2. Retorna ao passo 2 do fluxo base. <p>Pós-condição: N/A.</p>

Tabela 9: Caso de Uso Manter Exceções
<p>Caso de Uso: Manter Exceções Ator: Usuário Administrador Pré-condição: Usuário conectado ao sistema. Descrição: O usuário poderá cadastrar, editar ou excluir exceções a serem verificadas durante o monitoramento. Fluxo Base:</p> <ol style="list-style-type: none"> 1. No menu “cadastros” da tela principal o usuário seleciona “Exceções”; 2. O sistema apresenta em tela o formulário de cadastro de exceções 3. O usuário seleciona a opção “novo”; 4. O usuário preenche os dados obrigatórios que se referem ao usuário que será cadastrado e confirma; 5. O sistema verifica e insere as informações no banco de dados. <p>Pós-condição: Exceção cadastrada no banco de dados.</p>
<p>Fluxo Alternativo: Se, no passo 2 do fluxo base, o usuário optar por cancelar a opção então:</p> <ol style="list-style-type: none"> 1. Termina o processo. <p>Pós-Condição: O caso de uso é encerrado sem que a exceção fosse cadastrada.</p>
<p>Fluxo de Exceção: Se, no passo 5 do fluxo base, o sistema identificar que a exceção já se encontra cadastrada então:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a mensagem “Já existe esta exceção cadastrada”; 2. Retorna ao passo 2 do fluxo base. <p>Pós-condição: N/A.</p>

Tabela 10: Caso de Uso Parametrizar Integração
<p>Caso de Uso: Parametrizar Integração Ator: Usuário Administrador Pré-condição: Tecnologia Cadastrada. Descrição: Após cadastrar a tecnologia o usuário poderá cadastrar, editar ou excluir os seus parâmetros de integração. Fluxo Base:</p> <ol style="list-style-type: none"> 1. No menu “cadastros” da tela principal o usuário seleciona “Parâmetros”; 2. O sistema apresenta a tela o formulário de cadastro de Parâmetros; 3. O usuário seleciona a opção “novo”; 4. O usuário preenche os dados e confirma; 5. O sistema verifica e insere as informações no banco de dados. <p>Pós-condição: Tecnologia com a integração parametrizada.</p>
<p>Fluxo Alternativo: Se no passo 1 do fluxo base o usuário optar por cancelar a opção então:</p> <ol style="list-style-type: none"> 1. Termina o processo. <p>Pós-Condição: O caso de uso é encerrado sem a integração da tecnologia parametrizada.</p>
<p>Fluxo de Exceção</p> <p>Se, no passo 5 do fluxo base, o sistema identificar a falta de informações obrigatórias para o a parametrização do monitoramento da tecnologia então:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a mensagem “É necessário preencher todos os campos para parametrizar o monitoramento das tecnologias!”; 2. Retorna ao passo 2 do fluxo base. <p>Se, no passo 5 do fluxo base o sistema identificar que já existem parâmetros cadastrados para a tecnologia então:</p> <ol style="list-style-type: none"> 1. O sistema apresenta a mensagem “Já existem parâmetros cadastrados para esta tecnologia!”; 2. Retorna ao passo 2 do fluxo base. <p>Pós-condição: N/A.</p>

Tabela 11: Caso de Uso Ativar Monitoramento
<p>Caso de Uso: Ativar Monitoramento Ator: Usuário Administrador Pré-condição: Tecnologia com integração parametrizada e não monitorada. Descrição: O usuário poderá ativar o monitoramento de integração de uma tecnologia se esta já estiver devidamente parametrizada. Fluxo Base:</p> <ol style="list-style-type: none"> 1. No menu “monitoramento” da tela principal o usuário seleciona “tecnologias”; 2. O sistema apresenta em tela o formulário de ativação/desativação de monitoramento com uma lista de tecnologias, já parametrizadas; 3. O usuário marca a opção para ativar o monitoramento e confirma; 4. O sistema ativa o monitoramento da integração da tecnologia; <p>Pós-condição: Tecnologia com o monitoramento de sua integração ativa.</p>
<p>Fluxo Alternativo: Se no passo 3 do fluxo base o usuário optar por cancelar a opção então:</p> <ol style="list-style-type: none"> 1. Termina o processo. <p>Pós-Condição: O caso de uso é encerrado com o monitoramento da integração da tecnologia se mantendo no estado desativado.</p>

Tabela 12: Caso de Uso Desativar Monitoramento
<p>Caso de Uso: Desativar Monitoramento Ator: Usuário Administrador Pré-condição: Tecnologia com integração parametrizada e o estado de monitoramento ativado. Descrição: O usuário poderá desativar o monitoramento de integração de uma tecnologia quando esta estiver no seu estado ativo. Fluxo Base:</p> <ol style="list-style-type: none"> 1. No menu “monitoramento” da tela principal o usuário seleciona “tecnologias”; 2. O sistema apresenta em tela o formulário de ativação/desativação de monitoramento com uma lista de tecnologias, já parametrizadas; 3. O usuário marca a opção para desativar o monitoramento e confirma; 4. O sistema desativa o monitoramento da integração da tecnologia; <p>Pós-condição: Tecnologia com o monitoramento de sua integração desativa.</p>
<p>Fluxo Alternativo: Se no passo 2 do fluxo base o usuário optar por cancelar a opção então:</p> <ol style="list-style-type: none"> 1. Termina o processo. <p>Pós-Condição: O caso de uso é encerrado com o monitoramento da integração da tecnologia se mantendo no estado ativado.</p>

Tabela 13: Caso de Uso Visualizar Situação das Integrações

Caso de Uso: Visualizar Situação das Integrações

Ator: Usuário Administrador e Suporte

Pré-condição: Monitoramento das integrações de tecnologias ativas.

Descrição: O sistema deverá apresentar em tela a relação de tecnologias que possuem interface de integração com a empresa e que o seu monitoramento esteja ativado. Assim apresentando informações relacionadas a cada pesquisa realizada pela sua integração como, horário em que se executou a pesquisa, total de registros inserido, tempo percorrido durante a execução e velocidade média de inserção.

Fluxo Base:

1. No menu “monitoramento” da tela principal o usuário seleciona “iniciar”;
2. O sistema apresenta em tela o formulário de monitoramento e contendo as tecnologias que possuem o monitoramento de suas integrações ativas na parte superior e a grade de alertas na parte inferior;

Pós-condição: Caso de uso encerrado com o sistema aberto e a relação de tecnologias monitoradas e a lista de alertas apresentados ao usuário.

Fluxo Alternativo:

Se, no passo 2 do fluxo base o sistema não encontrar tecnologias com o monitoramento de integração ativo então:

1. O sistema apresenta a mensagem “Não há monitoramento de integração ativo”;

Pós-condição: Caso de uso encerrado com o sistema aberto sem as tecnologias listadas na tela.

<p>Tabela 14: Caso de Uso Visualizar Gráficos de Desempenho</p> <p>Caso de Uso: Visualizar Gráficos de Desempenho Ator: Usuário Administrador e Suporte Pré-condição: Tecnologias com monitoramento ativo informando os dados de integração na tela. Descrição: Para cada tecnologia haverá a opção para visualizar em gráfico o comparativo entre os resultados adquiridos durante a execução de suas integrações de dados e os valores definidos como parâmetro para esta execução. Fluxo Base: <ol style="list-style-type: none"> 1. Na tela de monitoramento o usuário seleciona a tecnologia monitorada; 2. O usuário clica com o botão direito do mouse sobre a tecnologia selecionada; 3. O sistema apresenta as opções “Visualizar gráfico de registros” e “Visualizar gráfico de velocidade”; 4. O usuário escolhe o tipo de gráfico a ser visualizado e confirma; 5. O sistema apresenta em tela o formulário de gráfico com o resultado das últimas execuções da integração realizadas em gráfico de linha. Pós-condição: Gráfico em tela apresentando dados referentes às últimas execuções da integração realizadas nos últimos trinta minutos efetuando um comparativo entre estes resultados apresentados e os parâmetros pré-definidos para a execução do processo.</p> <hr/> <p>Fluxo Alternativo: Se no passo 4 do fluxo base o usuário optar por cancelar o processo então: <ol style="list-style-type: none"> 1. Termina o processo. Pós-condição: Processo finalizado sem que o usuário tenha visualizado as informações da integração em gráfico.</p>

1. Na tela de monitoramento o usuário seleciona a tecnologia monitorada;
2. O usuário clica com o botão direito do mouse sobre a tecnologia selecionada;
3. O sistema apresenta as opções “Visualizar gráfico de registros” e “Visualizar gráfico de velocidade”;
4. O usuário escolhe o tipo de gráfico a ser visualizado e confirma;
5. O sistema apresenta em tela o formulário de gráfico com o resultado das últimas execuções da integração realizadas em gráfico de linha.

1. Termina o processo.

Tabela 15: Caso de Uso Verificar e Gerenciar Alertas

Caso de Uso: Verificar e Gerenciar Alertas

Ator: Usuário Administrador e Suporte

Pré-condição: O resultado das execuções da integração estar abaixo do desempenho esperado de acordo com os valores parametrizados.

Descrição: Caso uma integração não estiver executando seus processos de pesquisa ou suas pesquisas obtiverem resultados abaixo do esperado, de acordo com os parâmetros pré-definidos, o sistema deverá emitir um alerta informando ao usuário o ocorrido para que este verifique.

Fluxo Base:

1. O sistema apresenta o alerta de exceção na grade de alertas da tela de monitoramento;
2. O usuário seleciona a mensagem de alerta na grade;
3. O sistema apresenta ao lado da grade as informações referentes à mensagem de alerta;
4. O usuário clica em “Verificar”;
5. O sistema remove a mensagem de alerta da grade;

Pós-condição: A mensagem de alerta passa a ser visualizada na tela de Alertas em Tratamento.

Fluxo Alternativo:

Se no passo 4 do fluxo base o usuário não clicar em “verificar” então:

1. O sistema manterá a mensagem de alerta na grade;
2. Termina o processo.

Pós-condição: O alerta permanecerá na tela até que seja efetuada sua leitura para averiguação do ocorrido.

Tabela 16: Caso de Uso Tratar Alertas

Caso de Uso: Tratar Alertas

Ator: Usuário Administrador e Suporte

Pré-condição: Efetuada leitura do alerta pelo usuário.

Descrição: Após o usuário efetuar a leitura do alerta o sistema passa a apresentar a mensagem de alerta na tela de Alertas em Tratamento.

Fluxo Base:

1. O sistema apresenta o alerta de exceção na grade da tela de Alertas em Tratamento;
2. O usuário seleciona a mensagem de alerta na grade;
3. O sistema apresenta ao lado da grade as informações referentes à mensagem de alerta, o usuário que efetuou leitura e a hora em que a mensagem foi lida, além de apresentar os procedimentos a serem seguidos e o campo para comentários do usuário;
4. O usuário insere comentários ao passo que efetua os procedimentos e clica em "Add Info";
5. O sistema insere as informações no banco de dados;

Pós-condição: Informações adicionadas pelo usuário ao alerta em tratamento.

Fluxo Alternativo:

Se no passo 4 do fluxo base usuário optar por não adicionar as informações então:

1. Termina o processo.

Pós-condição: Alerta em tratamento sem informações adicionadas pelo usuário.

Tabela 17: Caso de Uso Finalizar Alertas

Caso de Uso: Finalizar Alertas

Ator: Usuário Administrador e Suporte

Pré-condição: Alerta estabilizado.

Descrição: Após o sistema identificar a estabilidade da exceção passa a apresentar a mensagem de alerta na tela de Alertas Estabilizados.

Fluxo Base:

1. O sistema apresenta o alerta de exceção na grade da tela de Alertas Estabilizados;
2. O usuário seleciona a mensagem de alerta na grade;
3. O sistema apresenta ao lado da grade as informações referentes à mensagem de alerta, o usuário que efetuou leitura e a hora em que a mensagem foi lida, além de apresentar os procedimentos a serem seguidos, o campo para comentários do usuário e o horário de estabilidade da exceção;
4. O usuário insere clica em "Finalizar";
5. O sistema remove a mensagem de alerta da grade;

Pós-condição: A mensagem de alerta passa a não ser mais visualizada no sistema.

Fluxo Alternativo:

Se no passo 4 do fluxo base usuário optar por não finalizar o alerta então:

1. Termina o processo.

Pós-condição: Alerta estabilizado sendo visualizado na grade.

Tabela 18: Caso de Uso Gerar Relatórios de Alertas

Caso de Uso: Gerar Relatório de Alertas

Ator: Usuário Administrador

Pré-condição: Armazenamento em banco de dados das informações colhidas durante o monitoramento das integrações.

Descrição: O usuário poderá gerar relatórios, referente aos alertas gerados, em datas anteriores definidas por períodos.

Fluxo Base:

1. O usuário seleciona o menu “relatórios” da tela principal;
2. O sistema apresenta as opções “Alertas não verificados”, “Alertas Verificados” e “Gráficos”;
3. O usuário escolhe uma das opções;
4. O sistema apresenta em tela o formulário de relatório;
5. O usuário define as datas do período em que deve ser realizada a pesquisa;
6. O usuário clica em “pesquisar” e confirma;
7. O sistema apresenta em tela o relatório.

Pós-condição: Informações de alertas extraídas pelo sistema de acordo com o período estipulado pelo usuário.

Fluxo Alternativo:

Se no passo 6 do fluxo base o usuário optar por cancelar o processo então:

1. Termina o processo.

Pós-condição: Processo finalizado sem a extração de informações de alertas no relatório.

3.2.5 Diagrama de Classes do Sistema

O diagrama de classes do sistema descreve a estrutura que compõem o sistema a Figura 15 apresenta a estrutura do *software* de monitoramento.

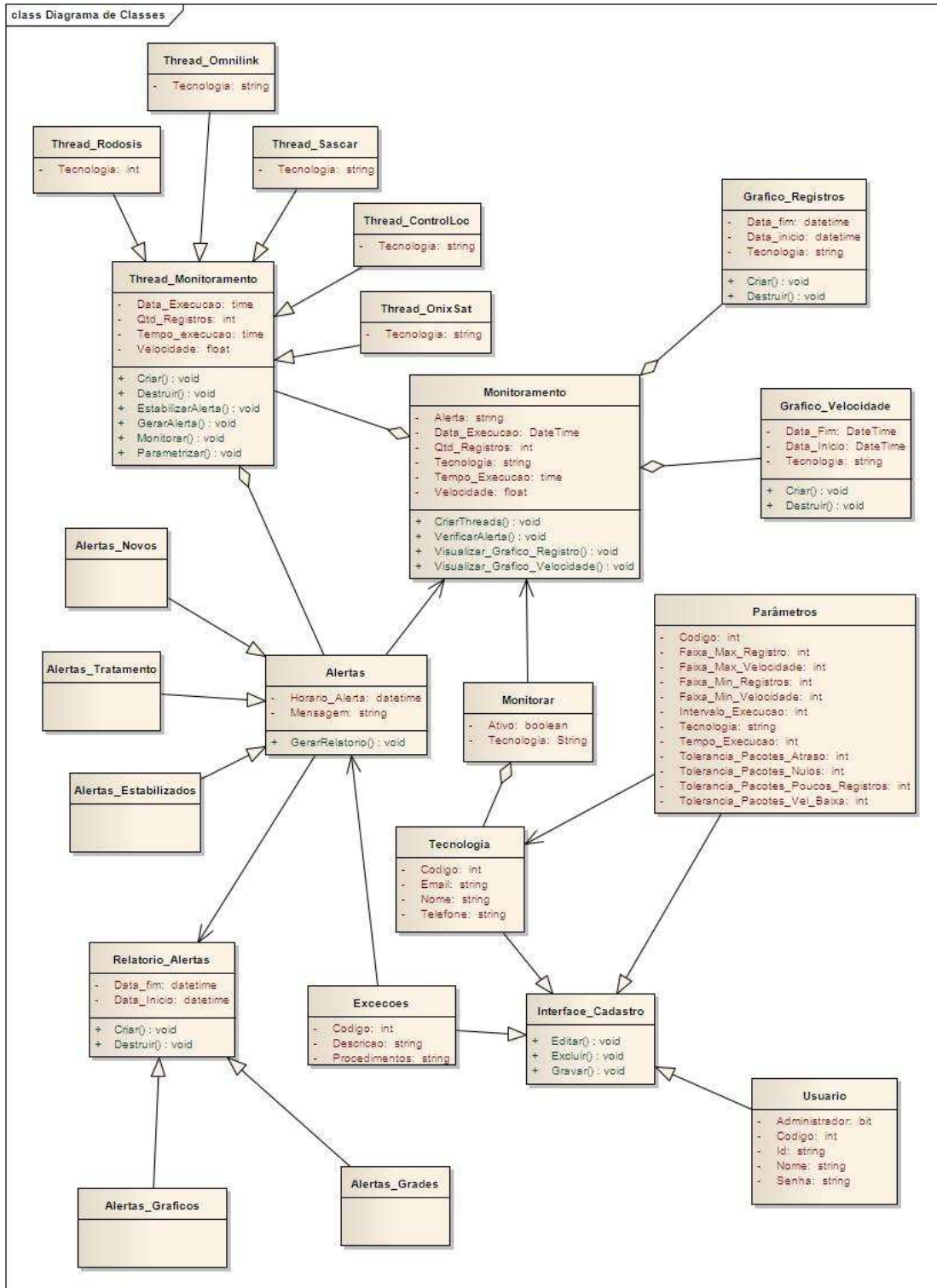


Figura 15: Diagrama de Classes do Sistema.

3.2.6 Diagrama de Entidade-Relacionamento do Banco de Dados

O diagrama de entidade-relacionamento descreve a estrutura e a forma como são armazenadas as informações em um banco de dados. A Figura 16 apresenta a estrutura do banco de dados utilizado pelo *software* de monitoramento.

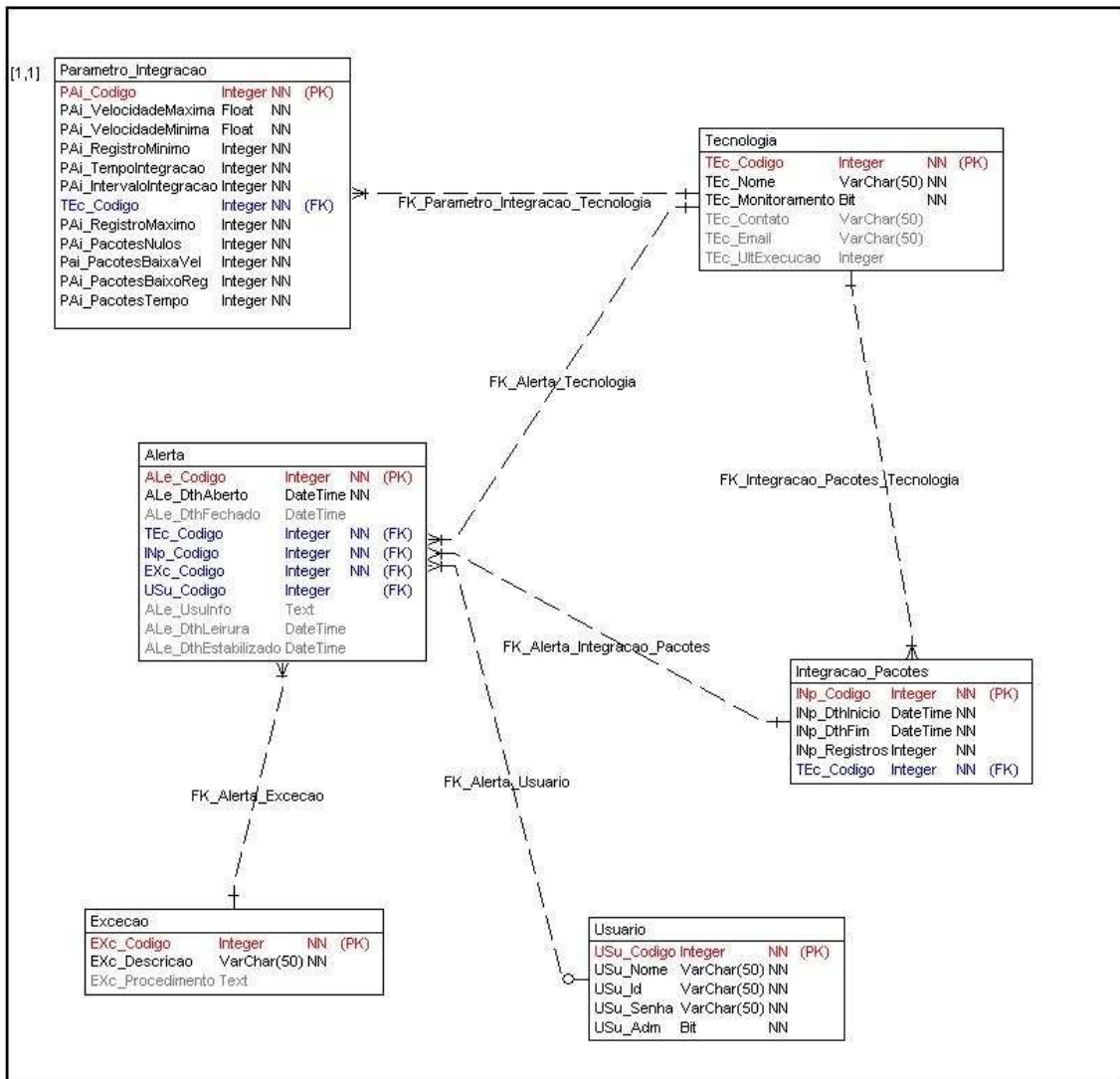


Figura 16: Diagrama de Entidade-Relacionamento do Banco de Dados.

3.2.7 Diagramas de Seqüência do Sistema

O diagrama de seqüência do sistema apresenta o processo realizado através do *software* informando de maneira seqüencial as tarefas executadas. A Figura 17 apresenta a seqüência de tarefas executadas para efetuar o login no sistema.

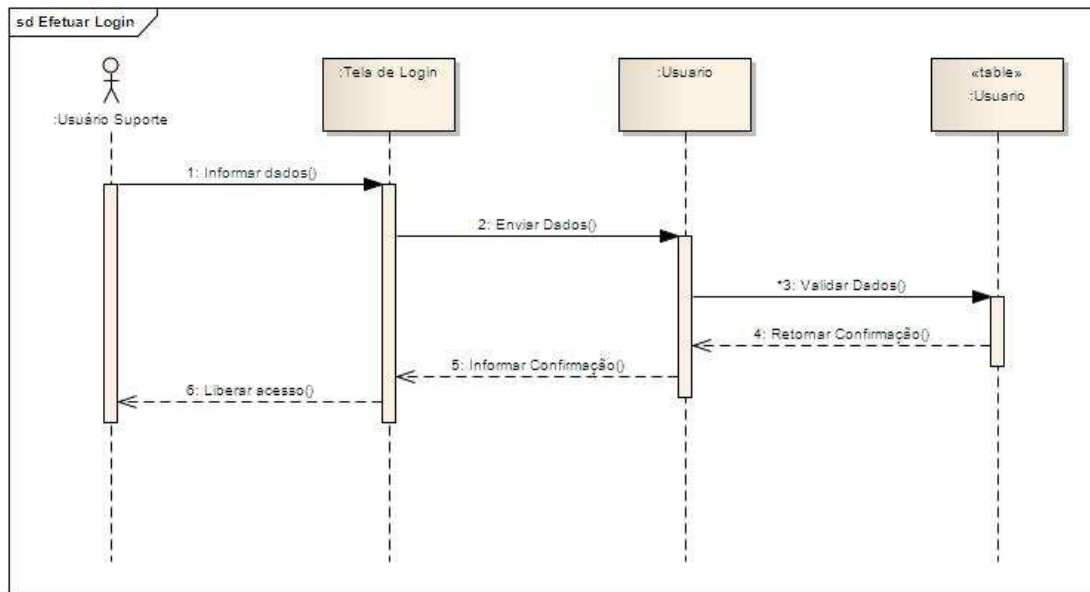


Figura 17: Diagrama de Seqüência Efetuar Login.

A Figura 18 apresenta a seqüência de tarefas executadas para efetuar o cadastro de usuário.

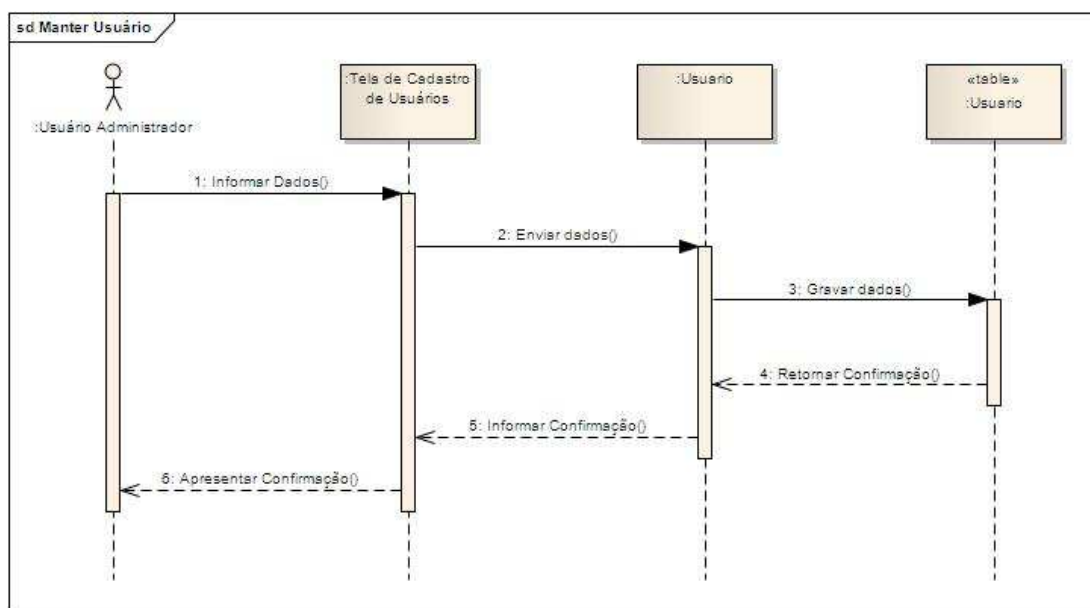


Figura 18: Diagrama de Seqüência Manter Usuário.

A Figura 19 apresenta a seqüência de tarefas executadas para efetuar o cadastro de tecnologia.

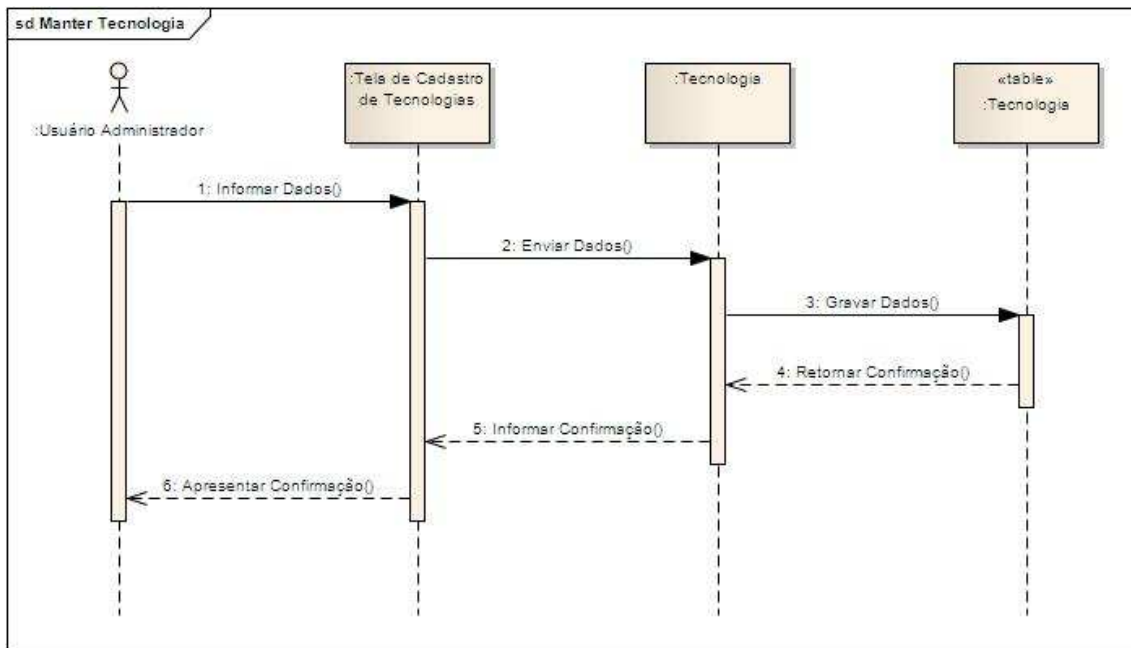


Figura 19: Diagrama de Seqüência Manter Tecnologia.

A Figura 20 apresenta a seqüência de tarefas executadas para efetuar o cadastro de exceções.

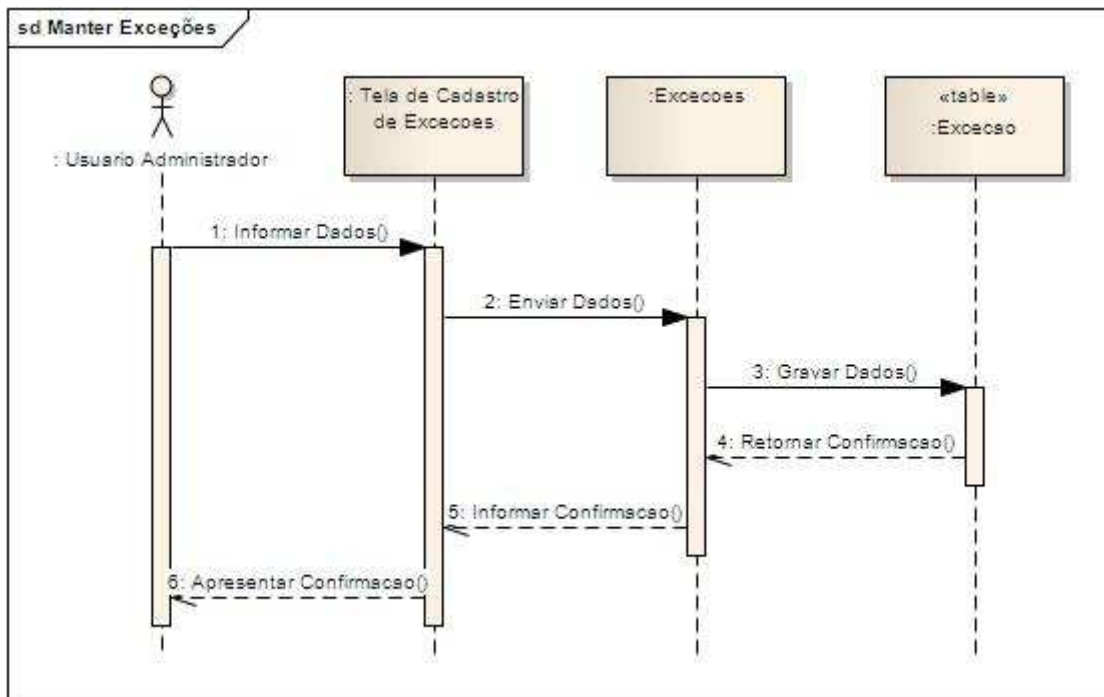


Figura 20: Diagrama de Seqüência Manter Exceções.

A Figura 21 apresenta a seqüência de tarefas executadas para efetuar a parametrização das integrações.

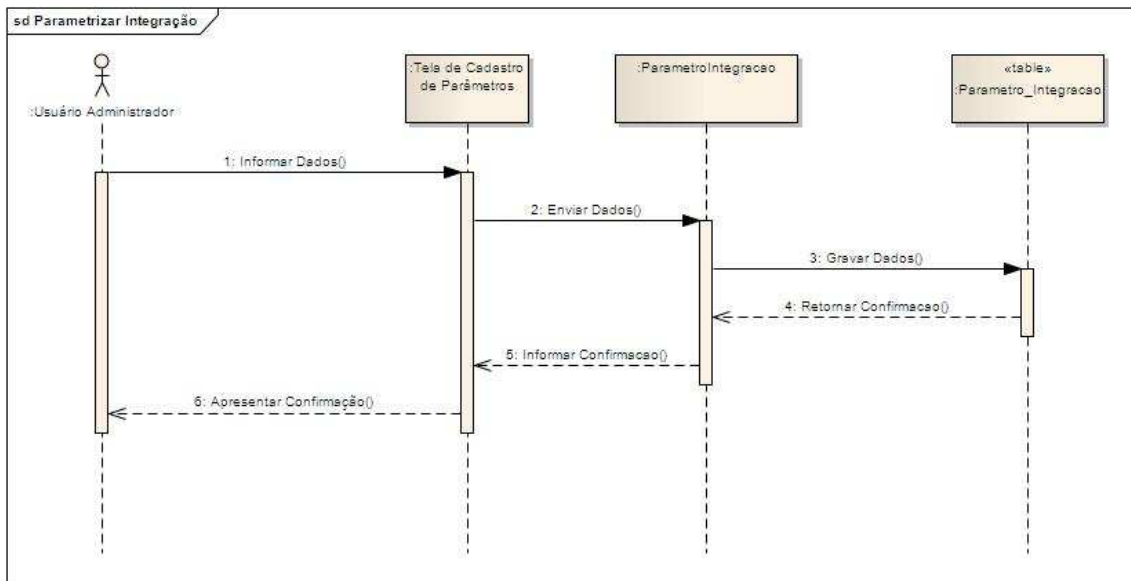


Figura 21: Diagrama de Seqüência Parametrizar Integração.

A Figura 22 apresenta a seqüência de tarefas executadas para efetuar a ativação do monitoramento das integrações.

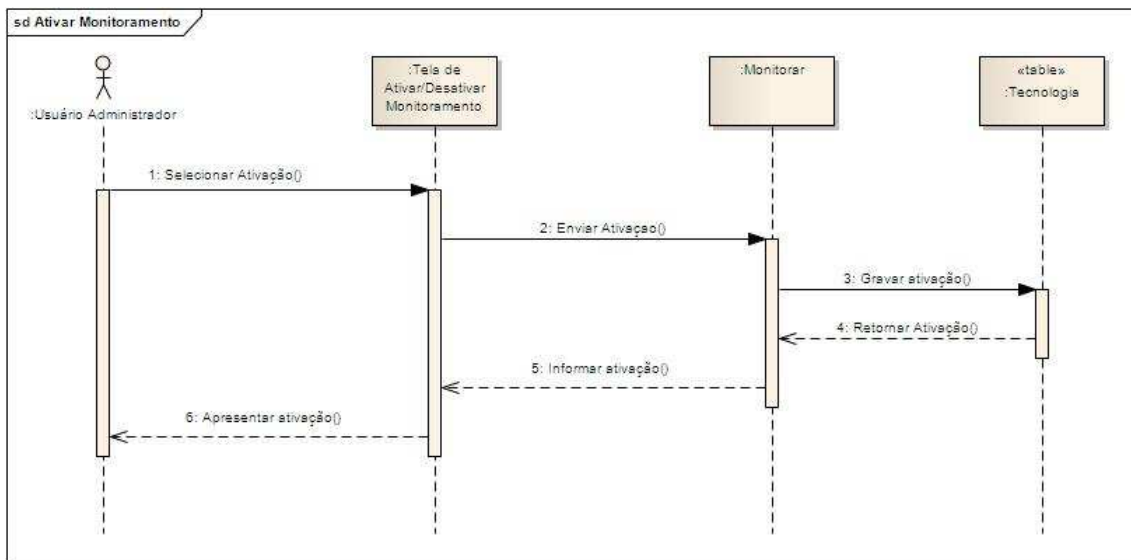


Figura 22: Diagrama de Seqüência Ativar Monitoramento.

A Figura 23 apresenta a seqüência de tarefas executadas para efetuar a desativação do monitoramento das integrações.

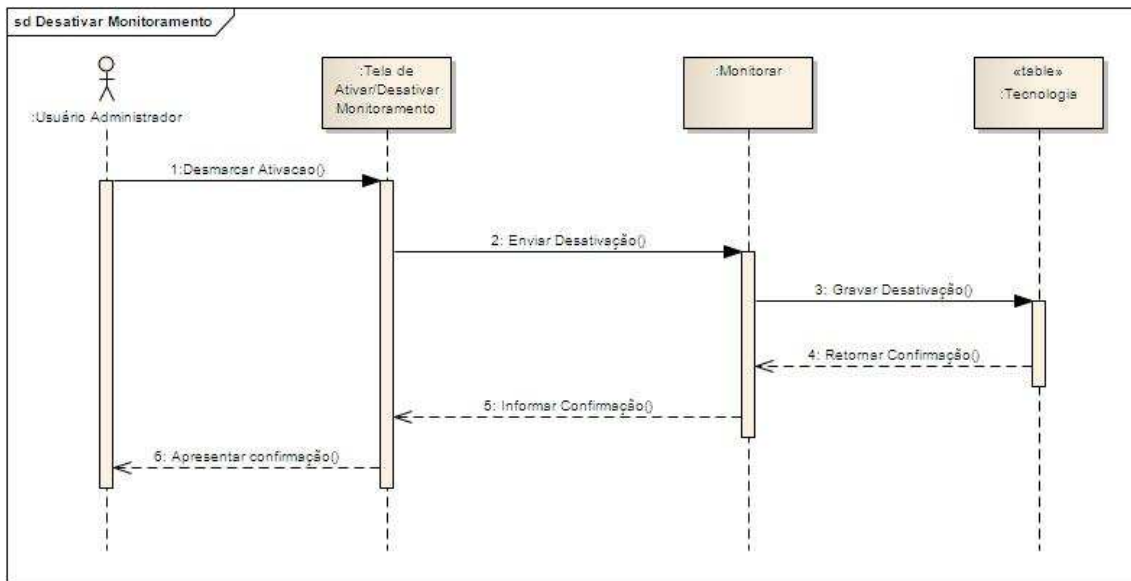


Figura 23: Diagrama de Seqüência Desativar Monitoramento.

A Figura 24 apresenta a seqüência de tarefas executadas para efetuar a visualização do gráfico de desempenho.

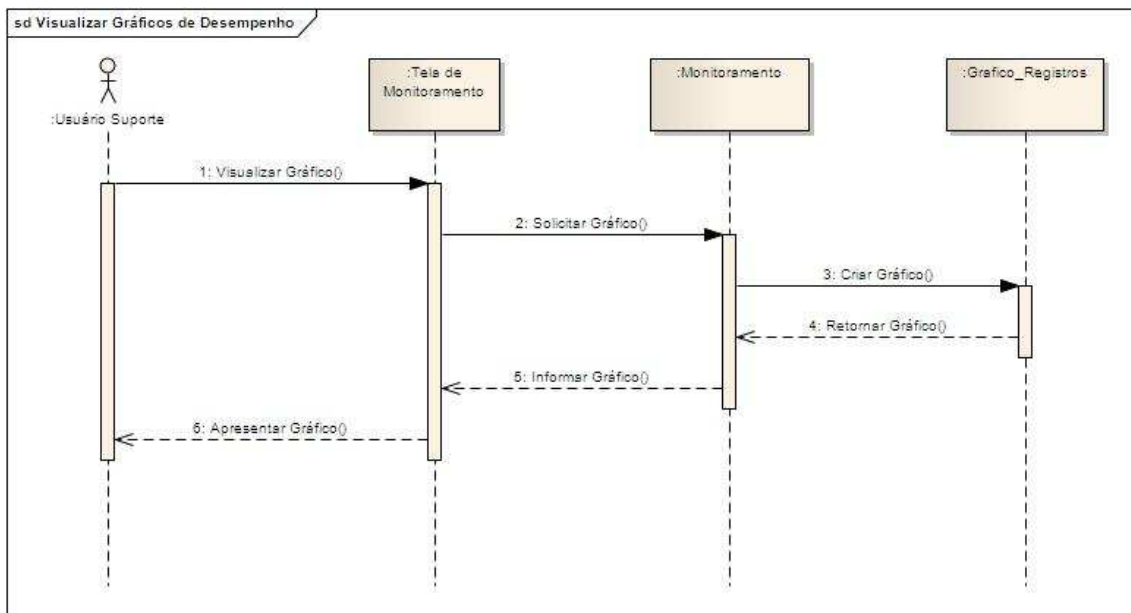


Figura 24: Diagrama de Seqüência Gráficos de Desempenho.

A Figura 25 apresenta a seqüência de tarefas executadas para efetuar a visualização da situação das integrações.

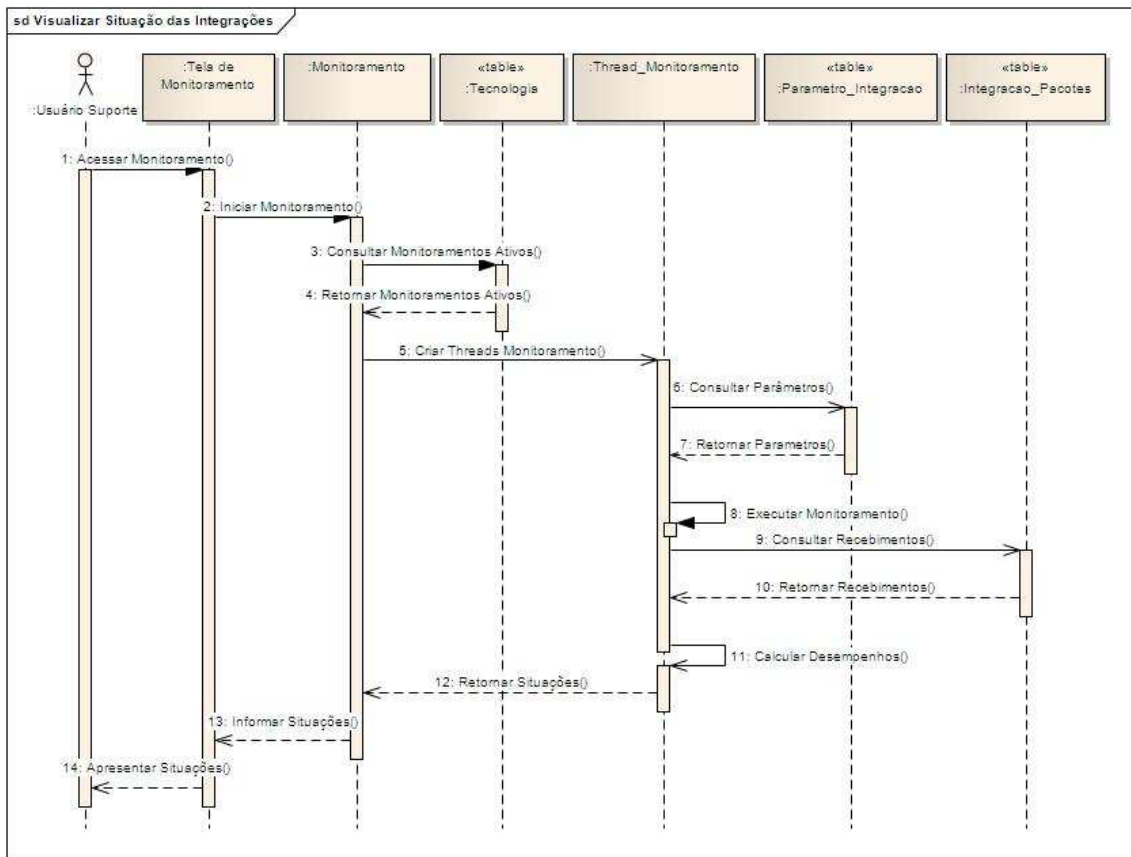


Figura 25: Diagrama de Seqüência Situação das Integrações.

A Figura 26 apresenta a seqüência de tarefas executadas para efetuar visualização de alertas novos.

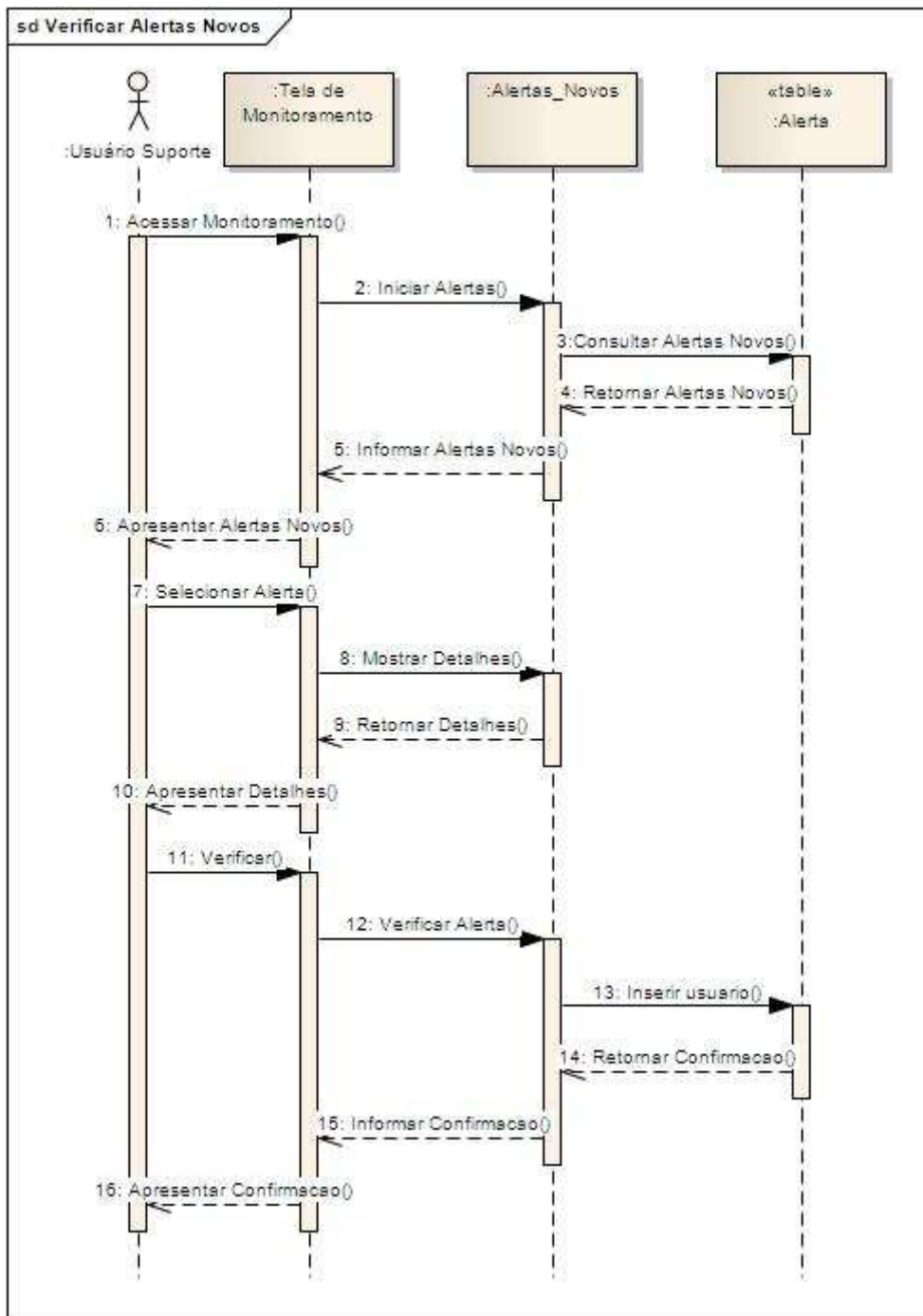


Figura 26: Diagrama de Seqüência Verificar Alertas Novos.

A Figura 27 apresenta a seqüência de tarefas executadas para efetuar a visualização de alertas em tratamento.

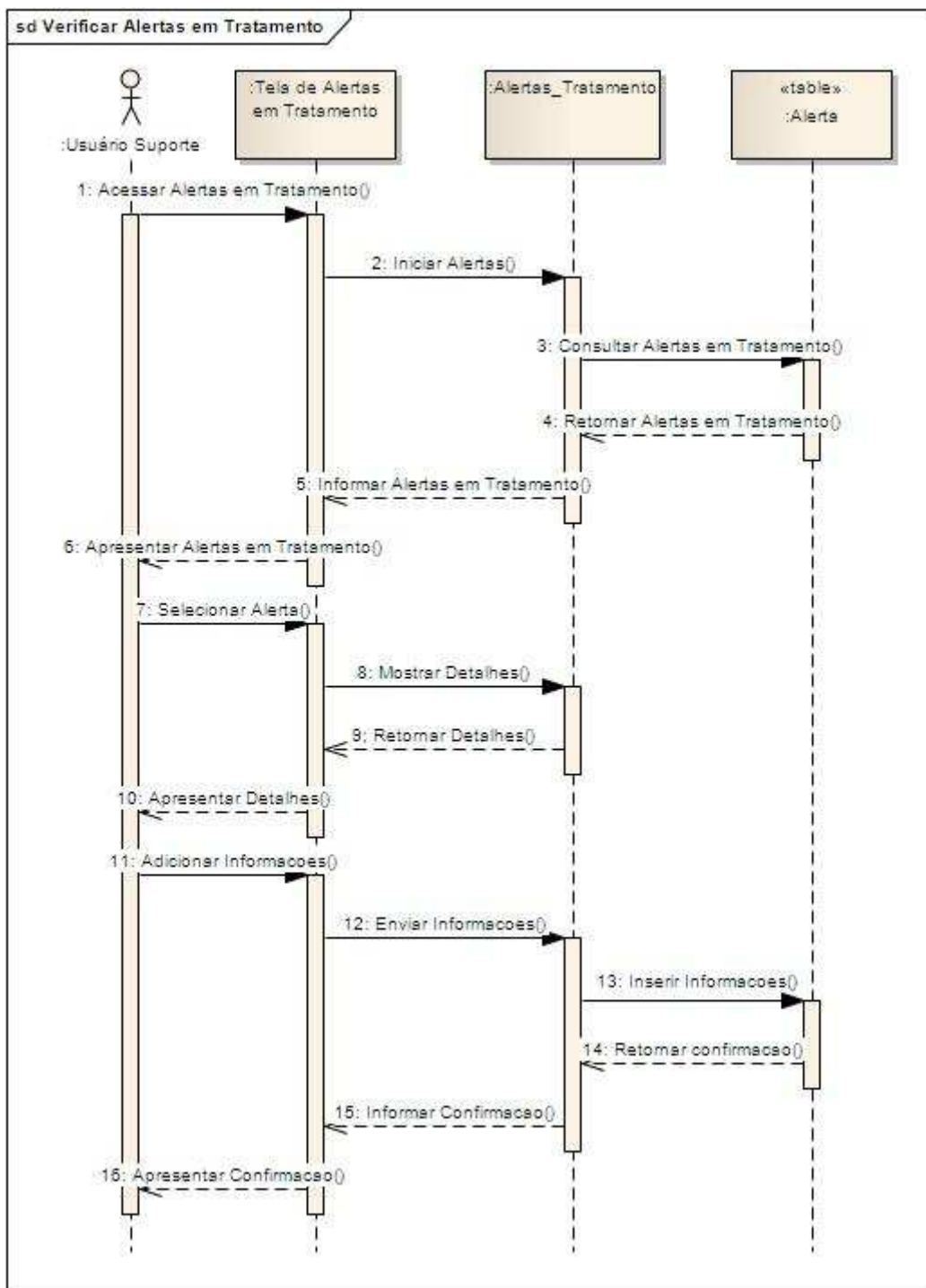


Figura 27: Diagrama de Seqüência Verificar Alertas em Tratamento.

A Figura 28 apresenta a seqüência de tarefas executadas para efetuar a visualização de alertas estabilizados.

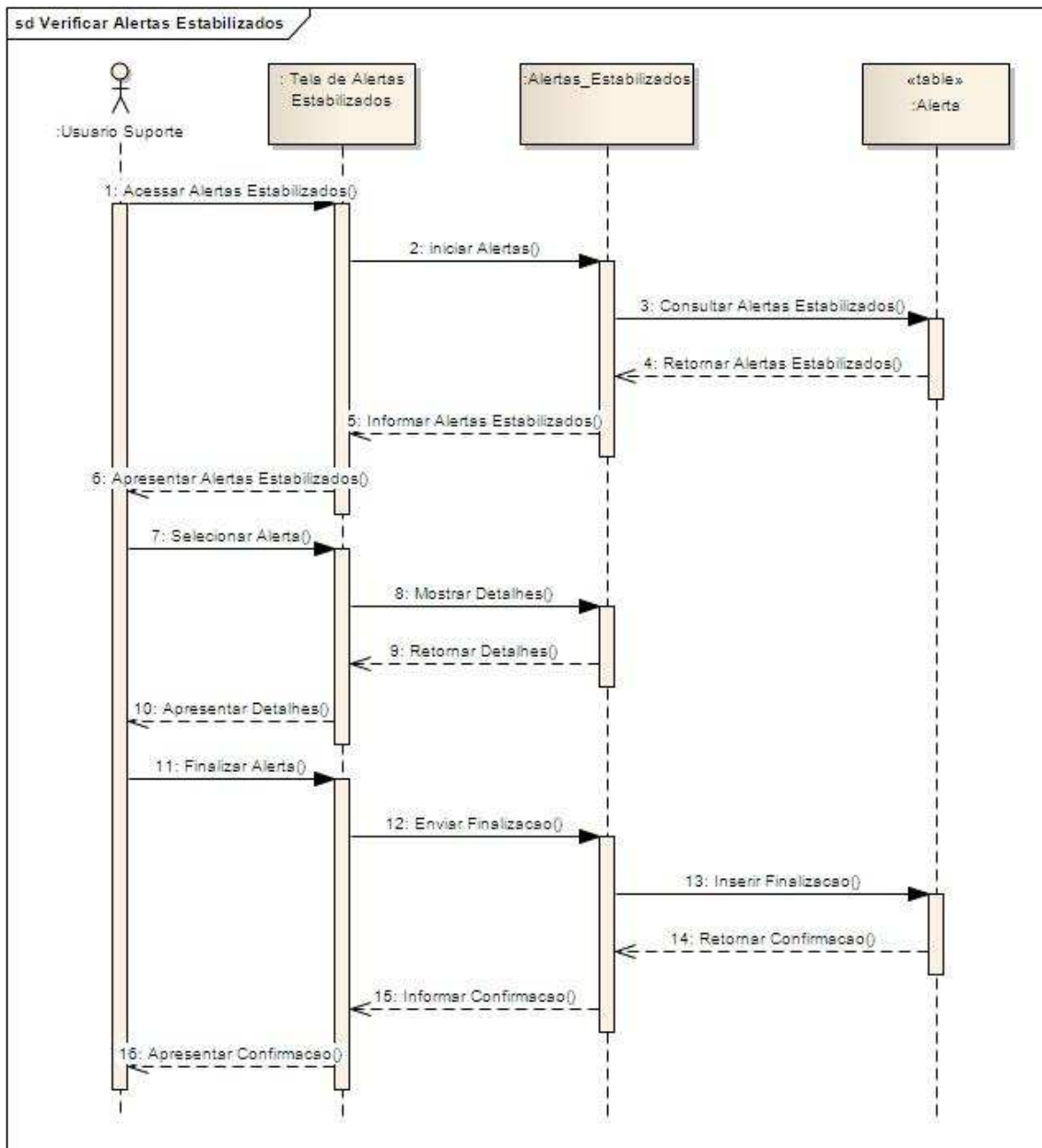


Figura 28: Diagrama de Seqüência Verificar Alertas Estabilizados.

A Figura 29 apresenta a seqüência de tarefas executadas para gerar relatório de alertas.

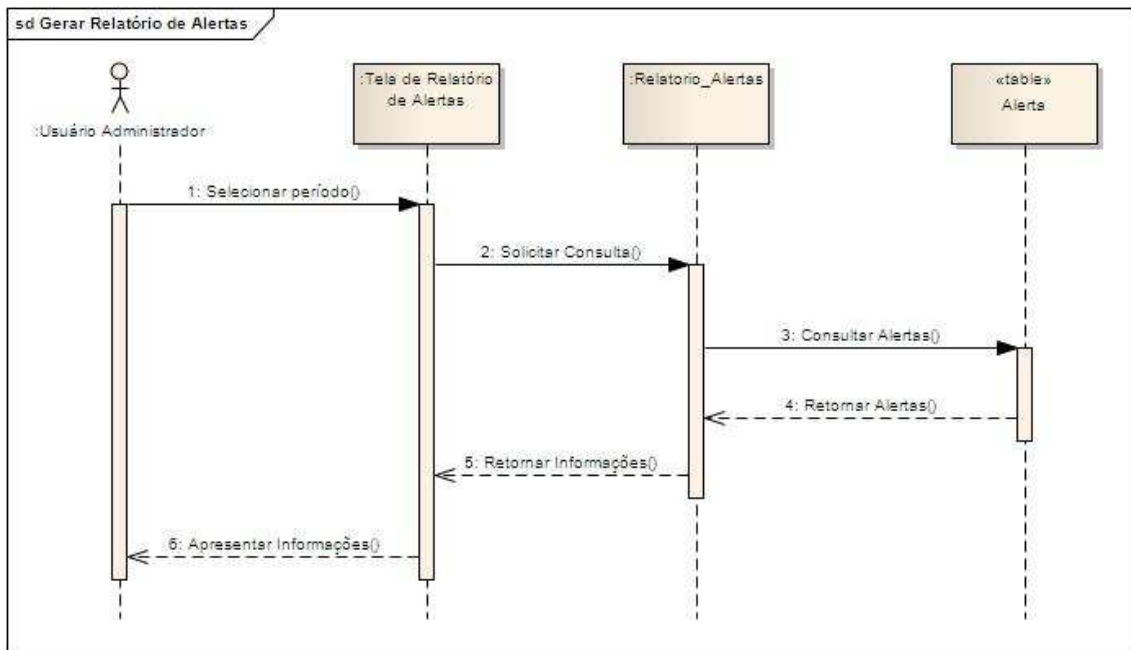


Figura 29: Diagrama de Seqüência Gerar Relatórios de Alertas.

3.2.8 Interface do Sistema

Neste ítem serão apresentadas as telas do sistema envolvendo o módulo administrador e o módulo suporte. O que difere os dois módulos é a liberação de cadastros e relatórios para o usuário administrador.

A Figura 30 representa a tela inicial do sistema, ou tela de login, onde se permite os usuários cadastrados a terem acesso ao sistema de acordo com o seu perfil (administrador ou suporte).



Figura 30: Tela de Login.

A Figura 31 apresenta a tela principal do sistema no módulo administrador, onde são disponibilizadas todas as opções do sistema para o usuário, os menus Cadastros, Relatórios e Monitoramento.

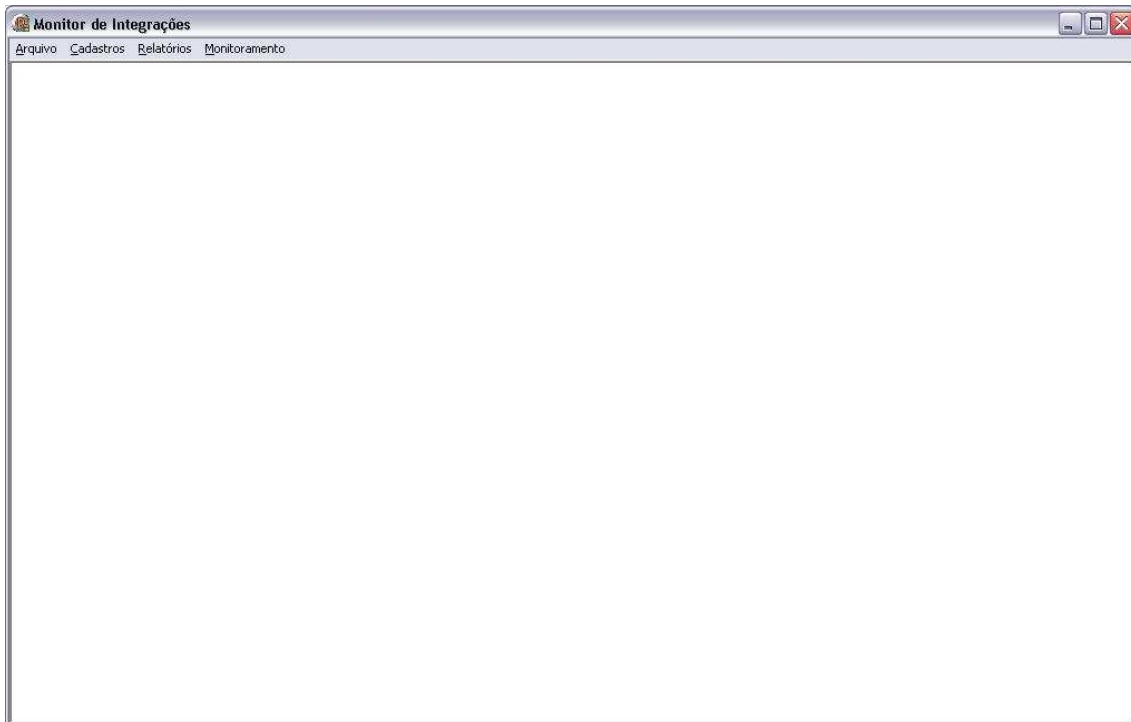


Figura 31: Tela Principal.

A Figura 32 é o cadastro de usuários, disponibilizando ao usuário administrador a criação de novos usuários que tenham permissão para utilizar o sistema.

The image shows a window titled 'Cadastro de Usuários'. The menu bar includes 'Primeiro', 'Anterior', 'Próximo', 'Último', 'Novo', 'Gravar', 'Excluir', 'Procurar', 'Cancelar', and 'Sair'. The form contains the following fields:

- Código: 15
- Nome: Bruce Banner
- Login: hulk Admin
- Senha: ••••••
- Confirm: (empty)

Figura 32: Tela de Cadastro de Usuários.

A Figura 33 é o cadastro de tecnologias, que permite ao usuário administrador inserir novas tecnologias que possuem interface de integração com a empresa.

The image shows a window titled 'Cadastro de Tecnologias'. The menu bar includes 'Primeiro', 'Anterior', 'Próximo', 'Último', 'Novo', 'Gravar', 'Excluir', 'Procurar', 'Cancelar', and 'Sair'. The form contains the following fields:

- Código: 5
- Nome: OnixSat
- Telefone: 11-3256-0087
- Email: suporte@onixsat.com.br

Figura 33: Tela de Cadastro de Tecnologia.

A Tela Figura 34 é o cadastro de parâmetros, que permite ao usuário administrador definir os parâmetros da integração ideal para cada tecnologia.



Figura 34: Tela de Cadastro de Parâmetros.

A Figura 35 apresenta o cadastro de exceções, permitindo ao usuário administrador cadastrar exceções que venham a ocorrer quando os resultados de suas integrações não estiverem de acordo com os parâmetros cadastrados para a tecnologia, e os procedimentos a serem realizados para estabilizar o recebimento de informações.

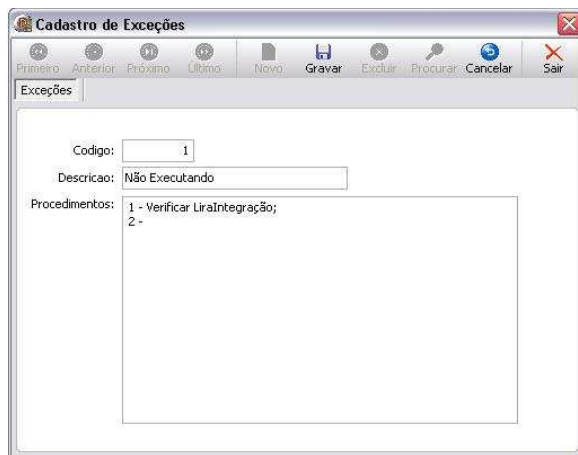


Figura 35: Tela de Cadastro de Exceções.

A Figura 36 mostra as tecnologias monitoradas pelo sistema. Nesta tela é possível que o usuário administrador ative ou desative o monitoramento de integração de uma tecnologia escolhida por ele.

Código	Nome	Monitorado
9	ControlLoc	<input checked="" type="checkbox"/>
17	Omnalink	<input checked="" type="checkbox"/>
5	OnixSat	<input checked="" type="checkbox"/>
3	Rodosis	<input checked="" type="checkbox"/>
14	Sascar	<input checked="" type="checkbox"/>

Figura 36: Tela de Tecnologias Monitoradas.

A Figura 37 apresenta o monitoramento onde o usuário administrador ou suporte pode verificar:

1. Situação das integrações monitoradas pelo sistema;
2. Novos alertas gerados pelo sistema;
3. Detalhes do alerta selecionado para que se possa verificar a exceção ocorrida;
4. Alertas em tratamento e alertas estabilizados.

Monitoramento					
Tecnologia:	ControlLoc	Tecnologia:	Omnalink	Tecnologia:	OnixSat
Último Pacote recebido:	04/06/2010 16:15:20	Último Pacote recebido:	04/06/2010 16:15:27	Último Pacote recebido:	04/06/2010 16:14:50
Nº de Registros:	68	Nº de Registros:	27	Nº de Registros:	496
Duração:	00:00:02.000	Duração:	00:00:02.000	Duração:	00:00:27.000
Velocidade:	34,00 reg/s	Velocidade:	13,50 reg/s	Velocidade:	18,37 reg/s
Tecnologia:	Rodosis	Tecnologia:	Sascar	1	
Último Pacote recebido:	04/06/2010 16:15:24	Último Pacote recebido:	04/06/2010 16:15:25		
Nº de Registros:	4	Nº de Registros:	69		
Duração:	00:00:01.000	Duração:	00:00:06.000		
Velocidade:	04,00 reg/s	Velocidade:	11,50 reg/s		
Alerta		Tecnologia			
Poucos Registros		Rodosis			
2					
Código:		1744			
Alerta:		Poucos Registros			
Tecnologia:		Rodosis			
Integração:		11191			
Data Alerta:		4/6/2010 16:13:47			
Procedimentos:		1 - Entrar em contato com a Tecnologia.			
3					
Verificar					
Alertas em Tratamento: 1		Alertas Estabilizados: 0		4	

Figura 37: Tela de Monitoramento.

A Figura 38 exibe os Alertas em tratamento, onde o usuário administrador ou suporte pode verificar os alertas que ainda não se encontram estabilizados e que

estão em verificação por ele, permitindo a inclusão de informações referentes à sua atuação na busca da estabilidade do processo.

Alerta	Tecnologia
Atraso	OnixSat

Código: 1695
 Alerta: Atraso
 Tecnologia: OnixSat
 Integração: 9609
 Data Alerta: 3/6/2010 18:55:59

Procedimentos:
 1 - Verificar quantidade de registros recebidos por pacote;

Usuário: Raphael Silveira Leitura: 3/6/2010 21:10:18

Informações do Usuário:
 A média de registros recebidos por pacote nas últimas integrações variam entre 200 e 300 registros com uma velocidade média de inserção de 10 reg/segundo.
 O que indica que o atraso está relacionado ao envio de informações pela tecnologia, pois normalmente este número de registros por pacote levam em média 15 segundos para serem inseridos.

Add Info

Figura 38: Tela de Alertas em Tratamento.

A Figura 39 representa os alertas estabilizados, onde o usuário administrador ou suporte poderá visualizar os alertas em que foram identificadas a estabilidade no processo de integração, permitindo ao usuário inserir suas considerações finais após sua atuação para normalizar o processo e finalizar o alerta.

Alerta	Tecnologia
Poucos Registros	Rodosis

Código: 1744
 Alerta: Poucos Registros
 Tecnologia: Rodosis
 Integração: 11191
 Data Alerta: 4/6/2010 16:13:47 Estabilizado: 4/6/2010 16:17:55

Procedimentos:
 1 - Entrar em contato com a Tecnologia.

Usuário: Raphael Silveira Leitura: 4/6/2010 16:16:01

Informações do Usuário:
 Entrado em contato com a tecnologia, onde fomos informados que o sistema deles está passando por oscilações de comunicação.
 Número de registro por pacotes retornou ao normal. Não há mais oscilação no sistema.

Add Info Finalizar

Figura 39: Tela de Alertas Estabilizados.

A Figura 40 apresenta o gráfico de registros inseridos por pacote integrado nos últimos trinta minutos onde o usuário administrador ou suporte escolhe na tela de monitoramento a tecnologia a ser visualizada, tendo como resultado em gráfico de linha as quantidades de registros adquiridos nas últimas integrações realizadas e a faixa de quantidade máxima e mínima registros pintada no gráfico.

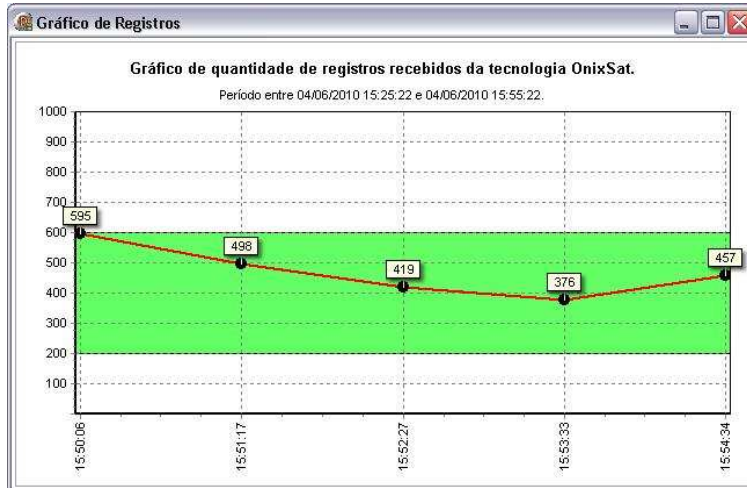


Figura 40: Tela de Gráfico de Registros.

A Figura 41 apresenta o gráfico de velocidade de inserção de registros por pacote integrado nos últimos trinta minutos, onde o usuário administrador ou suporte escolhe na tela de monitoramento a tecnologia a ser visualizada, tendo como resultado em gráfico de linha as velocidades de inserção nas últimas integrações realizadas e a faixa de velocidade máxima e mínima de inserção de registros pintada no gráfico.

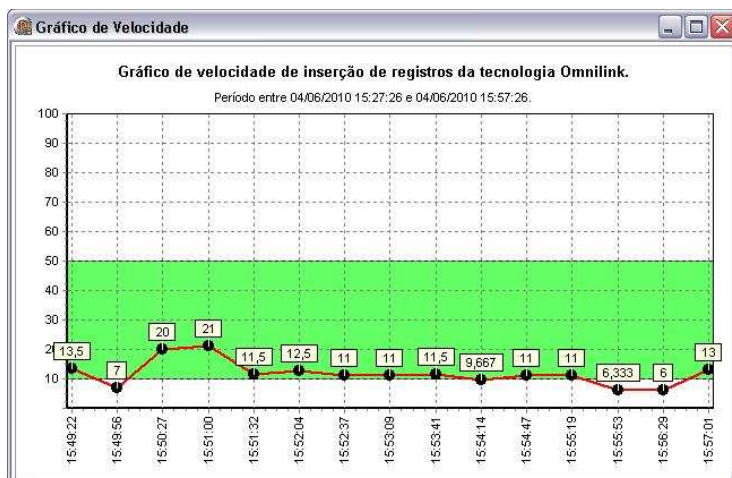


Figura 41: Tela de Gráfico de Velocidade.

A Figura 42 exibe o relatório gráfico de alertas, onde o usuário administrador seleciona o período de pesquisa inserindo a data de início e de fim para realizar a

consulta, tendo como resultado todos os alertas ocorridos no período e suas representações percentuais apresentadas em um gráfico de pizza.

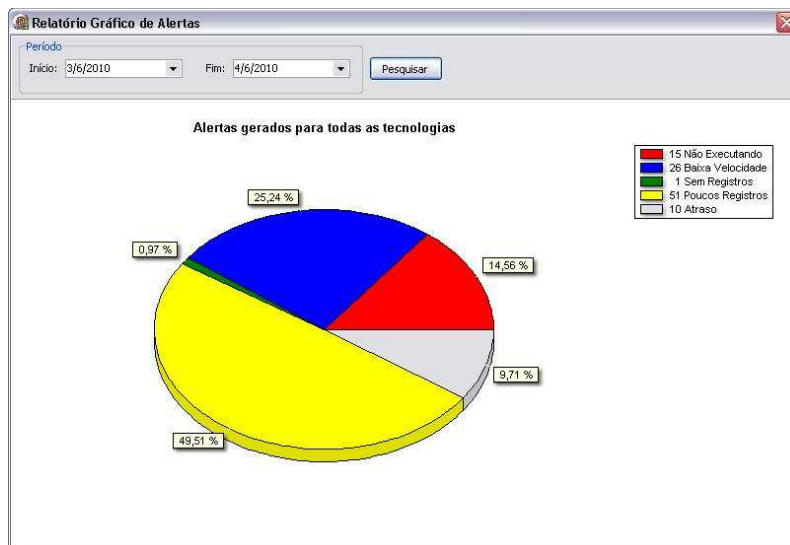


Figura 42: Tela de Relatório Gráfico de Alertas.

A Figura 43 exibe o relatório gráfico de alertas por tecnologia, onde o usuário administrador seleciona o período de pesquisa inserindo a data de início e de fim e define a tecnologia a ser consultada, tendo como resultado todos os alertas ocorridos no período para a tecnologia escolhida e suas representações percentuais apresentadas em um gráfico de pizza.

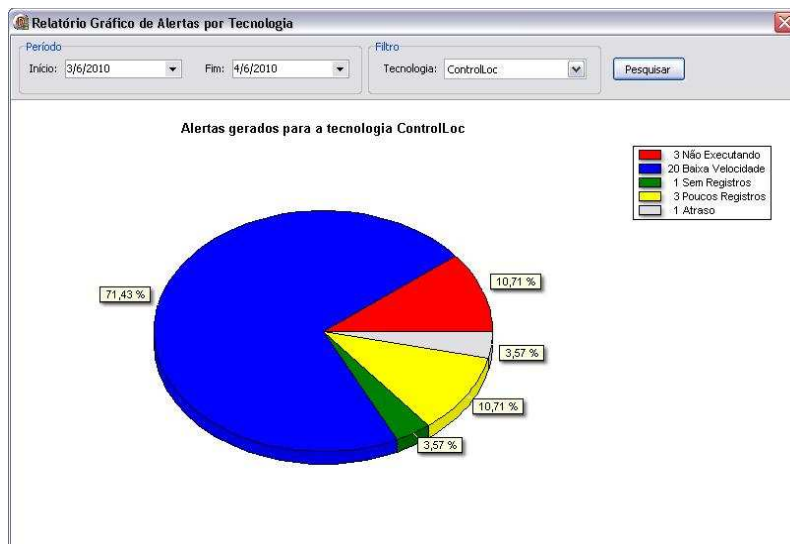


Figura 43: Tela de Relatório Gráfico de Alertas por Tecnologia.

A Figura 44 exibe o relatório gráfico de alertas verificados e alertas não verificados, onde o usuário administrador seleciona o período de pesquisa inserindo a data de início e de fim e define a tecnologia a ser consultada, tendo como

resultado a quantificação de alertas verificados por usuários e alertas não verificados e suas respectivas representações percentuais apresentadas em gráfico de pizza.



Figura 44: Tela de Relatório Gráfico de Alertas Verificados x Não Verificados.

A Figura 45 representa o relatório de alertas não verificados. Nela o usuário administrador seleciona o período da pesquisa inserindo da data de início e de fim e realiza a consulta, tendo como retorno uma lista com informações dos alertas ocorridos que não foram verificados pelos usuários.

Data Alerta	Data Estabilidade	Tipo Alerta	Tecnologia	Tempo Normalidade
4/6/2010 16:25:37	4/6/2010 16:25:49	Baixa Velocidade	ControlLoc	00:00:12:000
4/6/2010 16:19:41	4/6/2010 16:20:04	Baixa Velocidade	ControlLoc	00:00:23:000
4/6/2010 16:14:16	4/6/2010 16:14:44	Baixa Velocidade	ControlLoc	00:00:28:000
4/6/2010 16:06:31	4/6/2010 16:06:31	Não Executando	Sascar	00:00:00:000
4/6/2010 16:06:30	4/6/2010 16:06:30	Não Executando	OnixSat	00:00:00:000
4/6/2010 16:06:30	4/6/2010 16:06:30	Não Executando	Rodosis	00:00:00:000
4/6/2010 16:06:29	4/6/2010 16:06:29	Não Executando	Omnalink	00:00:00:000
4/6/2010 16:06:29	4/6/2010 16:06:29	Não Executando	ControlLoc	00:00:00:000
4/6/2010 16:03:39	4/6/2010 16:05:48	Poucos Registros	Rodosis	00:02:09:000
4/6/2010 16:00:17	4/6/2010 16:01:10	Poucos Registros	Rodosis	00:00:53:000
4/6/2010 14:51:05	4/6/2010 15:49:24	Não Executando	Omnalink	00:58:19:000
4/6/2010 14:51:05	4/6/2010 15:49:24	Não Executando	OnixSat	00:58:19:000
4/6/2010 14:51:05	4/6/2010 15:49:24	Não Executando	Rodosis	00:58:19:000
4/6/2010 14:51:05	4/6/2010 15:49:24	Não Executando	Sascar	00:58:19:000
4/6/2010 14:51:04	4/6/2010 15:49:24	Não Executando	ControlLoc	00:58:20:000

Figura 45: Tela de Relatório de Alertas não Verificados.

A Figura 46 representa o relatório de alertas verificados. Nela o usuário administrador seleciona o período da pesquisa inserindo da data de início e de fim e realiza a consulta, tendo como retorno uma lista com informações dos alertas ocorridos que foram verificados pelos usuários.

Relatório de Alertas Verificados

Período
 Início: 4/6/2010 Fim: 4/6/2010 Pesquisar

Drag a column header here to group by that column

Data Alerta	Data Estabilidade	Tipo de Alerta	Tecnologia	Usuário
4/6/2010 16:35:18	4/6/2010 16:36:11	Poucos Registros	Rodosis	Raphael Silveira
4/6/2010 16:13:47	4/6/2010 16:17:55	Poucos Registros	Rodosis	Raphael Silveira

Figura 46: Tela de Relatório de Alertas Verificados.

A Figura 47 representa os detalhes do alerta, onde o usuário administrador seleciona um alerta na lista existente na tela de relatório de alertas verificados e tem acesso aos detalhes da ação do usuário na tentativa de estabilizar o processo.

Relatório de Alertas Verificados

Período
 Início: 4/6/2010 Fim: 4/6/2010 Pesquisar

Drag a column header here to group by that column

Data Alerta	Data Estabilidade	Tipo de Alerta	Tecnologia	Usuário
4/6/2010 16:35:18	4/6/2010 16:36:11	Poucos Registros	Rodosis	Raphael Silveira
4/6/2010 16:13:47	4/6/2010 16:17:55	Poucos Registros	Rodosis	Raphael Silveira

Detalhes do Alerta

Código: 1744

Tipo de alerta: Poucos Registros

Data do Alerta: 4/6/2010 16:13:47

Data de Estabilidade: 4/6/2010 16:17:55

Tecnologia: Rodosis

Usuário Leitura: Raphael Silveira

Tempo para leitura: 00:02:14

Tempo de atuação: 00:01:53

Tempo para solução: 00:04:08

Informações do usuário

Entrado em contato com a tecnologia, onde fomos informados que o sistema deles está passando por oscilações de comunicação.

Número de registro por pacotes retornou ao normal. Não há mais oscilação no sistema.

Figura 47: Tela de Detalhes do Alerta.

4 CONCLUSÃO

O presente estudo desenvolveu-se num período de seis meses. Ele permitiu o desenvolvimento de um sistema que proporciona o monitoramento das integrações de dados entre a empresa Angellira e as tecnologias ControlLoc®, Omnilink®, OnixSat®, Rodosis® e Sascar® em tempo real, que possibilita ações imediatas na prevenção e correção de eventuais anormalidades no recebimento de informações.

O *software* apresenta informações ao usuário através de *feedback* gráfico auxiliando na compreensão dos dados. Ele efetua a leitura de informações provenientes do processo de integração e efetua cálculos matemáticos que permitem avaliar o desempenho desse processo. Estes cálculos determinam identificar se a integração está atrasada, se não está sendo realizada, se não está recebendo informações, se a velocidade de inserção de registros está lenta e se estiver recebendo poucos registros.

Entretanto o *software* pode operar de uma maneira mais abrangente, pois neste primeiro momento era preciso utilizar as informações do processo de integração de maneira mais proveitosa e não houve preocupação em explorar todas as possibilidades que a criação desta ferramenta possibilitaria. Estas possibilidades vão desde calcular a eficiência das tecnologias envolvidas no processo até a avaliação dos operadores de suporte envolvidos.

REFERÊNCIAS

AMORIN, Cynthia Helena Gomes de; CAMARA, Simone Miranda; CARLINI JUNIOR, Reginaldo José. A Utilização do Sistema de Rastreamento de Veículos no Transporte de Cargas: um estudo junto à segsat. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 13., Bauru, SP, 2006. Disponível em: <<http://www.simpep.feb.unesp.br/>>. Acesso em: 12 junho 2010.

ANEFALOS, Lílian Cristina. **Gerenciamento de Frotas do Transporte Rodoviário de Cargas Utilizando sistemas de Rastreamento por Satélite**. 1999. 134 f. Dissertação (Mestrado em Economia Aplicada)-Escola Superior de Agricultura Luiz Queiroz, Universidade de São Paulo, Piracicaba, 1999.

ANGELLIRA RASTREAMENTO SATELITAL <<http://www.angellira.com.br>> . Acesso em: 15 outubro 2009.

CONTROLLOC. **Descrição da Base de Integração**. 2008.

CUNHA, Davi. **Web Services, SOAP e Aplicações Web**. 2002. Disponível em: <http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html>. Acesso em: 12 setembro 2009.

DEITEL, Harvey; DEITEL, Paul J.; NIETO, Ramon . **XML: Como Programar**. 1 ed. Porto Alegre: Bookman, 2003. 974 p.

GUIMARÃES, Célio. **Introdução a Linguagens de Marcação: HTML, XHTML, SGML, XML**. 2005. Disponível em: <<http://www.ic.unicamp.br/~celio/inf533/docs/markup.html>>. Acesso em: 01 setembro 2009.

LETHAM, Lawrence. **GPS Made Easy**. 5 ed. Canada: Mountaineers Books, 2008. 193 p.

MAROTTA, Bruno. Threads e Conexões. **Revista Clube Delphi**, n. 54, 2007. Disponível em: <<http://www.devmedia.com.br>>. Acesso em: 15 abril 2010.

PAULI, Guinther. Threads: Concorrência no Delphi e Kylix. **Revista Clube Delphi**, n. 21, 2005. Disponível em: <<http://www.devmedia.com.br>>. Acesso em: 15 abril 2010.

MOURA, Luiz. **Avaliação do Impacto dos Sistemas Integrados de Rastreamento de Veículos na Logística e Gerenciamento de Risco**. 2005. Disponível em: <<http://www.guialog.com.br/Y610.htm>>. Acesso em: 12 junho 2010.

NEVES, Marco Antônio Oliveira. **Gestão de Risco Movimenta R\$ 4 bilhões no País**. Disponível em: <<http://www.aslog.org.br>>. Acesso em: 25 maio 2010.

OMG. **Unified Modeling Language (UML): Version 2.2**. 2010. Disponível em: <<http://www.omg.org>> Acesso em: 01 maio 2010.

OMNILINK. **IAS: Interface de Acesso à Central Saver Turbo Versão 6.0**. 2009.

ONIXSAT. **Integração Protocolo XML: Webservice 2.0**. 2009.

RECKZIEGEL, Mauricio. **Entendendo os Web Services**. 2006. Disponível em: <http://imasters.uol.com.br/artigo/4245/webservices/entendendo_os_webservices> Acesso em: 12 setembro 2009.

REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação**. 3 ed. rev. e ampl. Rio de Janeiro: Brasport, 2005. 324 p.

RODOSIS. **Rodosis Integração 2,32: Protocolo de Comunicação**. 2008.

SAMPAIO, Cleuton. **SOA e Web Services em Java**. BrandSport. 1 ed. Rio de Janeiro: Brasport, 2006. 168 p.

SASCAR. **Manual de Comandos Via Integração**. 2008

SCHEFFER, Rosely. **Uma Visão Geral Sobre Threads**. 2007. 6 f. Universidade Estadual de Maringá , Departamento de Informática, Campo Mourão, 2007.

SILVA FILHO, Antonio Mendes da; **Programando com XML**. 1 ed. Rio de Janeiro: Campus, 2004. 332 p.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. Addison-Wesley, 2007. 568 p.

TENENBAUM, Andrew; WOODHULL, Albert. **Sistemas Operacionais**. 2 ed. Porto Alegre: Bookman, 1999.

W3C. **Extensible Markup Language (XML)**. 2009. Disponível em: <<http://www.w3c.org>>. Acesso em: 01 setembro 2009.