

Documentação Integração WebService

by Trucks Control Serviços de Logística Ltda

Integração Trucks Control

WebService

Conteúdo

Parte I Sobre a Integração	6
Parte II Introdução a Linguagem XML	8
1 O que é?.....	9
2 Diferença entre o HTML e o XML.....	9
Parte III Informações	11
Parte IV XML para Requisições	17
1 Requisição de Acessórios.....	17
2 Requisição de Comando.....	22
3 Requisição de Espelhamento	33
SPY	33
Veículo	46
4 Requisição de Macro.....	60
5 Requisição de Spy.....	70
6 Requisição de Veículo.....	77
7 Requisição de Referência Entrega.....	101
8 Requisição de Status Comandos - I.E.....	102
9 Requisição de Inteligência Embarcada	106
Grupo de Macro	107
Perfil Operacional	113
Pontos de Controle	120
Cerca Eletrônica	132
Outras Requisições	140
10 Requisição de Telemetria.....	143
11 Requisição do Horário do Servidor.....	153
12 Requisição de Motorista.....	154
Parte V Erros para Envio Incorreto do XML	162
Parte VI Relacionamentos	164
Parte VII Exemplo em C#	167
1 Requisição.....	167
2 Descompactação.....	169
Parte VIII Lista de Comandos para SPY	172

Parte IX Lista de Comandos por Equipamento	175
Parte X Erros Retornados	202
1 Envio de Comandos.....	202
2 Envio de Comandos SPY.....	205
Parte XI O que há de Novo ?	207
Parte XII Baixar versão em pdf	209
Index	0

Parte



1 Sobre a Integração



O sistema de Integração da Trucks Control é disponibilizado através de um Webservice.

Temos um link através do qual os clientes conseguem requisitar as informações dos equipamentos espelhados para sua conta.

Hoje nosso sistema de integração envia todas as informações das mensagens dos veículos: latitude, longitude, rua, rodovia, eventos gerados, macros, etc...

Parte



2 Introdução a Linguagem XML



OBS: A criação de um atributo novo na estrutura XML, não deve causar qualquer transtorno, falha ou mal funcionamento do sistema.

A criação de novos atributos será uma prática normal, sendo documentado no manual de integração e distribuído a medida que se torne necessário.

Escolhemos essa forma de trabalho, para poder atender a qualquer momento, alguma necessidade em particular de algum cliente, de forma a não causar impacto nos demais.

O que é?

XML é a abreviação de EXtensible Markup Language (Linguagem extensível de formatação). Trata-se de uma linguagem que é considerada uma grande evolução na internet.



Diferença entre o HTML e o XML

O HTML e o XML tem lá suas semelhanças, sendo a principal o fato de utilizar tags (palavras-chaves e parâmetros). Em ambas as linguagens, cada tag consiste em duas partes, uma que inicia e outra que fecha o comando.



2.1 O que é?

XML é a abreviação de EXtensible Markup Language (Linguagem extensível de formatação). Trata-se de uma linguagem que é considerada uma grande evolução na internet.

O XML é uma especificação técnica desenvolvida pela W3C (World Wide Web Consortium - entidade responsável pela definição da área gráfica da internet), para superar as limitações do HTML, que é o padrão das páginas da Web.

A linguagem XML é definida como o formato universal para dados estruturados na Web. Esses dados consistem em tabelas, desenhos, parâmetros de configuração, etc. A linguagem então trata de definir regras que permitem escrever esses documentos de forma que sejam adequadamente visíveis ao computador.

2.2 Diferença entre o HTML e o XML

O HTML e o XML tem lá suas semelhanças, sendo a principal o fato de utilizar tags (palavras-chaves e parâmetros). Em ambas as linguagens, cada tag consiste em duas partes, uma que inicia e outra que fecha o comando.

No entanto, em muitos casos, se uma tag é aberta no HTML e não é fechada, a página é exibida mesmo assim. Já no XML, se houver qualquer erro desse tipo, a aplicação simplesmente pára. Percebe-se com esse exemplo, que o HTML é uma linguagem mais tolerante, enquanto o XML é altamente rígido. Isso pode até parecer uma desvantagem, mas se for, é compensada pela extensibilidade do XML.

Para um melhor entendimento, veja o seguinte fato: no HTML, a tag `<p> </p>` indica o início e o fim de um parágrafo. No XML, as tags são usadas para definir blocos de dados.

O que isso quer dizer? Quer dizer que, `<p> </p>` podem significar qualquer coisa que o programador desejar. Por exemplo, `<p> </p>` podem significar peso, pessoa, nome, endereço, classe, carro, enfim, o que o usuário quiser que represente. Por essa característica, o XML é até considerado por muitos uma linguagem capaz de gerar outras linguagens, visto que quem define os comandos e suas funções é o programador.

A praticidade é tanta que torna-se possível um usuário criar uma coleção própria de tags e aplicá-las nas páginas e documentos que desejar.

Parte



3 Informações

Informações Fornecidas pela Integração

Acessórios: Todos os acessórios disponíveis para os veículos;

Acessórios Veículos: Todos os acessórios instalados por veículos;

Macro Grupo: Todos os grupos de macros disponíveis para o cliente;

Macro Ativo: Indica qual grupo de macro está ativo para cada veículo, sendo um grupo no sentido Central-Veículo e outro Veículo-Central;

Macro Item: Todos os itens que compõem cada grupo de macro;

Veículos: Informações de todos os veículos espelhados para o cliente;

Mensagens Veículos: Todas as mensagens dos veículos espelhados para o cliente;

Spy's: Informações de todos os Spy's espelhados para o cliente;

Mensagens Spy's: Todas as mensagens dos Spy's espelhados para o cliente;

Inteligência Embarcada: Direito e transferencia de I.E., Perfis Embarcados, etc;

Telemetria: Todos os relatórios de Telemetria que o equipamento envia.

Espelhamento: Cancelar, Rejeitar, Aceitar, etc;

Interação com o Equipamento

Comandos veículos: Envio de comandos aos veículos;

Macros: Envio de macros aos veículos;

Comando Spy: Envio de comandos aos equipamentos Spy;

Inteligência Embarcada: embarque de pontos de controle, embarque de macros, limpeza de I.E., etc;

Info - Requisições

O sistema consegue recuperar apenas as mensagens de até 1(uma) hora atrasada, caso o sistema fique parado por mais de cinquenta minutos, as mensagens fora desse intervalo serão perdidas.

A quantidade máxima de mensagens recebidas por solicitação dependerá da frequência com que são feitas as requisições.

Para cada requisição deve ser respeitado um intervalo de tempo, ou seja, uma mesma requisição não pode ser repetida dentro de um determinado intervalo de tempo.

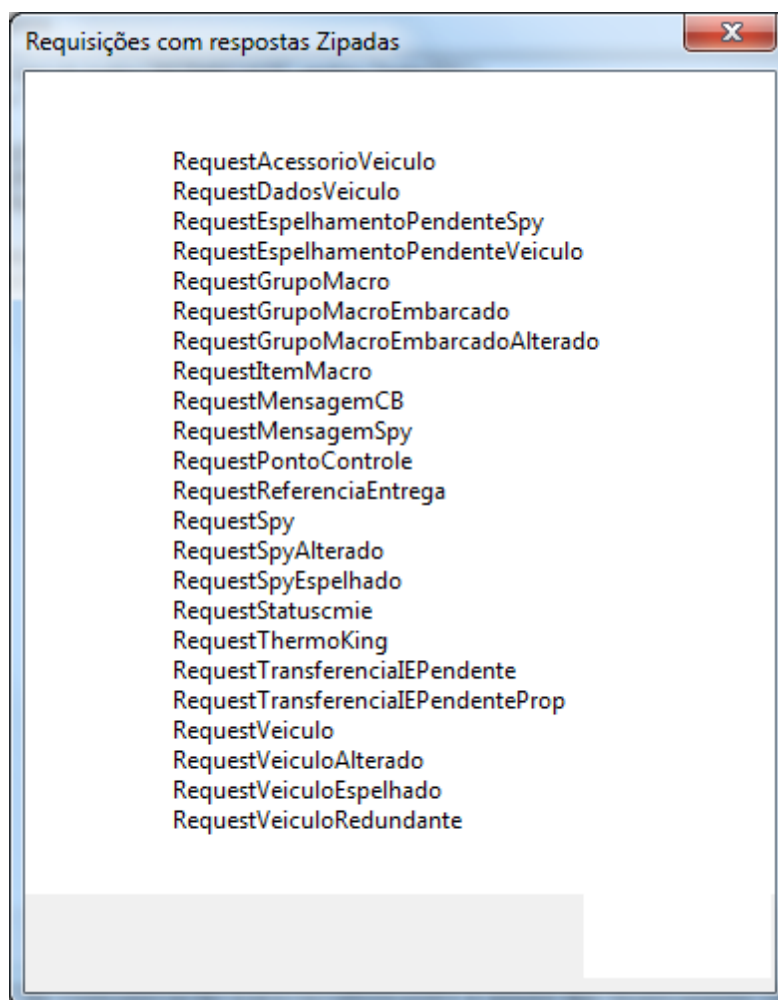
Na tabela abaixo temos descrito estes intervalos de tempo para cada requisição:

Requisição	Tempo
RequestAcessorio	5 minutos
RequestAcessorioVeiculo	5 minutos
RequestCarretas	5 minutos
RequestCercaEletronica	5 minutos
RequestCercaEletronicaEmbarcada	5 minutos
RequestDadosVeiculo	30 segundos
RequestEspelhamentoPendenteVeiculo	30 segundos
RequestGrupoMacro	30 segundos
RequestGrupoMacroEmbarcado	5 minutos
RequestItemMacro	5 minutos
RequestMensagemCB	30 segundos
RequestMensagemSpy	30 segundos
RequestPerfilOperacional	30 segundos
RequestPerfilOperacionalEmbarcado	30 segundos
RequestPontoControle	30 segundos
RequestReferenciaEntrega	10 segundos
RequestSpy	5 minutos
RequestStatuscmie	30 segundo
RequestTelemetria	60 minutos
RequestTelemetrialtem	24 horas
RequestThermoKing	30 segundos
RequestTranferencialEPendente	5 minutos
RequestTranferencialEStatus	30 segundos
RequestTransferencialEPendentecli	5 minutos
RequestVeiculo	5 minutos
RequestVeiculoEspelhado	5 minutos
RequestVeiculoRedundante	5 minutos

OBS: A quantidade máxima de mensagens que podem ser enviadas por requisição é

30.

Na imagem a seguir o informativo das requisições que necessitam descompactação:



Links para conexão

Os links de comunicação para recebimento e envio das mensagens são:

<http://webservice.onixsat.com.br>

<http://webservice1.onixsat.com.br>

O Webservice da Trucks Control não foi desenvolvido utilizando o protocolo soap, o Webservice foi desenvolvido para trabalhar com requisição/resposta puro XML (HttpRequest e HttpResponse). Sendo assim, não existe o WSDL.

Tecnologias Trucks Control

- Satellite System (eqp = 1)
- Hybrid System (eqp = 2)
- Light GSM 1 (eqp = 3)
- Satellite Sky (eqp = 4)
- Smart Híbrido (eqp = 6)
- SpyTrack (eqp = 7)
- Smart GSM (eqp = 8)
- Slim GSM 1 (eqp = 9)
- Light GSM 2 (eqp = 10)
- Slim GSM 2 (eqp = 11)
- Trailer GSM System (eqp = 12)
- Slim GSM 3 (eqp = 13)
- Slim GSM 4 (eqp = 33)
- SpyTrack2 (eqp = 14)
- Rail Patrol (eqp = 29)
- Smart2 Híbrido (eqp = 35)
- Smart 2 GSM (eqp = 36)
- SmartMid Híbrido (eqp = 45)
- SmartMid GSM (eqp = 46)
- Connect Smart GSM (eqp = 54)
- Connect Smart Híbrido (eqp = 55)

Parte



4 XML para Requisições

Na sequencia estarão descritos os XML's que devem ser enviados ao link para requisição das mensagens.

Alguns campos são opcionais, ou seja, serão enviados somente quando seu valor for diferente do default. Sendo assim, a aplicação obrigatoriamente deve verificar se o XML recebido contém os campos opcionais.

Conforme novas funcionalidades estiverem disponíveis nos equipamentos, as mesmas estarão disponíveis na integração. Sendo assim, novos campos podem ser enviados nos XML's atuais. A aplicação deverá trabalhar de forma que estes novos campos não interfiram no seu processo.

Os pacotes serão enviados no formato zip ou gzip (neste ultimo caso apenas se o desenvolvedor achar necessária a alteração, do contrário **ZIP** é o formato padrão). Neste Help fornecemos um exemplo de descompactação em C#, caso necessário temos exemplos em outras linguagens, entre em contato conosco. Os pacotes são enviados zipados em forma de arquivo zip ou gzip conforme preferencia sendo que esta deve ser informada para que possamos cadastrar o formato correto ao cliente, por padrao iremos mandar em formato zip. Os pacotes sofrem uma redução de 15 a 20 vezes do seu tamanho original.

4.1 Requisição de Acessórios

Acessórios (RequestAcessorio)

XML para requisição da lista dos Acessórios existentes

```
<RequestAcessorio>
  <login?></login>
  <senha?></senha>
</RequestAcessorio>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAcessorio>
  <Acessorio>
    <acvID?></acvID>           [int]
    <descricao?></descricao>   [varchar(50)]
  </Acessorio>
```

</ResponseAcessorio>

Descrição:

acvID: Id do acessório;

descrição: Descrição do acessório;

Exemplo

Requisição:

```
<RequestAcessorio>
  <login>12345678910</login>
  <senha>xxxx</senha>
</RequestAcessorio>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAcessorio>
  <Acessorio>
    <acvID>1</acvID>
    <descricao>Botão de Pânico Adicional</descricao>
  </Acessorio>
  <Acessorio>
    <acvID>2</acvID>
    <descricao>Kit Básico</descricao>
  </Acessorio>
  <Acessorio>
    <acvID>3</acvID>
    <descricao>Teclado de Macro</descricao>
  </Acessorio>
  <Acessorio>
    <acvID>4</acvID>
    <descricao>Sensor da Porta do Motorista</descricao>
  </Acessorio>
  <Acessorio>
    <acvID>5</acvID>
    <descricao>Sensor da Porta do Carona</descricao>
  </Acessorio>
</ResponseAcessorio>
```

☐ Acessórios Veículos (RequestAcessorioVeiculo)

XML para requisição dos acessórios que cada veículo possui. Serão enviados somente os acessórios dos novos veículos que foram espelhados.

```
<RequestAcessorioVeiculo>
  <login?></login>
  <senha?></senha>
  <todosItens>0</todosItens>
</RequestAcessorioVeiculo>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAcessorioVeiculo>
  <AcessorioVeiculo>
    <veilID?></veilID>    [int]
    <acvID?></acvID>    [int]
  </AcessorioVeiculo>
</ResponseAcessorioVeiculo>
```

Descrição:

veilID: Id do veículo;

acvID: Id do acessório;

☐ Exemplo

Requisição:

```
<RequestAcessorioVeiculo>
  <login>12345678910</login>
  <senha>xxxx</senha>
</RequestAcessorioVeiculo>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAcessorioVeiculo>
  <AcessorioVeiculo>
    <veilID>3279</veilID>
```

```

        <acvID>2</acvID>
    </AcessorioVeiculo>
    <AcessorioVeiculo>
        <veilID>3279</veilID>
        <acvID>3</acvID>
    </AcessorioVeiculo>
    <AcessorioVeiculo>
        <veilID>3279</veilID>
        <acvID>15</acvID>
    </AcessorioVeiculo>
    <AcessorioVeiculo>
        <veilID>3488</veilID>
        <acvID>2</acvID>
    </AcessorioVeiculo>
    <AcessorioVeiculo>
        <veilID>3488</veilID>
        <acvID>4</acvID>
    </AcessorioVeiculo>
</ResponseAcessorioVeiculo>

```

Acessórios Veículos (TODOS) - (RequestAcessorioVeiculo)

XML para requisição dos acessórios que cada veículo possui. Serão enviados todos os acessórios veículos espelhados.

```

<RequestAcessorioVeiculo>
    <login>?</login>
    <senha>?</senha>
    <todosItens>1</todosItens>
</RequestAcessorioVeiculo>

```

Reposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseAcessorioVeiculo>
    <AcessorioVeiculo>
        <veilID>?</veilID>    [int]
        <acvID>?</acvID>    [int]
    </AcessorioVeiculo>
</ResponseAcessorioVeiculo>

```

Descrição:

veilID: Id do veículo;

acvID: Id do acessório;

 **Exemplo****Requisição:**

```
<RequestAcessorioVeiculo>
  <login>12345678910</login>
  <senha>xxxx</senha>
  <todos>1</todos>
</RequestAcessorioVeiculo>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAcessorioVeiculo>
  <AcessorioVeiculo>
    <veilID>3279</veilID>
    <acvID>2</acvID>
  </AcessorioVeiculo>
  <AcessorioVeiculo>
    <veilID>3279</veilID>
    <acvID>3</acvID>
  </AcessorioVeiculo>
  <AcessorioVeiculo>
    <veilID>3279</veilID>
    <acvID>15</acvID>
  </AcessorioVeiculo>
  <AcessorioVeiculo>
    <veilID>3488</veilID>
    <acvID>2</acvID>
  </AcessorioVeiculo>
  <AcessorioVeiculo>
    <veilID>3488</veilID>
    <acvID>4</acvID>
  </AcessorioVeiculo>
</ResponseAcessorioVeiculo>
```

Nota

Para confirmar o tipo de teclado que o equipamento possui realizar a [requisição de AcessóriosVeiculos](#), onde o acvID = 3 - Teclado Pequeno e 17 e 66 - Teclado Grande

4.2 Requisição de Comando

Envio de Comandos para Veículos - (RequestEnvioComando)

XML para o envio de comandos para os veículos.

```
<RequestEnvioComando login="05899105909" senha="123456">
  <comando>
    <id>?</id>
    <veilD>?</veilD>
    <mensagem>?</mensagem>
    <usuario>?</usuario>
  </comando>
</RequestEnvioComando>
```

Descrição:

id: é um identificador seqüencial para o envio de comandos aos veículos, este valor nunca deve ser repetido, pois caso isto ocorra, ficará comprometido o status do comando;

veilD: identificação do veículo;

mensagem: [código do comando](#). Ao enviar um comando é necessário saber o tipo de equipamento, informação disponível no campo eqp, quando solicitada as informações dos veículos.

usuario: usuário que está enviando o comando;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioComando>
  <comando>
    <id>?</id>          [int]
    <status>?</status>  [int]
    <erro>?</erro>     [int]
  </comando>
</ResponseEnvioComando>
```

Descrição:

id: Identificador seqüencial do envio do comando;

status: Status do envio :

2 – comando processado com sucesso (servidor processou e enviou o comando ao veículo);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – enviado (comando enviado, aguardando processamento do servidor (comando ainda não foi enviado para o equipamento));

5 – comando rejeitado no servidor;

erro: [Código de erro](#), caso haja um erro no envio de um comando.

 **Exemplo****Requisição:**

```
<RequestEnvioComando login="12345678910" senha="xxxx">
  <comando>
    <id>34</id>
    <veild>45872</veild>
    <mensagem>1B</mensagem>
    <usuario>carlao</usuario>
  </comando>
  <comando>
    <id>35</id>
    <veild>45872</veild>
    <mensagem>9C</mensagem>
    <usuario>carlao</usuario>
  </comando>
</RequestEnvioComando>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioComando>
  <comando>
    <id>34</id>
    <status>4</status>
    <erro>0</erro>
  </comando>
  <comando>
    <id>35</id>
    <status>4</status>
    <erro>0</erro>
  </comando>
</ResponseEnvioComando>
```

Envio de Macros (mensagens) - (RequestEnvioMacro)

XML para o envio de macros para os veículos.

```
<RequestEnvioMacro login="?" senha="?">
  <macro tipo="1">
    <id>?</id>
    <veilD>?</veilD>
    <tfrID>?</tfrID>
    <usuario>?</usuario>
  </macro>
</RequestEnvioMacro>
```

Descrição:

id: Identificador seqüencial para envio de macros. Este valor deve ser único, nunca deve ser repetido para o envio de macros (a solicitação de macro livre deve seguir esta mesma seqüência).

veilD: Identificador do veículo;

tfrID: Código da macro. O código da macro é obtido ao solicitar na "solicitação Macro Item";

usuario: Nome do usuário;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioMacro>
  <macro>
    <id>?</id>      [int]
    <status>?</status> [int]
    <erro>?</erro>   [int]
  </macro>
</ResponseEnvioMacro>
```

Descrição:

id: Identificador seqüencial do envio da macro;

status: Status do envio :

- 1 – a processar;
- 2 – comando processado com sucesso (servidor processou e enviou o comando ao veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – enviado (comando enviado, aguardando processamento do servidor (comando ainda não foi enviado para o equipamento));
- 5 – comando rejeitado no servidor;

Erro: [Código de erro](#), caso haja um erro no envio de uma macro.

 **Exemplo****Requisição:**

```
<RequestEnvioMacro login="12345678910" senha="xxxx">
  <macro tipo="1">
    <id>13</id>
    <veild>49752</veild>
    <tfrID>16867</tfrID>
    <usuario>carlao</usuario>
  </macro>
</RequestEnvioMacro>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioMacro>
  <macro>
```

```

        <id>13</id>
        <status>4</status>
        <erro>0</erro>
    </macro>
</ResponseEnvioMacro>

```

Envio de Macros (mensagens livres) - (RequestEnvioMacro)

XML para o envio de macros livres / formulários para os veículos.

```

<RequestEnvioMacro login="?" senha="?">
  <macro tipo="2">
    <id>?</id>
    <veilD>?</veilD>
    <tfrID>?</tfrID>
    <tfrParam>?</tfrParam>
    <usuario>?</usuario>
  </macro>
</RequestEnvioMacro>

```

Descrição:

id: Identificador seqüencial para envio de macros. Este valor deve ser único, nunca deve ser

repetido para o envio de macros.

veilD: Identificador do veículo;

tfrID: O código da macro é obtido na "solicitação Macro Item";

tfrParam: Descrição da macro livre / formulário. Este campo também é obtido na solicitação Macro Item, no campo cont;

Usuário: Nome do usuário;

EXEMPLO PARA ENVIO DE MACRO LIVRE:

- Para mensagens do tipo Macro Formulário TD3X, o campo tfrParam deve ser preenchido conforme o campo cont da Macro Item:

tfrContent	tfrParam
Mensagem livre: \144\1	Mensagem de até 144 caracteres do tipo

144 - tamanho máx. da string	alfanumérico (1)*. Ex: Que horas vai chegar?
1 - tipo de dados.	No final de cada item da mensagem deve vir o símbolo " ".
Nome: \20\1 CEP: \14\0 DATA : \8\2 HORA : \5\3	Ex: João da Silva 87080520 140508 1200
FIXO A \0\4 FIXO B \0\4 CODIGO: \15\1 FIXO C \0\4	Ex: 12345 Obs: os campos do tipo 4 não devem ser enviados.

- Os tipos de dados são : 0 – Numérico; 1 – Alfanumérico; 2 – Data; 3 – Hora; 4 – Somente Leitura;
- Verificar no exemplo acima os formatos que devem ser enviados as datas e horas.

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioMacro>
  <macro>
    <id>?</id>          [int]
    <status>?</status>  [int]
    <erro>?</erro>      [int]
  </macro>
</ResponseEnvioMacro>
```

Descrição:

id: Identificador seqüencial do envio da macro;

status: Status do envio :

- 1 – a processar;
- 2 – comando processado com sucesso (servidor processou e enviou o comando ao veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – enviado (comando enviado, aguardando processamento do servidor (comando ainda não foi enviado para o equipamento));
- 5 – comando rejeitado no servidor;

Erro: [Código de erro](#), caso haja um erro no envio de uma macro.

Exemplo

Requisição:

```
<RequestEnvioMacro login="12345678910" senha="xxx">
  <macro tipo="2">
    <id>14</id>
    <veild>49752</veild>
    <tfrID>16865</tfrID>
    <tfrParam>teste web service</tfrParam>
    <usuario>carlao</usuario>
  </macro>
</RequestEnvioMacro>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioMacro>
  <macro>
    <id>14</id>
    <status>4</status>
    <erro>0</erro>
  </macro>
</ResponseEnvioMacro>
```

Envio de Comandos para SPY (RequestEnvioComandoSpy)

XML para o envio de comandos para os equipamentos.

```
<RequestEnvioComandoSpy login="?" senha="?">
  <comandoSpy>
    <id>?</id>
    <spyID>?</spyID>
    <mensagem>?</mensagem>
    <usuario>?</usuario>
  </comandoSpy>
</RequestEnvioComandoSpy>
```

Descrição:

id: Identificador seqüencial para envio dos comandos SPY's. Este valor deve ser único, nunca deve ser repetido para o envio de comandos SPY's, pois caso isto ocorra, ficará comprometido o status do comando;

spyID: identificação do SPY;

mensagem: [código do comando](#).

Usuário: usuário que está enviando o comando;

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioComandoSpy>
  <comandoSpy>
    <id>?</id>           [int]
    <status>?</status>   [int]
    <erro>?</erro>       [int]
  </comandoSpy>
</ResponseEnvioComandoSpy>
```

Descrição:

id: Identificador seqüencial do envio do comando SPY;

status: Status do envio:

- 2 – comando processado com sucesso (servidor processou e enviou o comando ao equipamento);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – enviado (comando enviado, aguardando processamento do servidor (comando ainda não foi enviado para o equipamento));
- 5 – comando rejeitado no servidor;

erro: [Código de erro](#), caso haja um erro no envio de um comando para um SPY.

 **Exemplo**

Requisição:

```
<RequestEnvioComandoSpy login="12345678910" senha="xxxx">
  <comandoSpy>
    <id>5</id>
    <spyId>412</spyId>
    <mensagem>01</mensagem>
    <usuario>carlao</usuario>
  </comandoSpy>
</RequestEnvioComandoSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioComandoSpy>
  <comandoSpy>
    <id>5</id>
    <status>4</status>
    <erro>0</erro>
  </comandoSpy>
</ResponseEnvioComandoSpy>
```

Comando exclusivo SpyTrack2 (modo escuta) - (RequestEnvioComandoSpy)

Envio de comando habilitar modo escuta para o SpyTrack2, este comando faz com que o equipamento seja configurado para receber uma ligação de um aparelho de telefonia celular convencional. Uma vez esta chamada recebida o ligador conseguirá ouvir o que está acontecendo ao redor do equipamento.

```
<RequestEnvioComandoSpy login="?" senha="?">
  <comandoSpy>
    <id>?</id>
    <spyID>?</spyID>
    <mensagem>16</mensagem>
    <tel>?</tel>
    <time>?</time>
    <usuario>?</usuario>
  </comandoSpy>
</RequestEnvioComandoSpy>
```

Descrição:

id: Identificador seqüencial para envio do comando SpyTrack2. Este valor deve ser único, nunca deve ser repetido para o envio de comando SpyTrack2, pois caso isto ocorra, ficará comprometido o status do comando;

spyID: identificação do SpyTrack2;

mensagem: código do comando. Para este tipo de comando sempre será o código 16.

tel: deverá ser colocado o número de telefone que irá efetuar a ligação para o SpyTrack2, deverá ser colocado o código de área do telefone que efetuará a ligação e em seguida o numero do telefone sem nenhum caracter separando os números. Exemplo: 4399999999

time: significa quanto tempo o equipamento SpyTrack2 ficará no aguardo desta ligação, caso não receba esta ligação dentro do período programado pelo 'time' ele voltará a configuração normal de posicionamento. O limite mínimo e Maximo para esta configuração em minutos é de 3 e 5 respectivamente, apenas números inteiros, sendo que se colocado um numero menor do que o mínimo, será utilizado o valor mínimo e se colocado um valor maior do que o Maximo, o Maximo será considerado.

usuário: usuário que está enviando o comando;

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioComandoSpy>
  <comandoSpy>
    <id>?</id>           [int]
    <status>?</status>   [int]
    <erro>?</erro>       [int]
  </comandoSpy>
</ResponseEnvioComandoSpy>
```

Descrição:

id: Identificador seqüencial do envio do comando SpyTrack2;

status: Status do envio :

2 – comando processado com sucesso (servidor processou e enviou o comando ao veículo);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – enviado (comando enviado, aguardando processamento do servidor (comando ainda não foi enviado para o equipamento));

5 – comando rejeitado no servidor;

erro: [Código de erro](#), caso haja um erro no envio de um comando para um SpyTrack2.

Exemplo

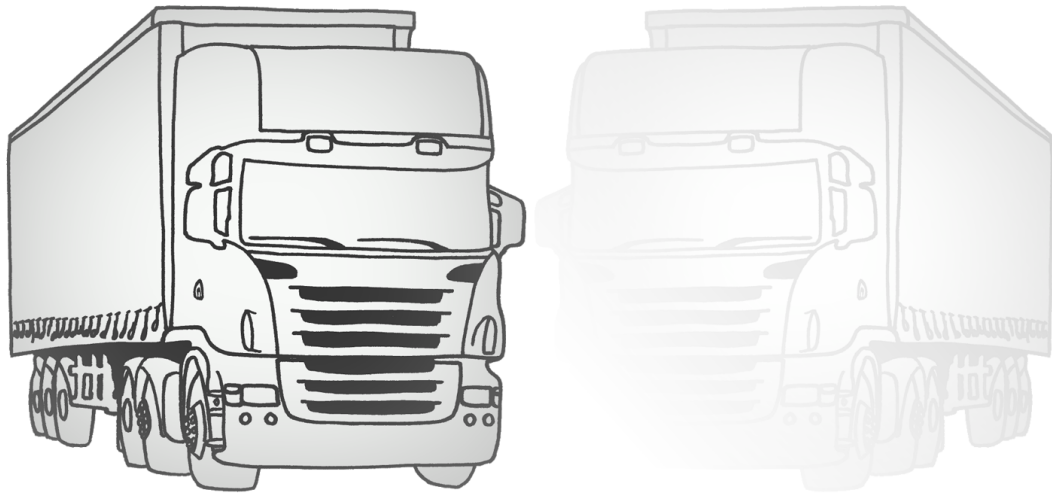
Requisição:

```
<RequestEnvioComandoSpy login="?" senha="?">
  <comandoSpy>
    <id>1</id>
    <spyID>23456</spyID>
    <mensagem>16</mensagem>
    <tel>4399999999</tel>
    <time>3</time>
    <usuario>integracao</usuario>
  </comandoSpy>
</RequestEnvioComandoSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEnvioComandoSpy>
  <comandoSpy>
    <id>5</id>
    <status>4</status>
    <erro>0</erro>
  </comandoSpy>
</ResponseEnvioComandoSpy>
```

4.3 Requisição de Espelhamento



Espelhamento de Equipamentos

- [SPY](#)
- [Veículo](#)

4.3.1 SPY

Novo Espelhamento (Proprietário) - (RequestNovoEspelhamentoSpy)

XML para realizar o espelhamento de um Spy.

```
<RequestNovoEspelhamentoSpy login="?" senha="?">
  <espelhamento>
    <id?</id>
    <spyID?</spyID>
    <cmd?</cmd>
    <alvo?</alvo>
    <cgccpf?</cgccpf>
    <usuario?</usuario>
  </espelhamento>
```

```
</RequestNovoEspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial do espelhamento enviado pelo cliente. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente.

spyID: Identificador do dispositivo;

cmd: Permissão para o envio de comando ao Spy (0 – Não permitido ; 1 - Permitido);

alvo: Permite criar alvo para o dispositivo portatil;

cgccpf: CNPJ/CPF do cliente para o qual o dispositivo será espelhado;

usuario: Usuário que está realizando o espelhamento.;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseNovoEspelhamentoSpy>
  <espelhamento>
    <id?</id>           [int]
    <status?</status>  [int]
    <erro?</erro>      [int]
  </espelhamento>
</ResponseNovoEspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento realizado;

3 – espelhamento não realizado;

Erro: [Código de erro](#) caso haja um erro no espelhamento de um dispositivo.

Exemplo**Requisição:**

```
<RequestNovoEspelhamentoSpy login="1234567890" senha="xxxx">
  <espelhamento>
    <id>1</id>
    <spyID>23654</spyID>
    <cmd>1</cmd>
    <alvo>1</alvo>
    <cgccpf>1234567890</cgccpf>
    <usuario>integracao</usuario>
  </espelhamento>
</RequestNovoEspelhamentoSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseNovoEspelhamentoSpy>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
</ResponseNovoEspelhamentoSpy>
```

SPY Espelhado (Proprietário) - (RequestSpyEspelhado)

Lista para quem estão espelhados os equipamentos do proprietário

```
<RequestSpyEspelhado>
  <login>?</login>
  <senha>?</senha>
</RequestSpyEspelhado>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseSpyEspelhado>
  <SpyEspelhado>
    <spyID>?</spyID>          [int]
    <cmd>?</cmd>              opc[bit]
    <alvo>?</alvo>            opc[bit]
    <cgccpf>?</cgccpf>        [varchar(14)]
    <cliente>?</cliente>      [varchar(50)]
    <sit>?</sit>              opc[bit]
  </SpyEspelhado>
```

```
</ResponseSpyEspelhado>
```

Descrição:

spyID: Identificador do dispositivo;

cmd: Permissão para o envio de comando ao Spy (0 – Não permitido ; 1 - Permitido);

alvo: Permite criar alvo para o dispositivo portatil;

cgccpf: CNPJ/CPF do cliente para o qual o dispositivo será espelhado;

cliente: Nome do cliente;

sit: Situação do espelhamento : 0 – Pendente ; 1 – Aceito.

 **Exemplo**

Requisição:

```
<RequestSpyEspelhado>  
  <login>1234567890</login>  
  <senha>xxxx</senha>  
</RequestSpyEspelhado>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ResponseSpyEspelhado>  
  <SpyEspelhado>  
    <spyID>23654</spyID>  
    <cmd>1</cmd>  
    <alvo>1</alvo>  
    <cgccpf>0123456789</cgccpf>  
    <cliente>Teste Integração</cliente>  
    <sit>1</sit>  
  </SpyEspelhado>  
</ResponseSpyEspelhado>
```

 **Cancelar Espelhamento (Proprietário) -**

(RequestCancelarEspelhamentoSpy)

XML para cancelamento do espelhamento por parte do proprietário.

```
<RequestCancelarEspelhamentoSpy login="?" senha="?">
  <espelhamento tipo="2">
    <id?</id>
    <spyID?</spyID>
    <cgccpf?</cgccpf>
    <usuario?</usuario>
  </espelhamento>
</RequestCancelarEspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial para cancelamento do espelhamento (cliente e proprietário). Este valor deve ser único, nunca deve ser repetido, será o controle do cliente.

spyID: Identificador do dispositivo;

cgccpf: CNPJ/CPF do cliente para o qual o dispositivo será espelhado;

usuario: Usuario que está cancelando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCancelarEspelhamentoSpy>
  <espelhamento>
    <id?</id>           [int]
    <status?</status>  [int]
    <erro?</erro>      [int]
  </espelhamento>
</ResponseCancelarEspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento cancelado;

3 – espelhamento não cancelado;

Erro: [Código de erro](#) caso haja um erro no espelhamento de um dispositivo.

Exemplo

Requisição:

```
<RequestCancelarEspelhamentoSpy login="1234567890" senha="xxxx">
  <espelhamento tipo="2">
    <id>1</id>
    <spyID>23654</spyID>
    <cgccpf>0123456789</cgccpf>
    <usuario>integracao</usuario>
  </espelhamento>
</RequestCancelarEspelhamentoSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCancelarEspelhamentoSpy>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
</ResponseCancelarEspelhamentoSpy>
```

Cancelar Espelhamento (Cliente) - (RequestCancelarEspelhamentoSpy)

XML para cancelamento do espelhamento por parte do proprietário.

```
<RequestCancelarEspelhamentoSpy login="?" senha="?">
  <espelhamento tipo="1">
    <id>?</id>
    <spyID>?</spyID>
    <cgccpf>?</cgccpf>
    <usuario>?</usuario>
  </espelhamento>
</RequestCancelarEspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial para cancelamento do espelhamento (cliente e proprietário). Este valor deve ser único, nunca deve ser repetido, será o controle do cliente.

spyID: Identificador do dispositivo;

cgccpf: CNPJ/CPF do cliente para o qual o dispositivo será espelhado;

usuario: Usuario que está cancelando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCancelarEspelhamentoSpy>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status>  [int]
    <erro>?</erro>      [int]
  </espelhamento>
</ResponseCancelarEspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento cancelado;

3 – espelhamento não cancelado;

Erro: [Código de erro](#) caso haja um erro no espelhamento de um dispositivo.

 **Exemplo****Requisição:**

```
<RequestCancelarEspelhamentoSpy login="1234567890" senha="xxx">
  <espelhamento tipo="1">
    <id>2</id>
    <spyID>23654</spyID>
    <cgccpf>0123456789</cgccpf>
    <usuario>integracao</usuario>
  </espelhamento>
```

```
</RequestCancelarEspelhamentoSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCancelarEspelhamentoSpy>
  <espelhamento>
    <id>2</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
</ResponseCancelarEspelhamentoSpy>
```

 **Espelhamentos Pendentes (RequestEspelhamentoPendenteSpy)**

XML para requisição dos espelhamentos pendentes.

```
<RequestEspelhamentoPendenteSpy>
  <login>?</login>
  <senha>?</senha>
</RequestEspelhamentoPendenteSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEspelhamentoPendenteSpy>
  <EspelhamentoPendenteSpy>
    <spyID>?</spyID>      [int]
    <serie>?</serie>      [int]
    <desc>?</desc>        [varchar(50)]
    <cmd>?</cmd>          opc[bit]
    <alvo>?</alvo>        opc[bit]
    <prop>?</prop>        [varchar(50)]
  </EspelhamentoPendenteSpy>
</ResponseEspelhamentoPendenteSpy>
```

Descrição:

spyId: Identificador do dispositivo;

serie: número de serie do dispositivo portatil;

desc: descritivo do dispositivo portatil (geralmente algum nome/sinonimo, para identificação do mesmo);

cmd: Permissão para o envio de comando ao Spy (0 – Não permitido ; 1 - Permitido);

alvo: Permite criar alvo para o dispositivo portatil;

Prop: Nome do Proprietário do equipamento;

Exemplo

Requisição:

```
<RequestEspelhamentoPendenteSpy>
  <login>1234567890</login>
  <senha>xxxx</senha>
</RequestEspelhamentoPendenteSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEspelhamentoPendenteSpy>
  <EspelhamentoPendenteSpy>
    <spyID>23654</spyID>
    <serie>99999</serie>
    <desc>teste</desc>
    <cmd>1</cmd>
    <alvo>1</alvo>
    <prop>Teste Integracao</prop>
  </EspelhamentoPendenteSpy>
</ResponseEspelhamentoPendenteSpy>
```

Aceitar Espelhamento Pendente (RequestAREspelhamentoVeiculo)

XML para aceitar os espelhamentos pendentes.

```
<RequestAREspelhamentoVeiculo login="?" senha="?">
  <espelhamento tipo="1">
    <id>?</id>
    <spyID>?</spyID>
    <desc>?</desc>
    <usuario>?</usuario>
  </espelhamento>
```

```
</RequestAREspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente para aceitar ou rejeitar um espelhamento;

spyID: Identificador do dispositivo;

desc: descritivo do dispositivo portatil (geralmente algum nome/sinonimo, para identificação do mesmo);

usuario: Usuario que está aceitando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoSpy>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status>  [int]
    <erro>?</erro>      [int]
  </espelhamento>
</ResponseAREspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento aceito;

3 – espelhamento não aceito;

Erro: [Código de erro](#) caso haja um erro na aceitação do espelhamento de um veículo.

 **Exemplo**

Requisição:

```
<RequestAREspelhamentoVeiculo login="1234567890" senha="xxxx">
```

```
<espelhamento tipo="1">
  <id>1</id>
  <spyID>23654</spyID>
  <desc>teste integracao</desc>
  <usuario>teste</usuario>
</espelhamento>
</RequestAREspelhamentoVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoSpy>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
</ResponseAREspelhamentoSpy>
```

Rejeitar Espelhamentos Pendentes (Cliente) - (RequestAREspelhamentoVeiculo)

XML para o cliente rejeitar os espelhamentos pendentes.

```
<RequestAREspelhamentoSpy login="?" senha="?">
  <espelhamento tipo="2">
    <id>?</id>
    <spyID>?</spyID>
    <usuario>?</usuario>
  </espelhamento>
</RequestAREspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente para aceitar ou rejeitar um espelhamento;

spyID: Identificador do dispositivo;

usuario: Usuario que está aceitando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoSpy>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status> [int]
    <erro>?</erro>     [int]
  </espelhamento>
</ResponseAREspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento rejeitado;

3 – espelhamento não rejeitado;

Erro: [Código de erro](#) caso haja um erro na rejeição do espelhamento de um veículo.

 **Exemplo**

Requisição:

```
<RequestAREspelhamentoSpy login="1234567890" senha="xxxx">
  <espelhamento tipo="2">
    <id>1</id>
    <spyID>23654</spyID>
    <usuario>teste</usuario>
  </espelhamento>
</RequestAREspelhamentoSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoSpy>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
```

```
</espelhamento>
</ResponseAREspelhamentoSpy>
```

Rejeitar Espelhamentos Pendentes (Proprietário) - (RequestAREspelhamentoVeiculo)

XML para o cliente rejeitar os espelhamentos pendentes.

```
<RequestAREspelhamentoSpy login="?" senha="?">
  <espelhamento tipo="3">
    <id>?</id>
    <spyID>?</spyID>
    <cgccpf>?</cgccpf>
    <usuario>?</usuario>
  </espelhamento>
</RequestAREspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente para aceitar ou rejeitar um espelhamento;

spyID: Identificador do dispositivo;

cgccpf: CNPJ/CPF do cliente para o qual o dispositivo será espelhado;

usuario: Usuario que está aceitando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoSpy>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status>  [int]
    <erro>?</erro>     [int]
  </espelhamento>
</ResponseAREspelhamentoSpy>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento rejeitado;

3 – espelhamento não rejeitado;

Erro: [Código de erro](#) caso haja um erro na rejeição do espelhamento de um veículo.

Exemplo

Requisição:

```
<RequestAREspelhamentoSpy login="1234567890" senha="xxxx">
  <espelhamento tipo="3">
    <id>1</id>
    <spyID>23654</spyID>
    <cgccpf>0123456789</cgccpf>
    <usuario>teste</usuario>
  </espelhamento>
</RequestAREspelhamentoSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoSpy>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
</ResponseAREspelhamentoSpy>
```

4.3.2 Veículo

Novo Espelhamento (Proprietário) - (RequestNovoEspelhamentoVeiculo)

XML para realizar o espelhamento de um veículo.

```
<RequestNovoEspelhamentoVeiculo login="?" senha="?">
```

```
<espelhamento>
  <id>?</id>
  <veilD>?</veilD>
  <cmd>?</cmd>
  <IE>?</IE>
  <TIE>?</TIE>
  <validade>?</validade>
  <possocancelar>?</possocancelar>
  <cgccpf>?</cgccpf>
  <usuario>?</usuario>
</espelhamento>
</RequestNovoEspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial do espelhamento enviado pelo cliente. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente.

veilD: Identificador do veículo;

cmd: Permissão para o envio de comando ao veículo (0 – Não permitido ; 1 - Permitido);

IE: Direito a possuir a inteligência embarcada (0 – Não permitido; 1 - Permitido);

TIE: Transferência da inteligência embarcada no ato do espelhamento (0 – Não transfere; 1 - Transfere), obrigatoriamente o cliente deve possuir a permissão da inteligência embarcada (IE) e a IE deve estar com o proprietário;

validade: Deverá ser incluída a data de validade do espelhamento, no formato DD/MM/AAAA (00/00/0000);

OBS: A data de validade não poderá passar de 1 ano, nem pode ser colocada uma data anterior a data atual.

possocancelar: Informa se o espelhamento feito, poderá ser cancelado pelo proprietário (0 - Não posso cancelar ; 1 - Posso cancelar);

OBS: Nestes casos ficará a critério do recebedor o cancelamento, ou até que o espelhamento expire o prazo de validade.

cgccpf: CNPJ/CPF do cliente para o qual o veículo será espelhado;

usuario: Usuário que está realizando o espelhamento.;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseNovoEspelhamentoVeiculo>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status> [int]
    <erro>?</erro>     [int]
  </espelhamento>
</ResponseNovoEspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento realizado;

3 – espelhamento não realizado;

Erro: [Código de erro](#) caso haja um erro no espelhamento de um veículo.

 **Exemplo**

Requisição:

```
<RequestNovoEspelhamentoVeiculo login="1234567890" senha="xxxx">
  <espelhamento>
    <id>1</id>
    <veilID>00000</veilID>
    <cmd>1</cmd>
    <IE>1</IE>
    <TIE>0</TIE>
    <validade>16/04/2016</validade>
    <possocancelar>1</possocancelar>
    <cgccpf>0123456789</cgccpf>
    <usuario>teste</usuario>
  </espelhamento>
</RequestNovoEspelhamentoVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseNovoEspelhamentoVeiculo>
  <espelhamento>
    <id>1</id>
    <status>2</status>
```

```
<erro>0</erro>
</espelhamento>
</ResponseNovoEspelhamentoVeiculo>
```

Veículo Espelhado (Proprietário) - (RequestVeiculoEspelhado)

Lista para quem estão espelhados os equipamentos do proprietário

```
<RequestVeiculoEspelhado>
  <login>?</login>
  <senha>?</senha>
</RequestVeiculoEspelhado>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseVeiculoEspelhado>
  <VeiculoEspelhado>
    <veilD>?</veilD>
    <cmd>?</cmd>
    <IE>?</IE>
    <TIE>?</TIE>
    <cgccpf>?</cgccpf>
    <cliente>?</cliente>
    <validade>?</validade>
    <possocancelar>?</possocancelar>
  </VeiculoEspelhado>
</ResponseVeiculoEspelhado>
```

Descrição:

veilD: Identificação do veículo;

cmd: Permissão para o envio de comando ao veículo (0 – Não permitido ; 1 - Permitido);

IE: Equipamento possui inteligencia embarcada (0 – Não; 1 - Sim);

TIE: Transferência da inteligência embarcada no ato do espelhamento (0 – Não transfere; 1 - Transfere);

cgccpf: CNPJ/CPF do cliente;

cliente: Nome do cliente;

validade: Data de validade do espelhamento.

possocancelar: Indica se o proprietário pode cancelar o espelhamento (0 - Não permitido ; 1 - Permitido).

Exemplo

Requisição:

```
<RequestVeiculoEspelhado>
  <login>1234567890</login>
  <senha>xxxx</senha>
</RequestVeiculoEspelhado>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseVeiculoEspelhado>
  <VeiculoEspelhado>
    <veilD>00000</veilD>
    <cmd>1</cmd>
    <IE>1</IE>
    <TIE>0</TIE>
    <cgccpf>0123456789</cgccpf>
    <cliente>Teste Integracao</cliente>
    <validade>16/04/2016</validade>
    <possocancelar>1</possocancelar>
  </VeiculoEspelhado>
</ResponseVeiculoEspelhado>
```

Cancelar Espelhamento (Proprietário) - (RequestCancelarEspelhamentoVeiculo)

XML para cancelamento do espelhamento por parte do proprietário.

```
<RequestCancelarEspelhamentoVeiculo login="?" senha="?">
  <espelhamento tipo="2">
    <id>?</id>
    <veilD>?</veilD>
    <cgccpf>?</cgccpf>
```

```
        <usuario>?</usuario>
    </espelhamento>
</RequestCancelarEspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial para cancelamento do espelhamento (cliente e proprietário). Este valor deve ser único, nunca deve ser repetido, será o controle do cliente.

veilID: Identificador do veículo;

cgccpf: CNPJ do cliente que será cancelado o espelhamento do veículo;

usuario: Usuario que está cancelando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCancelarEspelhamentoVeiculo>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status> [int]
    <erro>?</erro>     [int]
  </espelhamento>
</ResponseCancelarEspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento cancelado;

3 – espelhamento não cancelado;

Erro: [Código de erro](#) caso haja um erro no espelhamento de um veículo.

 **Exemplo****Requisição:**

```
<RequestCancelarEspelhamentoVeiculo login="1234567890" senha="xxxx">
  <espelhamento tipo="2">
```

```

        <id>1</id>
        <veilID>00000</veilID>
        <cgccpf>0123456789</cgccpf>
        <usuario>teste</usuario>
    </espelhamento>
</RequestCancelarEspelhamentoVeiculo>

```

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseCancelarEspelhamentoVeiculo>
    <espelhamento>
        <id>1</id>
        <status>2</status>
        <erro>0</erro>
    </espelhamento>
</ResponseCancelarEspelhamentoVeiculo>

```

Cancelar Espelhamento (Cliente) - (RequestCancelarEspelhamentoVeiculo)

XML para cancelamento do espelhamento por parte do cliente que recebeu o espelhamento

```

<RequestCancelarEspelhamentoVeiculo login="?" senha="?">
    <espelhamento tipo="1">
        <id>?</id>
        <veilID>?</veilID>
        <usuario>?</usuario>
    </espelhamento>
</RequestCancelarEspelhamentoVeiculo>

```

Descrição:

ID: Identificador seqüencial para cancelamento do espelhamento (cliente e proprietário). Este valor deve ser único, nunca deve ser repetido, será o controle do cliente.

veilID: Identificador do veículo;

usuario: Usuario que está cancelando o espelhamento;

Quando o cliente estiver com a inteligência embarcada do equipamento e cancelar o espelhamento, a inteligência embarcada será transferida para o proprietário do

equipamento.**Resposta:**

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCancelarEspelhamentoVeiculo>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status> [int]
    <erro>?</erro>     [int]
  </espelhamento>
</ResponseCancelarEspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento cancelado;

3 – espelhamento não cancelado;

Erro: [Código de erro](#) caso haja um erro no espelhamento de um veículo.

 **Exemplo****Requisição:**

```
<RequestCancelarEspelhamentoVeiculo login="1234567890" senha="xxxx">
  <espelhamento tipo="1">
    <id>1</id>
    <veilD>00000</veilD>
    <usuario>teste</usuario>
  </espelhamento>
</RequestCancelarEspelhamentoVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCancelarEspelhamentoVeiculo>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
```

```
</ResponseCancelarEspelhamentoVeiculo>
```

Espelhamentos Pendentes (RequestEspelhamentoPendenteVeiculo)

XML para requisição dos espelhamentos pendentes.

```
<RequestEspelhamentoPendenteVeiculo>  
    <login>?</login>  
    <senha>?</senha>  
</RequestEspelhamentoPendenteVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ResponseEspelhamentoPendenteVeiculo>  
    <EspelhamentoPendenteVeiculo>  
        <veilID>?</veilID>  
        <placa>?</placa>  
        <cmd>?</cmd>  
        <IE>?</IE>  
        <TIE>?</TIE>  
        <prop>?</prop>  
        <validade>?</validade>  
        <propCancelamento>?</propCancelamento>  
    </EspelhamentoPendenteVeiculo>  
</ResponseEspelhamentoPendenteVeiculo>
```

Descrição:

veilID: Identificador do veículo;

placa: Placa do veículo;

cmd: Permissão para o envio de comando ao veículo (0 – Não permitido ; 1 - Permitido);

IE: Permite possuir a inteligencia embarcada (0 – Não permitido ; 1 - Permitido);

TIE: Transferência da inteligência embarcada no ato do espelhamento (0 – Não transfere; 1 - Transfere);

Prop: Nome do Proprietário do equipamento;

validade: Indica a validade do espelhamento;

propCancelamento: Indica se o proprietário tem direito de cancelar o espelhamento (0 - Não Permitido ; 1 - Permitido);

OBS: Nestes casos ficará a critério do recebedor o cancelamento, ou até que o espelhamento expire o prazo de validade.

Exemplo

Requisição:

```
<RequestEspelhamentoPendenteVeiculo>
  <login>1234567890</login>
  <senha>teste</senha>
</RequestEspelhamentoPendenteVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEspelhamentoPendenteVeiculo>
  <EspelhamentoPendenteVeiculo>
    <veilD>00000</veilD>
    <placa>INT0000</placa>
    <cmd>1</cmd>
    <IE>1</IE>
    <TIE>0</TIE>
    <prop>Teste Integracao</prop>
    <validade>16/04/2016</validade>
    <propCancelamento>1</propCancelamento>
  </EspelhamentoPendenteVeiculo>
</ResponseEspelhamentoPendenteVeiculo>
```

Aceitar Espelhamento Pendente (RequestAREspelhamentoVeiculo)

XML para aceitar os espelhamentos pendentes.

```
<RequestAREspelhamentoVeiculo login="?" senha="?">
  <espelhamento tipo="1">
    <id?></id>
    <veilD?></veilD>
    <desc?></desc>
    <usuario?></usuario>
  </espelhamento>
</RequestAREspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente para aceitar ou rejeitar um espelhamento;

veilD: Identificador do veiculo;

usuario: Usuario que está aceitando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoVeiculo>
  <espelhamento>
    <id?</id>          [int]
    <status?</status> [int]
    <erro?</erro>     [int]
  </espelhamento>
</ResponseAREspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento aceito;

3 – espelhamento não aceito;

Erro: [Código de erro](#) caso haja um erro na aceitação do espelhamento de um veículo.

Em algumas situações o espelhamento será realizado e não será possível realizar a transferência de I.E.. Neste caso o status do espelhamento será 2 e na tag <erro> teremos descrito o erro da transferência .

 **Exemplo**

Requisição:

```
<RequestAREspelhamentoVeiculo login="1234567890" senha="xxxx">
```

```
<espelhamento tipo="1">
  <id>1</id>
  <veilD>00000</veilD>
  <desc>teste integracao</desc>
  <usuario>teste</usuario>
</espelhamento>
</RequestAREspelhamentoVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoVeiculo>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
</ResponseAREspelhamentoVeiculo>
```

Rejeitar Espelhamentos Pendentes (Cliente) - (RequestAREspelhamentoVeiculo)

XML para o cliente rejeitar os espelhamentos pendentes.

```
<RequestAREspelhamentoVeiculo login="?" senha="?">
  <espelhamento tipo="2">
    <id>?</id>
    <veilD>?</veilD>
    <usuario>?</usuario>
  </espelhamento>
</RequestAREspelhamentoVeiculo>
```

Descrição:

ID: Identificador seqüencial da aceitação/rejeição do espelhamento. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente para aceitar ou rejeitar um espelhamento;

veilD: Identificador do veiculo;

usuario: Usuario que está rejeitando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoVeiculo>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status> [int]
    <erro>?</erro>     [int]
  </espelhamento>
</ResponseAREspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento rejeitado;

3 – espelhamento não rejeitado;

Erro: [Código de erro](#) caso haja um erro na rejeição do espelhamento de um veículo.

 **Exemplo**

Requisição:

```
<RequestAREspelhamentoVeiculo login="1234567890" senha="xxxx">
  <espelhamento tipo="2">
    <id>1</id>
    <veilD>00000</veilD>
    <usuario>teste</usuario>
  </espelhamento>
</RequestAREspelhamentoVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoVeiculo>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
</ResponseAREspelhamentoVeiculo>
```

Rejeitar Espelhamentos Pendentes (Proprietário) - (RequestAREspelhamentoVeiculo)

XML para o proprietário rejeitar os espelhamentos pendentes dos seus veículos.

```
<RequestAREspelhamentoVeiculo login="?" senha="?">
  <espelhamento tipo="3">
    <id>?</id>
    <veilD>?</veilD>
    <cgccpf>?</cgccpf>
    <usuario>?</usuario>
  </espelhamento>
</RequestAREspelhamentoVeiculo>
```

Descrição:

ID: Identificador seqüencial da aceitação/rejeição do espelhamento. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente para aceitar ou rejeitar um espelhamento;

veilD: Identificador do veículo;

cgccpf: CNPJ/CPF do cliente para o qual a requisição de espelhamento será cancelada;

usuario: Usuario que está rejeitando o espelhamento;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAREspelhamentoVeiculo>
  <espelhamento>
    <id>?</id>          [int]
    <status>?</status>  [int]
    <erro>?</erro>      [int]
  </espelhamento>
</ResponseAREspelhamentoVeiculo>
```

Descrição:

id: Identificador seqüencial da aceitação/rejeição do espelhamento enviado pelo cliente;

status: Status do espelhamento:

2 – espelhamento rejeitado;

3 – espelhamento não rejeitado;

Erro: [Código de erro](#) caso haja um erro na rejeição do espelhamento de um veículo.

Exemplo

Requisição:

```
<RequestAREspelhamentoVeiculo login="1234567890" senha="xxxx">
  <espelhamento tipo="3">
    <id>1</id>
    <veilD>00000</veilD>
    <cgccpf>0123456789</cgccpf>
    <usuario>teste</usuario>
  </espelhamento>
</RequestAREspelhamentoVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseNovoEspelhamentoSpy>
  <espelhamento>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </espelhamento>
</ResponseNovoEspelhamentoSpy>
```

4.4 Requisição de Macro

Macro Grupo (RequestGrupoMacro)

XML para requisição dos grupos de macros disponíveis para o cliente.

```
<RequestGrupoMacro>
  <login>?</login>
  <senha>?</senha>
```

```
</RequestGrupoMacro>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseGrupoMacro>
  <GrupoMacro>
    <tgrID?></tgrID>           [int]
    <nm?></nm>                 [varchar(40)]
    <dGrupo?></dGrupo> (opc)[varchar(200)]
    <dt?></dt>                 [datetime]
    <tp?></tp>                 [tinyint]
    <emb?></emb> (opc)[bit]
    <tec?></tec>               [int]
  </GrupoMacro>
</ResponseGrupoMacro>
```

Descrição:

tgrID: Id do grupo;

nm: Nome do grupo;

dGrupo: Descrição do grupo;

dt: Data de criação do grupo;

tp: 1 – Da central para o veículo; 2 – Do veículo para a central; 3 - Híbrido (1 e 2); 4 - Do veículo para a central (SEQUENCIAL);

emb: Indica se o cliente pode embarcar este grupo (0 – não ; 1 – sim);

tec: Indica para qual tipo de teclado o grupo foi criado. (1 - Teclado Grande; 2 - Teclado Pequeno)

Nota

Para confirmar o tipo de teclado que o equipamento possui realizar a [requisição de AcessóriosVeículos](#), onde o acvID = 3 - Teclado Pequeno e 17 ou 66 - Teclado Grande

Exemplo

Requisição:

```
<RequestGrupoMacro>
  <login>12345678910</login>
  <senha>xxxx</senha>
</RequestGrupoMacro>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseGrupoMacro>
  <GrupoMacro>
    <tgrID>262</tgrID>
    <nm>PADRÃO CV</nm>
    <dt>2008-03-03T11:07:22.53-03:00</dt>
    <tp>1</tp>
    <emb>>false</emb>
    <tec>6</tec>
  </GrupoMacro>
  <GrupoMacro>
    <tgrID>264</tgrID>
    <nm>PADRÃO VC</nm>
    <dt>2008-03-03T11:36:04.297-03:00</dt>
    <tp>2</tp>
    <emb>>false</emb>
    <tec>6</tec>
  </GrupoMacro>
</ResponseGrupoMacro>
```

Macro Item (RequestItemMacro)

XML para requisição dos itens (macros) de cada grupo de macro disponível para o cliente.

```
<RequestItemMacro>
  <login>?</login>
  <senha>?</senha>
  <todosItens>0</todosItens>
</RequestItemMacro>
```

OBS: Para requisitar apenas os novos, em <todosItens> coloque "0" da forma como está apresentado acima, para requisitar todos, coloque "1", conforme abaixo:

```
<RequestItemMacro>
  <login>?</login>
  <senha>?</senha>
  <todosItens>1</todosItens>
</RequestItemMacro>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseItemMacro>
  <ItemMacro>
    <tfrID>?</tfrID>      [int]
    <tgrID>?</tgrID>     [int]
    <nm>?</nm>          [varchar(50)]
    <cod>?</cod>        [tinyint]
    <cont>?</cont>      [varchar(1024)]
    <tp>?</tp>          [tinyint]
    <exc>?</exc>        (opc)[tinyint]
    <dir>?</dir>        [tinyint]
    <aut>?</aut>        [tinyint]
  </ItemMacro>
</ResponseItemMacro>
```

Descrição:

tfrID: Código da Macro (macro);

tgrID: Código do grupo ao qual a macro pertence;

nm: Nome da macro;

cod: Código da macro no CB; **quando o conteúdo for nulo, significa que a macro é referente a macros seqüenciadas, se tornando assim opcional.**

cont: Conteúdo da macro;

tp: 1 – Macro Pré-Definida; 2 – Macro Formulário TD3X; 3 – Senha/Contra-senha;

exc: 0 – Ativo; 1 – Excluído;

dir: 0 - Do Veículo para a Central 1 - Da Central para o Veículo;

aut: 0 - Não Exige autenticação Biometrica/Senha 1 - Exige autenticação Biometrica/Senha;

Exemplo 1

Requisição:

```
<RequestItemMacro>
  <login>12345678910</login>
  <senha>xxxx</senha>
  <todosItens>0</todosItens>
</RequestItemMacro>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseItemMacro>
  <ItemMacro>
    <tfrID>38912</tfrID>
    <tgrID>2022</tgrID>
    <nm>POSSO VIAJAR?</nm>
    <cod>39</cod>
    <cont>POSSO VIAJAR?</cont>
    <tp>1</tp>
    <exc>0</exc>
    <dir>0</dir>
    <aut>0</aut>
  </ItemMacro>
  <ItemMacro>
    <tfrID>38913</tfrID>
    <tgrID>2022</tgrID>
    <nm>INICIO DE DESCARREGAMENTO</nm>
    <cod>40</cod>
    <cont>INICIO DE DESCARREGAMENTO</cont>
    <tp>1</tp>
    <exc>0</exc>
    <dir>0</dir>
    <aut>1</aut>
  </ItemMacro>
  <ItemMacro>
    <tfrID>38914</tfrID>
```

```

        <tgrID>2022</tgrID>
        <nm>TERMINO DE DESCARREGAMENTO</nm>
        <cod>41</cod>
        <cont>TERMINO DE DESCARREGAMENTO</cont>
        <tp>1</tp>
        <exc>0</exc>
        <dir>0</dir>
        <aut>0</aut>
    </ItemMacro>
</ResponseltemMacro>

```

Exemplo 2

Requisição:

```

<RequestItemMacro>
  <login>12345678910</login>
  <senha>xxxx</senha>
  <todosItens>1</todosItens>
</RequestItemMacro>

```

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseltemMacro>
  <ItemMacro>
    <tfrID>38734</tfrID>
    <tgrID>2006</tgrID>
    <nm>Faz nada</nm>
    <cod>7</cod>
    <cont>Faz Nada</cont>
    <tp>1</tp>
    <exc>0</exc>
    <dir>0</dir>
  </ItemMacro>
  <ItemMacro>
    <tfrID>38753</tfrID>
    <tgrID>2013</tgrID>
    <nm>TESTEEEEEEEEEEEEEEEE</nm>
    <cod>1</cod>
    <cont>2\1\13\5|3\1\Contra Senha:\4\6|4\1\TESTEEEEEEEEEEEEEE\15\1</
cont>
    <tp>3</tp>
    <exc>0</exc>

```

```

        <dir>0</dir>
    </ItemMacro>
    <ItemMacro>
        <tfrID>38754</tfrID>
        <tgrID>2014</tgrID>
        <nm>CHEGADA ABRIR PORTA </nm>
        <cod>1</cod>
        <cont>2\1\13\5|3\1\Contra Senha:\4\6|4\1\CHEGADA ABRIR PORTA \15
\1</cont>
        <tp>3</tp>
        <exc>0</exc>
        <dir>0</dir>
    </ItemMacro>
    <ItemMacro>
        <tfrID>38755</tfrID>
        <tgrID>2014</tgrID>
        <nm>INICIO DE VIAJEM </nm>
        <cod>2</cod>
        <cont>2\1\13\5|3\1\Contra Senha:\4\6|4\1\INICIO DE VIAJEM \15\1</
cont>
        <tp>3</tp>
        <exc>0</exc>
        <dir>0</dir>
    </ItemMacro>
</ResponseItemMacro>

```

Macro Embarcado (RequestGrupoMacroEmbarcado)

XML para requisição dos Grupos de Macros embarcados e ativos por veículos.

```

<RequestGrupoMacroEmbarcado>
    <login>?</login>
    <senha>?</senha>
</RequestGrupoMacroEmbarcado>

```

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseGrupoMacroEmbarcado>
    <GrupoMacroEmbarcado>
        <veilID>?</veilID> [int]
        <ativoCV>?</ativoCV> [int]
        <ativoVC>?</ativoVC> [int]
    </GrupoMacroEmbarcado>
</ResponseGrupoMacroEmbarcado>

```

```
        <vc1>?</vc1>      (opc)[int]
        <vc2>?</vc2>      (opc)[int]
        <vc3>?</vc3>      (opc)[int]
    </GrupoMacroEmbarcado>
</ResponseGrupoMacroEmbarcado>
```

Descrição:

veilD: Id do veículo

ativoCV: grupo de macro central-veículo embarcado e ativo

ativoVC: grupo de macro veículo-central embarcado e ativo

vc1: grupo de macro veículo-central embarcado

vc2: grupo de macro veículo-central embarcado

vc3: grupo de macro veículo-central embarcado

Obs: Quando os campos vc1, vc2 e vc3 possuírem o valor -1, considerar o campo como vazio

Exemplo

Requisição:

```
<RequestGrupoMacroEmbarcado>
  <login>12345678910</login>
  <senha>xxxx</senha>
</RequestGrupoMacroEmbarcado>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseGrupoMacroEmbarcado>
  <GrupoMacroEmbarcado>
    <veilD>655</veilD>
    <ativoCV>15</ativoCV>
    <ativoVC>15</ativoVC>
    <vc1>-1</vc1>
```

```

        <vc2>-1</vc2>
        <vc3>-1</vc3>
    </GrupoMacroEmbarcado>
    <GrupoMacroEmbarcado>
        <veilD>45872</veilD>
        <ativoCV>868</ativoCV>
        <ativoVC>1599</ativoVC>
        <vc1>1996</vc1>
        <vc2>388</vc2>
        <vc3>405</vc3>
    </GrupoMacroEmbarcado>
    <GrupoMacroEmbarcado>
        <veilD>46893</veilD>
        <ativoCV>387</ativoCV>
        <ativoVC>388</ativoVC>
        <vc1>-1</vc1>
        <vc2>-1</vc2>
        <vc3>-1</vc3>
    </GrupoMacroEmbarcado>
</ResponseGrupoMacroEmbarcado>

```

Macro Embarcado - Alterado (RequestGrupoMacroEmbarcado)

XML para requisição dos Grupos de Macros embarcados e ativos por veículos, apenas os que foram alterados.

```

<RequestGrupoMacroEmbarcado>
    <login>?</login>
    <senha>?</senha>
    <alterados>1</alterados>
</RequestGrupoMacroEmbarcado>

```

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseGrupoMacroEmbarcado>
    <GrupoMacroEmbarcado>
        <veilD>?</veilD> [int]
        <ativoCV>?</ativoCV> [int]
        <ativoVC>?</ativoVC> [int]
        <vc1>?</vc1> (opc)[int]
        <vc2>?</vc2> (opc)[int]
        <vc3>?</vc3> (opc)[int]
    </GrupoMacroEmbarcado>
</ResponseGrupoMacroEmbarcado>

```

```
</GrupoMacroEmbarcado>  
</ResponseGrupoMacroEmbarcado>
```

Descrição:

veilID: Id do veículo

ativoCV: grupo de macro central-veículo embarcado e ativo

ativoVC: grupo de macro veículo-central embarcado e ativo

vc1: grupo de macro veículo-central embarcado

vc2: grupo de macro veículo-central embarcado

vc3: grupo de macro veículo-central embarcado

Obs: Quando os campos vc1, vc2 e vc3 possuírem o valor -1, considerar o campo como vazio

 **Exemplo****Requisição:**

```
<RequestGrupoMacroEmbarcado>  
  <login>12345678910</login>  
  <senha>xxxx</senha>  
  <alterados>1</alterados>  
</RequestGrupoMacroEmbarcado>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ResponseGrupoMacroEmbarcado>  
  <GrupoMacroEmbarcado>  
    <veilID>655</veilID>  
    <ativoCV>15</ativoCV>  
    <ativoVC>15</ativoVC>  
    <vc1>-1</vc1>  
    <vc2>-1</vc2>  
    <vc3>-1</vc3>
```

```
</GrupoMacroEmbarcado>
<GrupoMacroEmbarcado>
  <veilID>45872</veilID>
  <ativoCV>868</ativoCV>
  <ativoVC>1599</ativoVC>
  <vc1>1996</vc1>
  <vc2>388</vc2>
  <vc3>405</vc3>
</GrupoMacroEmbarcado>
<GrupoMacroEmbarcado>
  <veilID>46893</veilID>
  <ativoCV>387</ativoCV>
  <ativoVC>388</ativoVC>
  <vc1>-1</vc1>
  <vc2>-1</vc2>
  <vc3>-1</vc3>
</GrupoMacroEmbarcado>
</ResponseGrupoMacroEmbarcado>
```

4.5 Requisição de Spy

Mensagens SPY (RequestMensagemSpy)

XML para requisição das mensagens enviadas pelos SPY's.

```
<RequestMensagemSpy>
  <login>?</login>
  <senha>?</senha>
  <mid>?</mid>
</RequestMensagemSpy>
```

Descrição:

mid: Id (mrvid) da última mensagem recebida dos spy, as próximas mensagens a serem recebidas virão a partir deste id; Neste campo sempre deve ser informado o mid da última mensagem recebida spy.

Os campos que possuem a indicação (opc) são opcionais, serão enviados somente quando seu valor for diferente do default. Por exemplo: os campos do tipo bit, toda vez que seu valor for 0 (zero) ele não será enviado.

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseMensagemSpy>
  <MensagemSpy>
    <mId>?</mId>           [bigint]
    <spyID>?</spyID>      [int]
    <serie>?</serie>      [int]
    <dtHora>?</dtHora>    [datetime]
    <lat>?</lat>          [char(9)]
    <lon>?</lon>          [char(9)]
    <mun>?</mun>          [varchar(50)]
    <uf>?</uf>            [char(2)]
    <rod>?</rod>          (opc)[varchar(15)]
    <rua>?</rua>          (opc)[varchar(50)]
    <vGPS>?</vGPS>        (opc)[int]
    <vMax>?</vMax>        (opc)[smallint]
    <lowBat>?</lowBat>    (opc)[bit]
    <evtD>?</evtD>        (opc)[bit]
    <evtA>?</evtA>        (opc)[bit]
    <evtUP>?</evtUP>      (opc)[bit]
    <relAt>?</relAt>      (opc)[bit]
    <gps>?</gps>          [char(1)]
    <jammer>?</jammer>   (opc)[bit]
  </MensagemSpy>
</ResponseMensagemSpy>

```

Descrição:

mId: Id da mensagem;*

spyID: Id do SPY;

serie: Número de série;

dtHora: Data hora da mensagem;

lat: Latitude;

lon: Longitude;

mun: Município referente posição;

uf: Estado referente posição;

rod: Rodovia referente posição;

rua: Rua referente posição;

vGPS: Velocidade do Spy;

vMax: Velocidade máxima do Spy ultrapassada;

lowBat: Identificador de bateria fraca;

evtD: Spy desacoplado;

evtA: Spy acoplado;

evtUP: Resposta de solicitação de última posição;

relAt: 0 - Store Report (Offline) 1 - Forward Report (Online);

gps: indica o status do GPS (A - Posição atualizada; B - Posição imprecisa; L - Posição indefinida);

*** O campo mld é o id da mensagem spy. Na próxima requisição de mensagem deverá ser informado o id da última mensagem recebida. O sistema só envia as mensagens das últimas 2 horas.**

Exemplo

Requisição:

```
<RequestMensagemSpy>
  <login>12345678910</login>
  <senha>xxxx</senha>
  <mld>4711009</mld>
</RequestMensagemSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMensagemSpy>
  <MensagemSpy>
    <mld>9515721</mld>
    <spyID>1763</spyID>
```

```
<serie>792</serie>
<dtHora>2009-07-22T11:10:07-03:00</dtHora>
<lat>-29,95551</lat>
<lon>-51,17637</lon>
<mun>CANOAS</mun>
<uf>RS</uf>
<rod>BR-116</rod>
<rua>R EUGENIO GASPAR CARNEIRO</rua>
<vGPS>0</vGPS>
<relAt>1</relAt>
</MensagemSpy>
<MensagemSpy>
  <mId>9515984</mId>
  <spyID>805</spyID>
  <serie>2033</serie>
  <dtHora>2009-07-22T11:12:06-03:00</dtHora>
  <lat>-29,69023</lat>
  <lon>-51,94948</lon>
  <mun>BOM RETIRO DO SUL</mun>
  <uf>RS</uf>
  <rod>RS-287</rod>
  <vGPS>85</vGPS>
  <relAt>1</relAt>
</MensagemSpy>
</ResponseMensagemSpy>
```

SPY's (RequestSpy)

XML para requisição das informações gerais dos Spy's.

```
<RequestSpy>
  <login>?</login>
  <senha>?</senha>
</RequestSpy>
```

OBS: Se quiser receber apenas o que foi alterado na próxima requisição, solicitar conforme abaixo:

```
<RequestSpy>
  <login>?</login>
  <senha>?</senha>
  <alterados>1</alterados>
</RequestSpy>
```

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseSpy>
  <Spy>
    <spyID?></spyID>      [int]
    <serie?></serie>      [int]
    <tCmd?></tCmd>        [int]
    <vMax?></vMax>        [int]
    <desc?></desc>        [varchar(50)]
    <eCmd?></eCmd> (opc)[bit]
    <dtBat?></dtBat> (opc)[datetime]
    <nBat?></nBat> (opc)[int]
    <sAcop?></sAcop> (opc)[bit]
    <tipo?></tipo>      [bit]
    <tel?></tel> (opc)[bigint]
    <eqp?></eqp>      [int]
  </Spy>
</ResponseSpy>

```

Descrição:

spyID: Id do Spy;

serie: Número de série;

tCmd: Tempo de reenvio de comando;

vMax: Velocidade maxima cadastrada;

desc: Descrição do Spy;

eCmd: Indica se o cliente possui permissão para enviar comando;

dtBat: Data do registro do nível da bateria;

nBat: Nível de bateria;

sAcop: 0 - SPY Desacoplado; 1 - SPY Acoplado

tipo: Tipo de Instalação: 0 - Portátil (Default) / 1 - Fixo (Sempre Energizado)

tel: Fone GSM do equipamento SpyTrack2

eqp: 7 referente a equipamento SpyTrack e 14 referente a equipamento SpyTrack2

Exemplo

Requisição:

```
<RequestSpy>
  <login>12345678910</login>
  <senha>xxxx</senha>
</RequestSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseSpy>
  <Spy>
    <spyID>4</spyID>
    <serie>11</serie>
    <tCmd>0</tCmd>
    <vMax>-1</vMax>
    <desc>Spy 002</desc>
    <eCmd>1</eCmd>
    <dtBat>2008-07-07T12:31:52.533-03:00</dtBat>
    <nBat>25</nBat>
    <tipo>0</tipo>
  </Spy>
  <Spy>
    <spyID>95</spyID>
    <serie>89</serie>
    <tCmd>0</tCmd>
    <vMax>-1</vMax>
    <desc>SPY89</desc>
    <eCmd>1</eCmd>
    <dtBat>2009-06-24T01:59:26.477-03:00</dtBat>
    <nBat>4</nBat>
    <sAcop>1</sAcop>
    <tipo>0</tipo>
  </Spy>
</ResponseSpy>
```

Alterar dados Spy (RequestAlterarDadosSpy)

XML para alteração dos dados dos Spy's.

```
<RequestAlterarDadosSpy login="?" senha="?">
  <spy>
    <id?</id>
    <spyID?</spyID>
    <desc?</desc>
    <obs?</obs>
    <usuario?</usuario>
  </spy>
</RequestAlterarDadosSpy>
```

Descrição:

id: Identificador seqüencial da alteração enviada pelo cliente. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente.

spyID: Identificador do dispositivo;

desc: Descrição do Spy;

obs: Observação cadastrada para o Spy;

usuario: Usuário que está realizando a alteração.;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAlterarDadosSpy>
  <spy>
    <id?</id>           [int]
    <status?</status>   [int]
    <erro?</erro>       [int]
  </spy>
</ResponseAlterarDadosSpy>
```

Descrição:

id: Identificador seqüencial da alteração enviada pelo cliente;

status: Status da alteração:

2 – alteração realizada;

3 – alteração não realizado;

Erro: [Código de erro](#) caso haja um erro na alteração dos dados de um dispositivo.

Exemplo

Requisição:

```
<RequestAlterarDadosSpy login="1234567890" senha="xxxx">
  <spy>
    <id>1</id>
    <spyID>95</spyID>
    <desc>teste integracao</desc>
    <obs>teste</obs>
    <usuario>teste</usuario>
  </spy>
</RequestAlterarDadosSpy>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAlterarDadosSpy>
  <spy>
    <id>1</id>
    <status>3</status>
    <erro>2</erro>
  </spy>
</ResponseAlterarDadosSpy>
```

4.6 Requisição de Veículo

Mensagens Veículos (RequestMensagemCB)

XML para requisição das mensagens enviadas pelos veículos.

```
<RequestMensagemCB>
  <login>?</login>
  <senha>?</senha>
  <mId>?</mId>
</RequestMensagemCB>
```

mid: Id (mrvid) da última mensagem recebida dos veículos, as próximas mensagens a serem recebidas virão a partir deste id; Neste campo sempre deve ser informado o mid da última mensagem recebida dos veículos, quando for a primeira vez, utilizar 1 como parametro.

OBS: CASO NÃO EXISTA EQUIPAMENTOS VINCULADOS A CONTA, NÃO REALIZAR CONSULTAS DE MENSAGENS, PODERÁ ACARRETAR EM BLOQUEIO AUTOMATICO DO USUARIO.

Os campos que possuem a indicação (opc) são opcionais, serão enviados somente quando seu valor for diferente do default. Por exemplo: os campos <evt> do tipo bit, toda vez que seu valor for 0 (zero) ele não será enviado. O campo <evt4> possui os calores -1,0,1 , sendo assim, toda vez que ele for 0 (zero – ignição desligada) ele não será enviado. Os campos <descMacro> e <tfrID> serão enviados somente quando a mensagem for do tipo macro. Os valores dos sensores de temperatura <st> serão enviados somente quando a mensagem possuir os valores destes sensores.

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMensagemCB>
  <MensagemCB>
    <mId?</mId>                [bigint]
    <veilD?</veilD>            [int]
    <dt?</dt>                  [datetime]
    <lat?</lat>                 [varchar(10)]
    <lon?</lon>                 [varchar(10)]
    <mun?</mun>                 [varchar(50)]
    <uf?</uf>                   [char(2)]
    <rod?</rod>                 (opc)[varchar(50)]
    <rua?</rua>                 (opc)[varchar(50)]
    <vel?</vel>                 [numeric(9)]
    <evt1?</evt1>              (opc)[bit]
    <evt2?</evt2>              (opc)[bit]
    <evt3?</evt3>              (opc)[bit]
    <evt4?</evt4>              (opc)[bit]
    <evt5?</evt5>              (opc)[bit]
    <evt6?</evt6>              (opc)[bit]
    <evt8?</evt8>              (opc)[bit]
    <evt9?</evt9>              (opc)[bit]
    <evt10?</evt10>            (opc)[bit]
    <evt11?</evt11>            (opc)[bit]
    <evt12?</evt12>           (opc)[bit]
    <evt13?</evt13>           (opc)[bit]
```

<evt14>?</evt14>	(opc)[bit]
<evt15>?</evt15>	(opc)[bit]
<evt16>?</evt16>	(opc)[bit]
<evt17>?</evt17>	(opc)[bit]
<evt18>?</evt18>	(opc)[bit]
<evt19>?</evt19>	(opc)[bit]
<evt20>?</evt20>	(opc)[bit]
<evt21>?</evt21>	(opc)[bit]
<evt23>?</evt23>	(opc)[bit]
<evt24>?</evt24>	(opc)[bit]
<evt25>?</evt25>	(opc)[bit]
<evt26>?</evt26>	(opc)[bit]
<evt27>?</evt27>	(opc)[bit]
<evt28>?</evt28>	(opc)[bit]
<evt29>?</evt29>	(opc)[bit]
<evt30>?</evt30>	(opc)[bit]
<evt31>?</evt31>	(opc)[bit]
<evt32>?</evt32>	(opc)[bit]
<evt33>?</evt33>	(opc)[bit]
<evt34>?</evt34>	(opc)[bit]
<evt35>?</evt35>	(opc)[bit]
<evt36>?</evt36>	(opc)[bit]
<evt37>?</evt37>	(opc)[bit]
<evt38>?</evt38>	(opc)[bit]
<evt39>?</evt39>	(opc)[bit]
<evt40>?</evt40>	(opc)[bit]
<evt41>?</evt41>	(opc)[bit]
<evt42>?</evt42>	(opc)[bit]
<evt43>?</evt43>	(opc)[bit]
<evt44>?</evt44>	(opc)[bit]
<evt45>?</evt45>	(opc)[bit]
<evt46>?</evt46>	(opc)[bit]
<evt47>?</evt47>	(opc)[bit]
<evt48>?</evt48>	(opc)[bit]
<evt49>?</evt49>	(opc)[bit]
<evt50>?</evt50>	(opc)[bit]
<evt51>?</evt51>	(opc)[bit]
<evt52>?</evt52>	(opc)[bit]
<evt53>?</evt53>	(opc)[bit]
<evt54>?</evt54>	(opc)[bit]
<evt55>?</evt55>	(opc)[bit]
<evt56>?</evt56>	(opc)[bit]
<evt57>?</evt57>	(opc)[bit]
<evt58>?</evt58>	(opc)[bit]
<evt59>?</evt59>	(opc)[bit]
<evt60>?</evt60>	(opc)[bit]
<evt61>?</evt61>	(opc)[bit]

<evt62>?</evt62>	(opc)[bit]
<evt63>?</evt63>	(opc)[bit]
<evt64>?</evt64>	(opc)[bit]
<evt65>?</evt65>	(opc)[bit]
<evt72>?</evt72>	(opc)[bit]
<evt73>?</evt73>	(opc)[bit]
<evt74>?</evt74>	(opc)[bit]
<evt75>?</evt75>	(opc)[bit]
<evt76>?</evt76>	(opc)[bit]
<evt77>?</evt77>	(opc)[bit]
<evt78>?</evt78>	(opc)[bit]
<evt79>?</evt79>	(opc)[bit]
<evt80>?</evt80>	(opc)[bit]
<evt81>?</evt81>	(opc)[bit]
<evt82>?</evt82>	(opc)[bit]
<evt83>?</evt83>	(opc)[bit]
<evt84>?</evt84>	(opc)[bit]
<evt85>?</evt85>	(opc)[bit]
<evt86>?</evt86>	(opc)[bit]
<evt87>?</evt87>	(opc)[bit]
<evt88>?</evt88>	(opc)[bit]
<evt89>?</evt89>	(opc)[bit]
<evt90>?</evt90>	(opc)[bit]
<evt91>?</evt91>	(opc)[bit]
<evt92>?</evt92>	(opc)[bit]
<dMac>?</dMac>	(opc)[varchar(2048)]
<tfrID>?</tfrID>	(opc)[int]
<st1>?</st1>	(opc)[int]
<st2>?</st2>	(opc)[int]
<st3>?</st3>	(opc)[int]
<ori>#</ori>	[tinyint]
<tpMsg>?</tpMsg>	[tinyint]
<dtinc>?</dtinc>	[datetime]
<evtG>?</evtG>	(opc)[tinyint]
<rpm>?</rpm>	(opc)[smallint]
<odm>?</odm>	(opc)[int]
<lt>?</lt>	(opc)[int]
<mLog>?</mLog>	(opc)[int]
<pcNome>?</pcNome>	(opc)[varchar(50)]
<mot>?</mot>	(opc)[varchar(50)]
<motID>?</motID>	(opc)[int]
<prNome>?</prNome>	(opc)[varchar(50)]
<tv1>?</tv1>	(opc)[bit]
<tv2>?</tv2>	(opc)[bit]
<tv3>?</tv3>	(opc)[bit]
<tv4>?</tv4>	(opc)[bit]

</MensagemCB>

</ResponseMensagemCB>

Descrição:

mid: Id da mensagem; *5

veilD: Id do veículo;

dt: Data e hora da mensagem

lat: Latitude

lon: Longitude

mun: Município referente posição

uf: Estado referente posição

rod: Rodovia referente posição

rua: Rua referente posição

vel: Velocidade do veículo *6

evt1: Buser de mensagens do veículo acionado, Alerta de cabine.

evt2: Sirene do veículo acionada

evt3: Veículo bloqueado

evt4: Veículo com ignição acionada *1

evt5: Botão de pânico acionado

evt6: Botão de aviso de cabine pressionado

evt8: Desengate da carreta 1

evt9: Botão de trava de quinta roda pressionado

evt10: Trava de baú está destravada;

evt11: Pisca alerta acionado

evt12: Porta do carona aberta

evt13: Porta do motorista aberta

evt14: Porta do baú esta aberta

evt15: Botão de trava de baú pressionado

evt16: Bateria violada

evt17: Cabo do velocímetro violado

evt18: Cabo da RS232 violado

evt19: EDN1 - botão de abertura de baú acionado (Ligth GSM 1 apenas)

evt20: EDN2 - Solicitação de Abertura de Porta de Baú (Ligth GSM 1 apenas)

evt21: EDP1 - Entrada Genérica 1 (Ligth GSM 1 apenas)

evt23: EVS1- Violação por Movimento Indevido, Movimentação sem Sinal GPS ou sem Comunicação GSM (Ligth GSM 1 apenas)

evt24: SDP1 - Trava/Destrava (Ligth GSM 1 apenas)

evt25: SDP2 - Saída Genérica (Ligth GSM 1 apenas)

evt26: Define se a mensagem informa valor de temperatura

evt27: Desengate da carreta 2

evt28: Violação de painel

evt29: Teclado Desconectado

evt30: Indica que a contra senha enviada contém aviso de SINISTRO.

evt31: Botão de Pânico Escondido Pressionado

evt32: Sensor da Janela do motorista

evt33: Sensor da Janela do carona

evt34: Velocidade máxima excedida (GPS)

evt35: Valor máximo de RPM ultrapassado

evt36: Tentativas de senha de operação excedida para desbloqueio (perfil)

operacional ,ignição)

evt37: Sensor de Temperatura1: Falha de comunicação

evt38: Sensor de Temperatura2: Falha de comunicação

evt39: Sensor de Temperatura3: Falha de comunicação

evt40: Entrada Ponto de Controle (smart)

evt41: Saida Ponto de Controle (smart)

evt42: Caixa Violada (Trailer GSM System)

evt43: Bateria Fraca (Trailer GSM System, Slim GSM 3)

evt44: Interferencia de Sinal GSM – Possibilidade Jammer (Trailer GSM System, Slim GSM 3, Smart)

evt45: Finger print desconectado (Ignição Biométrica Violada);

evt46: Digital com Sinistro e sem permissão;

evt47: Digital identificada;

evt48: Digital identificada com Sinistro;

evt49: Digital identificada e sem permissão;

evt50: Digital não identificada;

evt51: Tempo de autenticação de digital excedido;

evt52: Saída do Raio de Manobra;

evt53: Tempo de Manobra excedido;

evt54: Tempo de veículo parado excedido (Smart);

evt55: Botão de Manutenção Emergencial: Quando pressionado gerará uma mensagem com prioridade máxima de manutenção emergencial; **(evento exclusivo para Rail Patrol)**

evt56: Botão de Manutenção na Via: Quando pressionado gerará uma solicitação de agendamento de manutenção da via férrea no ponto dado pelas coordenadas geográficas presentes nesta mensagem; **(evento exclusivo para Rail Patrol)**

evt57: Botão de Reconhecimento de Alerta: Quando pressionado gerará uma mensagem informando que o fiscal de linha está ciente sobre um alerta que informa proximidade de uma composição; **(evento exclusivo para Rail Patrol)**

evt58: Manutenção do Hardware; **(evento exclusivo para Rail Patrol)**

evt59: Senha identificada;

evt60: Senha identificada com Sinistro;

evt61: Senha não identificada;

evt62: Senha identificada e sem permissão;

evt63: Senha com Sinistro e sem permissão;

evt64: Tempo de autenticação de senha excedido;

evt65: Digital não identificada - Modo Manobrista;

evt72: Velocidade Excedida Tacografo;

evt73: Caixa de Travas Violada;

evt74: Excedido Tempo de Porta de Baú 1 Aberta;

evt75: Excedido Tempo de Porta de Baú 2 Aberta;

evt76: Excedido Tempo de Porta de Baú 3 Aberta;

evt77: Excedido Tempo de Porta de Baú 4 Aberta;

evt78: Porta do Cofre Aberta;

evt79: Excedido Tempo de Porta do Cofre Aberta;

evt80: Porta de Baú 1 Violada;

evt81: Porta de Baú 2 Violada;

evt82: Porta de Baú 3 Violada;

evt83: Porta de Baú 4 Violada;

evt84: Abertura de Teclado;

evt85: Veículo na Chuva;

evt86: Movimento sem GPS (Híbrido) Violado;

evt87: Excedido Tempo de Porta do Motorista Aberta;

evt88: Excedido Tempo de Porta do Carona Aberta;

evt89: Desconexão de Sirene;

evt90: Auto Travamentos da(s) Trava(s) de Baú;

evt91: Alerta de Entrada em Ponto de Rotograma;

evt92: Alerta de Saida de Ponto de Rotograma;

dMac: Descrição da Macro recebida *4

tfrID: Código da Macro *4

st1: Valor do sensor de temperatura 1

st2: Valor do sensor de temperatura 2

st3: Valor do sensor de temperatura 3

ori: Origem da Mensagem:

1 - Mensagens transmitidas por Satelite

2 - Mensagens transmitidas por GSM/GPRS Híbrido

3 - Mensagens transmitidas por GSM/GPRS City

4 - Mensagens transmitidas por GSM/GPRS Light

5 - Mensagens transmitidas por GSM/GPRS Smart

6 - Mensagens transmitidas por Satelite Smart

7 - Mensagens transmitidas por LoRa

tpMsg: Tipo da Mensagem : *2

1 - Tabela de Estado / 2 - Tabela de Log de Posição / 3 – Macro / 4 - Mensagem de Antena (Satelite) / 5 - Tabela de Estado Extendida

dtInc: Data e Hora de gravação no servidor da TrucksControl

evtG: Indica qual dos eventos acima é o evento gerador da mensagem. *3

rpm: Valor do rpm;

odm: Valor do odômetro;

lt: Quantidade total de litros no tanque;

mLog: Indica se a mensagem é de log. *7

pcNome: Indica o nome do ponto de controle quando da entrada e saída de ponto de controle (evt's 40 e 41);

mot: indica o nome do motorista reconhecido nos eventos de digital e senha;

motID: indica o ID do motorista reconhecido nos eventos de digital e senha;

prNome: Indica o nome do ponto de rotograma quando da entrada e saída de ponto de rotograma (evt's 91 e 92);

tv1: indica se a trava 1 está violada, quando ocorrer evt14;

tv2: indica se a trava 2 está violada, quando ocorrer evt14;

tv3: indica se a trava 3 está violada, quando ocorrer evt14;

tv4: indica se a trava 4 está violada, quando ocorrer evt14;

***1** O valor deste campo pode ser -1, 0 ou 1, pois em determinados casos a mensagem provinda do computador de bordo ou da antena não tem condições de informar se a ignição está ligada realmente ou não. Portanto por convenção quando este campo for igual a -1, a mensagem não tem condições de informar se a ignição está ligada ou não, portanto o processamento desta informação deve ser ignorado. Caso o valor do campo seja 0 ou 1 o processamento desta informação não deve ser ignorado. Caso seja 0 a ignição está desligada, caso seja 1 a ignição está ligada, mas neste caso não significa que o veículo está em movimento, o veículo pode estar com a ignição ligada porém pode estar parado, por isso, recomendamos que quando o valor do campo for 1 verificar o campo que informa a velocidade do veículo se esta for maior que zero o veículo realmente está em movimento, caso seja igual a zero o veículo está parado.

***2** Os tipos de mensagem são: Log de posição (somente posição do veículo, nenhum evento foi gerado); Tabela de estado (posição e evento gerado); Macro (somente posição e macro, não recebendo nenhum evento, todos os eventos virão zerados); Poll (mensagem enviada pelo satélite); Tabela de estado estendida. Quando a mensagem for do tipo Macro, será recebida apenas a macro e a posição do equipamento, não será recebido evento ou velocidade.

Tabela de estado estendida é referente ao comando de tabela de estado estendida.

*3 Por default, o evento de botão de pânico (evt5) é gerador. Ele não virá indicado nesta coluna.

Nesta coluna virá indicada apenas a numeração do evento. Ex: evento gerador é a porta do motorista (Evt13), o campo (evtG) irá receber o número 13.

*4 O campo descMacro é referente ao conteúdo da macro recebida. O campo tfrid é referente ao código da macro recebida. Cada item de macro possui um código único. Por exemplo, a macro INICIO DE VIAGEM para o grupo 15 vai possuir um código (tfrid) x, e a macro INICIO DE VIAGEM para o grupo 388 vai possuir um código (tfrid) y.

*5 O campo mrvid é o id da mensagem. Na próxima requisição de mensagem deverá ser informado o id da última mensagem recebida. O sistema só envia as mensagens das últimas 2 horas.

*6 Quando o valor da velocidade for -1 quer dizer que o veículo não enviou o valor da velocidade, ela deve ser desconsiderada.

*7 Alguns equipamentos podem enviar mensagens de log, ou seja, quando perdem a comunicação GPRS, eles armazenam as mensagens, enviando posteriormente, quando a comunicação GPRS restabelece. As mensagens chegam com seus horários de origem.

OBS: EM CASOS DE MENSAGENS MACROS DE TEXTO LIVRE OU PREENCHIVEIS QUE SÃO ENVIADAS PELO EQUIPAMENTO, FORA DE COBERTURA GPRS, O SISTEMA TRARÁ UMA MASCARA DESTA MENSAGEM E QUANDO O MESMO VOLTAR A COMUNICAR VIA GPRS O CONTEUDO DA MACRO SERÁ CARREGADO. ASSIM É NECESSÁRIO QUE O SISTEMA AUTALIZE ESTA INFORMAÇÃO NO BANCO PARA QUE O CONTEUDO NÃO SEJA PERDIDO.

Exemplo

Requisição:

```
<RequestMensagemCB>
  <login>12345678910</login>
  <senha>xxxx</senha>
  <mld>1461813764</mld>
</RequestMensagemCB>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMensagemCB>
  <MensagemCB>
    <mld>2103713531</mld>
```

```
<veilID>26318</veilID>
<dt>2009-07-22T11:25:04-03:00</dt>
<lat>-23,5425</lat>
<lon>-49,2500</lon>
<mun>TAQUARITUBA</mun>
<uf>SP</uf>
<rod>SP-255</rod>
<vel>0</vel>
<ori>5</ori>
<tpMsg>1</tpMsg>
<dtInc>2009-07-22T11:25:07-03:00</dtInc>
<odm>88877</odm>
</MensagemCB>
<MensagemCB>
  <mld>2103715763</mld>
  <veilID>53429</veilID>
  <dt>2009-07-22T11:25:29-03:00</dt>
  <lat>-23,3015</lat>
  <lon>-51,1788</lon>
  <mun>LONDRINA</mun>
  <uf>PR</uf>
  <rod>BR-369</rod>
  <rua>AV MARINGA</rua>
  <vel>2</vel>
  <evt4>1</evt4>
  <ori>5</ori>
  <tpMsg>2</tpMsg>
  <dtInc>2009-07-22T11:25:34-03:00</dtInc>
</MensagemCB>
<MensagemCB>
  <mld>2103715855</mld>
  <veilID>26318</veilID>
  <dt>2009-07-22T11:25:32-03:00</dt>
  <lat>-23,5425</lat>
  <lon>-49,2500</lon>
  <mun>TAQUARITUBA</mun>
  <uf>SP</uf>
  <rod>SP-255</rod>
  <vel>0</vel>
  <evt13>true</evt13>
  <ori>5</ori>
  <tpMsg>1</tpMsg>
  <dtInc>2009-07-22T11:25:35-03:00</dtInc>
  <evtG>13</evtG>
  <odm>88877</odm>
</MensagemCB>
<MensagemCB>
```

```
<mId>2104215707</mId>
<veiID>45872</veiID>
<dt>2009-07-22T13:19:26-03:00</dt>
<lat>-23,3020</lat>
<lon>-51,1767</lon>
<mun>LONDRINA</mun>
<uf>PR</uf>
<rod>BR-369</rod>
<vel>-1</vel>
<evt4>-1</evt4>
<dMac>FIM DE VIAGEM FIM DE VIAGEM </dMac>
<tfrID>16881</tfrID>
<ori>5</ori>
<tpMsg>3</tpMsg>
<dtInc>2009-07-22T13:19:26-03:00</dtInc>
<odm>0</odm>
</MensagemCB>
</ResponseMensagemCB>
```

Veículos (RequestVeiculo)

XML para requisição das informações gerais dos veículos.

Os campos que possuem a indicação (opc) são opcionais, serão enviados somente quando seu valor for diferente do default.

```
<RequestVeiculo>
  <login>?</login>
  <senha>?</senha>
</RequestVeiculo>
```

OBS: Se quiser receber apenas o que foi alterado na próxima requisição, solicitar conforme abaixo:

```
<RequestVeiculo>
  <login>?</login>
  <senha>?</senha>
  <alterados>1</alterados>
</RequestVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseVeiculo>
  <Veiculo>
    <veilD?></veilD>          [int]
    <placa?></placa>          [char(7)]
    <vs?></vs>                [char(5)]
    <st1?></st1>              (opc)[bit]
    <st2?></st2>              (opc)[bit]
    <st3?></st3>              (opc)[bit]
    <tCmd?></tCmd>            [int]
    <tMac?></tMac>            (opc)[bit]
    <eCmd?></eCmd>           (opc)[bit]
    <tp?></tp>                [int]
    <ta?></ta>                [int]
    <eqp?></eqp>              [int]
    <mot?></mot>              (opc)[varchar(40)]
    <prop?></prop>            (opc)[varchar(50)]
    <dlE?></dlE>             (opc)[bit]
    <IE?></IE>                (opc)[bit]
    <loc?></loc>              (opc)[bit]
    <ident?></ident>
    <vManut?></vManut>
    <valEspelhamento?></valEspelhamento>
    <propCancelamento?></propCancelamento>
  </Veiculo>
</ResponseVeiculo>
```

Descrição:

veilD: Id do veículo;

placa: Placa do Veículo;

vs: Versão do Comp. de Bordo do Veículo;

st1: Indica se o veículo possui Sensor Temperatura 1;

st2: Indica se o veículo possui Sensor Temperatura 2;

st3: Indica se o veículo possui Sensor Temperatura 3;

tCmd: Tempo para reenvio de comando;

tMac: Indica se o veículo possui Teclado de Macro;

eCmd: Indica se o cliente possui permissão para enviar comando;

tp: Indica qual o Temporizador padrão;

ta: Indica qual o Temporizador Satelital atual;

eqp: Tipo do equipamento;

mot: Nome do motorista;

prop: indica se o cliente é proprietário do veículo;

dIE: indica se o cliente tem direito a inteligência embarcada do veículo.

IE: indica se o cliente está com a inteligência embarcada do equipamento.

loc: indica se o equipamento está sinalizado como "localizador" (0 - Não ; 1 - Sim).

ident: informa a identificação do equipamento, uma espécie de sinônimo, registrada para o veículo, no Enterprise.

vManut: indica se o veículo encontra-se em Manutenção, nestes casos, comandos não poderão ser enviados ao mesmo.

valEspelhamento: indica ao recebedor do espelhamento, qual a data de expiração do mesmo.

propCancelamento: indica se o proprietário tem direito de cancelar o espelhamento.

Exemplo

Requisição:

```
<RequestVeiculo>
  <login>12345678910</login>
  <senha>xxxx</senha>
</RequestVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseVeiculo>
  <Veiculo>
```

```

        <veilID>9305</veilID>
        <placa>DES0001</placa>
        <vs>1.00</vs>
        <tCmd>0</tCmd>
        <tMacro>>true</tMacro>
        <eCmd>>true</eCmd>
        <tp>5</tp>
        <ta>5</ta>
        <eqp>10</eqp>
        <loc>0</loc>
        <ident>CAMINHAO BAU</ident>
        <vManut>0</vManut>
        <valEspelhamento>16/04/2016</valEspelhamento>
        <propCancelamento>0</propCancelamento>
    </Veiculo>
    <Veiculo>
        <veilID>9306</veilID>
        <placa>DES0002</placa>
        <vs>4.00</vs>
        <tCmd>5</tCmd>
        <eCmd>>true</eCmd>
        <tp>5</tp>
        <ta>5</ta>
        <eqp>8</eqp>
        <dIE>>true</dIE>
        <loc>1</loc>
        <ident>TESTE</ident>
        <vManut>1</vManut>
        <valEspelhamento>16/04/2016</valEspelhamento>
        <propCancelamento>1</propCancelamento>
    </Veiculo>
</ResponseVeiculo>

```

Veículos Redundantes (RequestVeiculoRedundante)

XML para requisição das informações gerais dos veículos redundantes.

Os campos que possuem a indicação (opc) são opcionais, serão enviados somente quando seu valor for diferente do default.

```

<RequestVeiculoRedundante>
  <login>?</login>
  <senha>?</senha>
</RequestVeiculoRedundante>

```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseVeiculoRedundante>
  <veiculoRedundante>
    <veiID>?</veiID>
    <veiID2>?</veiID2>
  </veiculoRedundante>
</ResponseVeiculoRedundante>
```

Descrição:

veiID: Id do veículo;

veiID2: Id do equipamento redundancia;

OBS: para cada redundância que existir para o equipamento principal, será enviado uma tag adicional, como veiID2.

 **Exemplo****Requisição:**

```
<RequestVeiculoRedundante>
  <login>1234567890</login>
  <senha>xxxx</senha>
</RequestVeiculoRedundante>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseVeiculoRedundante>
  <veiculoRedundante>
    <veiID>49752</veiID>
    <veiID2>41101</veiID2>
  </veiculoRedundante>
  <veiculoRedundante>
    <veiID>00000</veiID>
    <veiID2>00001</veiID2>
  </veiculoRedundante>
</ResponseVeiculoRedundante>
```

☐ Dados dos Veículos (RequestDadosVeiculo)

XML para requisição de dados específicos dos veículos.

```
<RequestDadosVeiculo login="?" senha="?">
  <veiculo veilID="?"> [int]
    <vei>?</vei> [bit]
    <gMacro>?</gMacro> [bit]
    <gAtivo>?</gAtivo> [bit]
    <iMacro>?</iMacro> [bit]
  </veiculo>
</usuario>?</usuario>
</RequestDadosVeiculo>
```

Descrição:

veilID: Identificador do veículo;

vei: 0 ou 1, para receber dados referentes ao veículo; (placa, proprietario, versao de computador de bordo, etc)

gMacro: 0 ou 1, para receber os grupos de macro embarcados no equipamento;

gAtivo: 0 ou 1, para receber quais os grupos de macro ativos no equipamento;

iMacro: 0 ou 1, para receber quais são os itens de macro disponíveis no equipamento;

Obs: 0 - não solicita; 1 - solicita.

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseDadosVeiculo>
  <veiculo veilID="?" status="?">
    <vei>
      <placa>?</placa> [char(7)]
      <vs>?</vs> [char(5)]
      <st1>?</st1> (opc)[bit]
      <st2>?</st2> (opc)[bit]
      <st3>?</st3> (opc)[bit]
      <tCmd>?</tCmd> [int]
      <tMac>?</tMac> (opc)[bit]
```

```

        <eCmd>?</eCmd> (opc)[bit]
        <tp>?</tp> [int]
        <ta>?</ta> [int]
        <eqp>?</eqp> [int]
        <mot>?</mot> (opc)[varchar(40)]
        <prop>?</prop> (opc)[varchar(50)]
        <dlE>?</dlE> (opc)[bit]
        <IE>?</IE> (opc)[bit]
    </vei>
    <gMacro>
        <grupo>
            <tgrID>?</tgrID> [int]
            <nm>?</nm> [varchar(40)]
            <dGrupo>?</dGrupo> (opc)[varchar(200)]
            <dt>?</dt> [datetime]
            <tp>?</tp> [tinyint]
            <emb>?</emb> (opc)[bit]
        </grupo>
    </gMacro>
    <gAtivo>
        <cv>?</cv> [int]
        <vc>?</vc> [int]
    </gAtivo>
    <iMacro>
        <item>
            <tfrID>?</tfrID> [int]
            <tgrID>?</tgrID> [int]
            <nm>?</nm> [varchar(50)]
            <cod>?</cod> [tinyint]
            <cont>?</cont> [varchar(1024)]
            <tp>?</tp> [tinyint]
            <exc>?</exc> (opc)[tinyint]
            <dir>?</dir> [tinyint]
        </item>
    </iMacro>
</veiculo>
</ResponseDadosVeiculo>

```

Descrição:**veilD:** Identificador do veículo;**status:** Codigo de Erro para envio de Requisição**(VEI)****- placa:** Placa do Veículo;

- **vs:** Versão do Comp. de Bordo do Veículo;
- **st1:** Indica se o veículo possui Sensor Temperatura 1;
- **st2:** Indica se o veículo possui Sensor Temperatura 2;
- **st3:** Indica se o veículo possui Sensor Temperatura 3;
- **tCmd:** Tempo para reenvio de comando;
- **tMac:** Indica se o veículo possui Teclado de Macro;
- **eCmd:** Indica se o cliente possui permissão para enviar comando;
- **tp:** Indica qual o Temporizador padrão;
- **ta:** Indica qual o Temporizador Satelital atual;
- **eqp:** Tipo do equipamento;
- **mot:** Nome do motorista;
- **prop:** indica se o cliente é proprietário do veículo;
- **dIE:** indica se o cliente tem direito a inteligência embarcada do veículo.
- **IE:** indica se o cliente está com a inteligência embarcada do equipamento.

(GMACRO)

- **tgrID:** Id do grupo;
- **nm:** Nome do grupo;
- **dGrupo:** Descrição do grupo;
- **dt:** Data de criação do grupo;
- **tp:** 1 – Da central para o veículo; 2 – Do veículo para a central; 3 - Híbrido (1 e 2); 4 - Do veículo para a central (SEQUENCIAL);
- **emb:** Indica se o cliente pode embarcar este grupo (0 – não ; 1 – sim);

(GATIVO)

- **cv**: indica o sentido Central - Veículo;

- **vc**: indica o sentido Veículo - Central;

(IMACRO)

- **tfrID**: Código da Macro (macro);

- **tgrID**: Código do grupo ao qual a macro pertence;

- **nm**: Nome da macro;

- **cod**: Código da macro no CB; **quando o conteúdo for nulo, significa que a macro é referente a macros seqüenciadas, se tornando assim opcional.**

- **cont**: Conteúdo da macro;

- **tp**: 1 – Macro Pré-Definida; 2 – Macro Formulário TD3X; 3 – Senha/Contra-senha;

- **exc**: 0 – Ativo; 1 – Excluído;

- **dir**: 0 - Do Veículo para a Central 1 - Da Central para o Veículo;

Exemplo

Requisição:

```
<RequestDadosVeiculo login="1234567890" senha="xxxx">
  <veiculo veilID="45872">
    <vei>1</vei>
    <gMacro>1</gMacro>
    <gAtivo>1</gAtivo>
    <iMacro>1</iMacro>
  </veiculo>
  <usuario>teste</usuario>
</RequestDadosVeiculo>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseDadosVeiculo>
  <veiculo veilID="49752" status="0">
    <vei>
      <placa>STS0002</placa>
```

```
<vs>4.16</vs>
<st1>1</st1>
<st2>1</st2>
<st3>1</st3>
<tCmd>5</tCmd>
<tMac>1</tMac>
<eCmd>1</eCmd>
<tp>5</tp>
<ta>15</ta>
<eqp>6</eqp>
<mot>BANCADA STS</mot>
<dIE>1</dIE>
</vei>
<gMacro>
<grupo>
<tgrID>387</tgrID>
<nm>PADRAO ONIXSAT_CV</nm>
<dt>2008-03-27T10:21:40.047-03:00</dt>
<tp>1</tp>
<emb>1</emb>
</grupo>
<grupo>
<tgrID>6645</tgrID>
<nm>TREINAMENTO VICTOR </nm>
<dt>2011-01-13T16:38:19.527-02:00</dt>
<tp>2</tp>
</grupo>
</gMacro>
<gAtivo>
<cv>387</cv>
<vc>6645</vc>
</gAtivo>
<iMacro>
<item>
<tfrID>16867</tfrID>
<tgrID>387</tgrID>
<nm>MENSAGEM RECEBIDA - PRESSIONE ENTER</nm>
<cod>3</cod>
<cont>LIBERADO PARA VIAJAR</cont>
<tp>1</tp>
<dir>1</dir>
</item>
<item>
<tfrID>16875</tfrID>
<tgrID>387</tgrID>
<nm>MENSAGEM RECEBIDA - PRESSIONE ENTER</nm>
<cod>11</cod>
```

```
<cont>LIGAR PARA A CENTRAL ONIXSAT</cont>
<tp>1</tp>
<dir>1</dir>
</item>
<item>
<tfrID>138528</tfrID>
<tgrID>6645</tgrID>
<nm>TERMINO DE VIAGEM </nm>
<cod>2</cod>
<cont>TERMINO DE VIAGEM </cont>
<tp>1</tp>
<dir>0</dir>
</item>
</iMacro>
</veiculo>
</ResponseDadosVeiculo>
```

Alterar Dados dos Veículos (RequestAlterarDadosVeiculo)

XML para alteração de dados dos veículos. Estas alterações sofrerão efeito no Software New Enterprise também.

```
<RequestAlterarDadosVeiculo login="?" senha="?">
  <veiculo>
    <id>?</id>
    <veilID>?</veilID>
    <desc>?</desc>
    <motorista>?</motorista>
    <obs>?</obs>
    <usuario>?</usuario>
  </veiculo>
</RequestAlterarDadosVeiculo>
```

Descrição:

id: Identificador seqüencial da alteração enviada pelo cliente. Este valor deve ser único, nunca deve ser repetido, será o controle do cliente.

veilID: Identificador do veículo;

desc: Alteração da descrição do equipamento;

motorista: Alteração do motorista;

obs: Alteração da observação no equipamento

usuario: Usuario que efetuou a alteração;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAlterarDadosVeiculo>
  <veiculo>
    <id?</id>          [int]
    <status?</status>  [int]
    <erro?</erro>      [int]
  </veiculo>
</ResponseAlterarDadosVeiculo>
```

Descrição:

id: Identificador seqüencial da alteração enviada pelo cliente;

status: Status da alteração:

2 – alteração realizada;

3 – alteração não realizado;

Erro: [Código de erro](#) caso haja um erro na alteração dos dados de um veículo.

 **Exemplo**

Requisição:

```
<RequestAlterarDadosVeiculo login="1234567890" senha="xxx">
  <veiculo>
    <id>1</id>
    <veilD>45872</veilD>
    <desc>teste</desc>
    <motorista>Integracao</motorista>
    <obs>teste integracao</obs>
    <usuario>teste</usuario>
  </veiculo>
</RequestAlterarDadosVeiculo>
```

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseAlterarDadosVeiculo>
  <veiculo>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </veiculo>
</ResponseAlterarDadosVeiculo>

```

4.7 Requisição de Referência Entrega

Referência Entrega (RequestReferenciaEntrega)

XML para requisições dos pontos cadastrados como referencia de entrega

```

<RequestReferenciaEntrega>
  <login>?</login>
  <senha>?</senha>
</RequestReferenciaEntrega>

```

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseReferenciaEntrega>
  <ReferenciaEntrega>
    <veiID>?</veiID>           [int]
    <dtHora>?</dtHora>        [datetime]
    <latitude>?</latitude>    [varchar(9)]
    <longitude>?</longitude> [varchar(9)]
    <cnpj>?</cnpj>            [varchar(500)]
  </ReferenciaEntrega>
</ResponseReferenciaEntrega>

```

Descrição:

veiID: Id do veículo;

dtHora: Data e Hora da mensagem que gerou a Referencia Entrega

latitude: Latitude onde foi gerada a Referencia Entrega

longitude: Longitude onde foi gerada a Referencia Entrega

cnpj: CNPJ/Codigo da Referencia Entrega

Exemplo

Requisição:

```
<RequestReferenciaEntrega>
  <login>1234567890</login>
  <senha>xxxx</senha>
</RequestReferenciaEntrega>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseReferenciaEntrega>
  <ReferenciaEntrega>
    <veilD>12345</veilD>
    <dtHora>2011-01-10T03:30:10-02:00</dtHora>
    <latitude>-28,8753</latitude>
    <longitude>-49,4539</longitude>
    <cnpj>1234567890</cnpj>
  </ReferenciaEntrega>
</ResponseReferenciaEntrega>
```

4.8 Requisição de Status Comandos - I.E.

Status comandos, macros e Inteligencia Embarcada (RequestStatuscmie)

XML para requisição dos status dos comandos, macros e I.E. que foram enviados aos veículos.

```
<RequestStatuscmie>
  <login>?</login>
  <senha>?</senha>
</RequestStatuscmie>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatuscmie>
  <Statuscmie>
    <cmdID>?</cmdID>
    <Status>?</Status>
    <tipo>?</tipo>
  </Statuscmie>
</ResponseStatuscmie>
```

Descrição:

cmdID: Identificador do comando/macro/solicitação que foi enviado;

Status: Status do envio :

- 1 – a processar;
- 2 – comando processado com sucesso (servidor processou e enviou o comando ao veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – enviado (comando enviado, aguardando processamento do servidor (comando ainda não foi enviado para o equipamento));
- 5 – comando rejeitado no servidor;

Tipo: tipo da mensagem :

- 0 – comando;
- 1 – macro;
- 2 – embarque grupo macro;
- 3 – ativar / inativar grupo macro;
- 4 – limpeza dos grupos de macro;
- 5 – comando spy;
- 6 – Embarque Ponto de Controle;
- 7 – Limpeza Ponto de Controle
- 8 – Limpeza I. E.
- 9 - Embarque de Perfil Operacional
- 10 - Aplicar Perfil Padrão
- 16 - Embarque de Motorista
- 17 - Limpeza de Motorista

Exemplo

Requisição:

```
<RequestStatuscmie>
  <login>12345678910</login>
  <senha>xxxx</senha>
</RequestStatuscmie>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatuscmie>
  <Statuscmie>
    <cmdID>1</cmdID>
    <Status>2</Status>
    <tipo>4</tipo>
  </Statuscmie>
  <Statuscmie>
    <cmdID>5</cmdID>
    <Status>2</Status>
    <tipo>2</tipo>
  </Statuscmie>
  <Statuscmie>
    <cmdID>10</cmdID>
    <Status>2</Status>
    <tipo>0</tipo>
  </Statuscmie>
  <Statuscmie>
    <cmdID>38</cmdID>
    <Status>2</Status>
    <tipo>1</tipo>
  </Statuscmie>
  <Statuscmie>
    <cmdID>23</cmdID>
    <Status>2</Status>
    <tipo>1</tipo>
  </Statuscmie>
  <Statuscmie>
    <cmdID>27</cmdID>
    <Status>2</Status>
    <tipo>5</tipo>
  </Statuscmie>
</ResponseStatuscmie>
```

Obs: As operações abaixo terão requisição de status (requisição de Status de Comandos , Macros e IE Requisição de Status Comandos-IE), pois dependem da resposta do equipamento:

Comando veiculo

id: é um identificador seqüencial para o envio de comandos aos veículos, este valor nunca deve ser repetido, pois caso isto ocorra, ficará comprometido o status do comando;

Macro e macro livre

id: Identificador seqüencial para envio de macros. Este valor deve ser único, nunca deve ser repetido para o envio de macros (a solicitação de macro livre deve seguir está mesma seqüência).

Comando spy

id: Identificador seqüencial para envio dos comandos SPY's. Este valor deve ser único, nunca deve ser repetido para o envio de comandos SPY's, pois caso isto ocorra, ficará comprometido o status do comando;

Embarque de macros

id: Identificador seqüencial para embarque de macros. Este valor deve ser único, nunca deve ser repetido para o embarque de macros, pois caso isto ocorra, ficará comprometido o status do embarque;

Ativacao de macros

id: Identificador seqüencial para ativação dos grupos de macros. Este valor deve ser único, nunca deve ser repetido para o ativação dos grupos de macros, pois caso isto ocorra, ficará comprometido a ativação dos grupos de macros;

Limpeza de macros

id: Identificador seqüencial para limpeza dos grupos de macros. Este valor deve ser único, nunca deve ser repetido para o limpeza dos grupos de macros, pois caso isto ocorra, ficará comprometido a limpeza dos grupos de macros;

Embarque ponto de controle

id: Identificador seqüencial para embarque dos pontos de controle. Este valor deve ser único, nunca deve ser repetido.

Limpeza Ponto de Controle

id: Identificador seqüencial para limpeza dos pontos de controle. Este valor deve ser único, nunca deve ser repetido para o limpeza dos pontos de controle, pois caso isto ocorra, ficará comprometido a limpeza dos pontos de controle;

Limpeza IE

id: Identificador seqüencial para exclusão da I. E. . Este valor deve ser único, nunca deve ser repetido.

4.9 Requisição de Inteligência Embarcada



Inteligência Embarcada

- [Grupo de Macro](#)
- [Perfil Operacional](#)
- [Pontos de Controle](#)
- [Cerca Eletrônica](#)
- [Outras Requisições](#)

4.9.1 Grupo de Macro

Alguns equipamentos podem ter até 5 grupos de macros embarcados: 1 grupo no sentido central – veículo e 4 grupos no sentido veículo – central. Não é possível excluir um único grupo de macro do veículo, ao realizar a limpeza de macros, todos os grupos serão excluídos.

Embarque (RequestEmbarcarGrupoMacro)

XML para embarque de macros por veículos

```
<RequestEmbarcarGrupoMacro login="?" senha="?">
  <veilD?></veilD>           [int]
  <veilD?></veilD>           (opc)[int]
  <embarcarGrupoMacro>
    <id?></id>                [int]
    <tgrID?></tgrID>          [int]
    <ativar?></ativar>       [bit]
  </embarcarGrupoMacro>
  <usuario?></usuario>       [varchar (20)]
</RequestEmbarcarGrupoMacro>
```

Descrição:

veilD: identificador do veículo; pode ser incluído mais do que um veilid, para tanto basta adicionar mais uma tag <veilD> logo abaixo da primeira.

id: Identificador seqüencial para embarque de macros. Este valor deve ser único, nunca deve ser repetido para o embarque de macros, pois caso isto ocorra, ficará comprometido o status do embarque;

tgrid: grupo de macro a ser embarcado;

ativar: 0 – Indica que o grupo não será ativo; 1 – Indica que o grupo será ativo;

usuário: usuário que está embarcando o grupo;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarGrupoMacro status="?">
  <id value="?" status="?">
```

```
        <veilD status="?">?</veilD>
    </id>
</ResponseEmbarcarGrupoMacro>
```

Descrição:

ResponseEmbarcarGrupoMacro status="?": [Código de erro](#), caso haja um erro na requisição.

id value: Identificador do embarque de macros;;

status: Status do embarque :

- 2 – embarque processado com sucesso (servidor processou e embarcou o grupo no veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – embarcando (embarque enviado, aguardando processamento do servidor (embarque ainda não foi finalizado no equipamento));
- 5 – embarque rejeitado no servidor;

veilD status: [Código de erro](#), caso haja um erro no envio de um embarque para um veículo.

OBS: como podemos embarcar um grupo de macro para mais de um veículo, para cada veiid teremos um status de erro.

Nota

Caso o grupo de macro utilizado na tentativa de embarque não seja compatível com o teclado de macro que o equipamento possui, uma mensagem de erro será informada junto com o **código de erro 81**.

Para confirmar o tipo de teclado que o equipamento possui realizar a [requisição de AcessóriosVeículos](#), onde o acvID = 3 - Teclado Pequeno e 17 ou 66 - Teclado Grande

Exemplo

Requisição:

```
<RequestEmbarcarGrupoMacro login="12345678910" senha="xxxx">
```

```
<embarcarGrupoMacro>
  <id>5</id>
  <veiID>49752</veiID>
  <tgrID>387</tgrID>
  <ativar>1</ativar>
  <usuario>carlos</usuario>
</embarcarGrupoMacro>
<embarcarGrupoMacro>
  <id>6</id>
  <veiID>49752</veiID>
  <tgrID>388</tgrID>
  <ativar>1</ativar>
  <usuario>carlos</usuario>
</embarcarGrupoMacro>
<embarcarGrupoMacro>
  <id>7</id>
  <veiID>49752</veiID>
  <tgrID>941</tgrID>
  <ativar>0</ativar>
  <usuario>teste</usuario>
</embarcarGrupoMacro>
</RequestEmbarcarGrupoMacro>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarGrupoMacro>
  <embarcarGrupoMacro>
    <id>5</id>
    <status>4</status>
    <erro>0</erro>
  </embarcarGrupoMacro>
  <embarcarGrupoMacro>
    <id>6</id>
    <status>4</status>
    <erro>0</erro>
  </embarcarGrupoMacro>
  <embarcarGrupoMacro>
    <id>7</id>
    <status>4</status>
    <erro>0</erro>
  </embarcarGrupoMacro>
</ResponseEmbarcarGrupoMacro>
```

Ativação (RequestAtivarGrupoMacro)

XML para ativação dos grupos de macros por veículos

```
<RequestAtivarGrupoMacro login="?" senha="?">
  <ativarGrupoMacro>
    <id>?</id>           [int]
    <veiiD>?</veiiD>      [int]
    <tgrID>?</tgrID>       [int]
    <usuario>?</usuario>  [varchar(20)]
  </ativarGrupoMacro>
</RequestAtivarGrupoMacro>
```

Descrição :

id: Identificador seqüencial para ativação dos grupos de macros. Este valor deve ser único, nunca deve ser repetido para o ativação dos grupos de macros, pois caso isto ocorra, ficará comprometido a ativação dos grupos de macros;

veiiD: identificador do veículo;

tgrID: grupo de macro a ser ativado;

usuário: usuário que está ativando o grupo;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAtivarGrupoMacro>
  <ativarGrupoMacro>
    <id>?</id>           [int]
    <status>?</status>   [int]
    <erro>?</erro>       [int]
  </ativarGrupoMacro>
</ResponseAtivarGrupoMacro>
```

Descrição:

id: Identificador da ativação de macros;

status: Status da ativação :

- 2 – ativação processada com sucesso (servidor processou e ativou o grupo no veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);

- 4 – ativando (ativação enviada, aguardando processamento do servidor (ativação ainda não foi finalizada no equipamento));
 5 – ativação rejeitada no servidor;

Erro: [Código de erro](#), caso haja um erro no envio de uma ativação para um veículo.

Exemplo

Requisição:

```
<RequestAtivarGrupoMacro login="12345678910" senha="xxxx">
  <ativarGrupoMacro>
    <id>2</id>
    <veilD>49752</veilD>
    <tgrID>941</tgrID>
    <usuario>teste</usuario>
  </ativarGrupoMacro>
</RequestAtivarGrupoMacro>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseAtivarGrupoMacro>
  <ativarGrupoMacro>
    <id>2</id>
    <status>4</status>
    <erro>0</erro>
  </ativarGrupoMacro>
</ResponseAtivarGrupoMacro>
```

Limpeza - (RequestLimparGrupoMacro)

XML para limpeza dos grupos de macros por veículos

```
<RequestLimparGrupoMacro login="?" senha="?">
  <limparGrupoMacro>
    <id>?</id> [int]
    <veilD>?</veilD> [int]
    <usuario>?</usuario> [varchar(20)]
  </limparGrupoMacro>
</RequestLimparGrupoMacro>
```

Descrição:

id: Identificador seqüencial para limpeza dos grupos de macros. Este valor deve ser único, nunca deve ser repetido para o limpeza dos grupos de macros, pois caso isto ocorra, ficará comprometido a limpeza dos grupos de macros;

veilID: identificador do veículo;

usuário: usuário que está ativando o grupo;

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparGrupoMacro>
  <limparGrupoMacro>
    <id>?</id>           [int]
    <status>?</status>   [int]
    <erro>?</erro>       [int]
  </limparGrupoMacro>
</ResponseLimparGrupoMacro>
```

Descrição:

id: Identificador da limpeza de macros;

status: Status da limpeza:

2 – limpeza processada com sucesso (servidor processou e limpou o grupo no veículo);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – limpando(limpeza enviada, aguardando processamento do servidor (limpeza ainda não foi finalizada no equipamento));

5 – limpeza rejeitada no servidor;

Erro: [Código de erro](#), caso haja um erro no envio de uma ativação para um veículo.

 **Exemplo**

Requisição:

```
<RequestLimparGrupoMacro login="12345678910" senha="xxxx">
  <limparGrupoMacro>
    <id>2</id>
    <veilID>49752</veilID>
```

```
<usuario>teste</usuario>
</limparGrupoMacro>
</RequestLimparGrupoMacro>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparGrupoMacro>
  <limparGrupoMacro>
    <id>2</id>
    <status>4</status>
    <erro>0</erro>
  </limparGrupoMacro>
</ResponseLimparGrupoMacro>
```

4.9.2 Perfil Operacional

Obs: A criação de perfis operacionais estão disponíveis apenas no Software New Enterprise, via integração está disponível apenas a manipulação dos mesmos conforme descrito abaixo.

Perfis Operacionais Existentes (RequestPerfilOperacional)

XML para requisição de perfis operacionais do cliente.

```
<RequestPerfilOperacional>
  <login>?</login>
  <senha>?</senha>
</RequestPerfilOperacional>
```

Descrição:

Login: CGC/cpf do cliente integração ;

senha: senha do usuário do integração;

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponsePerfilOperacional>
  <PerfilOperacional>
    <poID?</poID> [int]
    <nm?</nm> [varchar(50)]
    <des? </des> [varchar(5000)]
  </PerfilOperacional>
</ResponsePerfilOperacional>
```

Descrição:

poID: identificador do perfil operacional

nm: nome do perfil operacional

des: descrição do perfil operacional

 **Exemplo**

Requisição:

```
<RequestPerfilOperacional>
  <login>1234567890</login>
  <senha>xxxx</senha>
</RequestPerfilOperacional>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponsePerfilOperacional>
  <PerfilOperacional>
    <poID>123</poID>
    <nm>Teste Integracao</nm>
    <des>Perfil criado para teste do Webservice</des>
  </PerfilOperacional>
</ResponsePerfilOperacional>
```

 **Perfil Operacional Embarcado
(RequestPerfilOperacionalEmbarcado)**

XML para requisição de perfis operacionais embarcados por veiculo do cliente.

```
<RequestPerfilOperacionalEmbarcado>
  <login>?</login>
  <senha>?</senha>
</RequestPerfilOperacionalEmbarcado>
```

Descrição:

Login: CGC/cpf do cliente integração ;

senha: senha do usuário do integração;

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponsePerfilOperacionalEmbarcado>
  <PerfilOperacionalEmbarcado veilID=?">
    <polD>?</polD> [int]
    <polD ativo="1">?</polD> [int]
  </PerfilOperacionalEmbarcado>
  <PerfilOperacionalEmbarcado>
    <polD ativo="1">?</polD> [int]
  </PerfilOperacionalEmbarcado>
  <Descricao>
    <perfil polD=?">?</perfil>
    <perfil polD=?">?</perfil>
    <perfil polD=?">?</perfil>
  </Descricao>
</ResponsePerfilOperacionalEmbarcado>
```

Descrição:

veilD: id do veiculo

polD: identificador do perfil operacional embarcado no veiculo

ativo: indica qual dos perfis embarcados está ativo no momento

descrição: cada perfil operacional (polD) será descrito apenas uma vez para nao gerar informações duplicadas.

OBS: cada equipamento Smart (GPRS ou Híbrido) que estiverem espelhados para a conta receberá esta informação de qual perfil está embarcado e qual está ativo. A tag que separa as informações por veículo é a veilID

Exemplo

Requisição:

```
<RequestPerfilOperacionalEmbarcado>
  <login>1234567890</login>
  <senha>xxxx</senha>
</RequestPerfilOperacionalEmbarcado>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponsePerfilOperacionalEmbarcado>
  <PerfilOperacionalEmbarcado veilID="00000">
    <polD>595</polD>
    <polD>100101</polD>
    <polD>100102</polD>
    <polD>100103</polD>
    <polD>100104</polD>
    <polD ativo="1">100102</polD>
  </PerfilOperacionalEmbarcado>
  <PerfilOperacionalEmbarcado veilID="45872">
    <polD ativo="1">15398</polD>
  </PerfilOperacionalEmbarcado>
  <Descricao>
    <perfil polD="595">PADRÃO ONIXSAT</perfil>
    <perfil polD="100101">pernoite</perfil>
    <perfil polD="100102">inicio de viagem</perfil>
    <perfil polD="100103">parada</perfil>
    <perfil polD="100104">chegada no cliente</perfil>
    <perfil polD="15398">PADRÃO TESTE</perfil>
  </Descricao>
</ResponsePerfilOperacionalEmbarcado>
```

Embarque de Perfil Operacional (RequestEmbarcarPerfilOperacional)

XML para requisição de embarque de perfis operacionais.

```
<RequestEmbarcarPerfilOperacional login="?" senha="?">
  <veilD?></veilD>
  <embarcarPerfilOperacional>
    <id?></id>
    <poID?></poID>
  </embarcarPerfilOperacional>
  <usuario?></usuario>
</RequestEmbarcarPerfilOperacional>
```

Descrição:

veilD: Identificador do equipamento ;

id: Identificador seqüencial para embarque dos perfis de operação. Este valor deve ser único, nunca deve ser repetido.;

poID: identificador do perfil operacional

usuário: usuário que está embarcando

OBS: poderá ser embarcado mais de um perfil para mais de um veículo ao mesmo tempo, basta incluir mais uma tag de veilD logo abaixo da primeira e abrir quantas tags <embarcarPerfilOperacional> forem necessárias de acordo como o numero de perfis que se deseja embarcar.

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarPerfilOperacional status="0">
  <id value="?" status="?">
    <veilD status="?"></veilD>
  </id>
</ResponseEmbarcarPerfilOperacional>
```

Descrição:

ResponseEmbarcarPerfilOperacional status="?": [Código de erro](#), caso haja um erro na requisição.

id value: Identificador do embarque de ponto de controle;

status: Status do embarque :

2 – embarque processado com sucesso (servidor processou e embarcou o grupo no veículo);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – embarcando (embarque enviado, aguardando processamento do servidor (embarque ainda não foi finalizado no equipamento));

5 – embarque rejeitado no servidor;

veilID status: [Código de erro](#), caso haja um erro no envio de um embarque para um veículo.

Exemplo

Requisição:

```
<RequestEmbarcarPerfilOperacional login="1234567890" senha="xxxx">
  <veilID>45872</veilID>
  <embarcarPerfilOperacional>
    <id>1</id>
    <polD>595</polD>
  </embarcarPerfilOperacional>
  <usuario>teste</usuario>
</RequestEmbarcarPerfilOperacional>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarPerfilOperacional status="0">
  <id value="1" status="4">
    <veilID status="0">45872</veilID>
  </id>
</ResponseEmbarcarPerfilOperacional>
```

Limpeza de Perfil Operacional (RequestLimparPerfilOperacional)

XML para limpeza dos perfis operacionais por veículos

```
<RequestLimparPerfilOperacional login="?" senha="?">
  <limparPerfilOperacional>
    <id>?</id> [int]
    <veilID>?</veilID> [int]
    <usuario>?</usuario> [varchar(20)]
```

```
</limparPerfilOperacional>
</RequestLimparPerfilOperacional>
```

Descrição:

id: Identificador seqüencial para limpeza dos perfis operacionais. Este valor deve ser único, nunca deve ser repetido para o limpeza dos perfis operacionais, pois caso isto ocorra, ficará comprometido a limpeza dos perfis operacionais;

veilID: identificador do veículo;

usuário: usuário que está realizando a limpeza;

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparPerfilOperacional>
  <limparPerfilOperacional>
    <id>?</id> [int]
    <status>?</status> [int]
    <erro>?</erro> [int]
  </limparPerfilOperacional>
</ResponseLimparPerfilOperacional>
```

Descrição:

id: Identificador da limpeza de perfis;

status: Status da limpeza:

- 2 – limpeza processada com sucesso (servidor processou e limpou o(os) perfil no veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – limpando(limpeza enviada, aguardando processamento do servidor (limpeza ainda não foi finalizada no equipamento));
- 5 – limpeza rejeitada no servidor;

Erro: [Código de erro](#), caso haja um erro no envio de limpeza de perfis em um veículo.

Exemplo

Requisição:

```
<RequestLimparPerfilOperacional login="1234567890" senha="xxxx">
```

```
<limparPerfilOperacional>
  <id>1</id>
  <veilD>45872</veilD>
  <usuario>teste</usuario>
</limparPerfilOperacional>
</RequestLimparPerfilOperacional>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparPerfilOperacional>
  <limparPerfilOperacional>
    <id>1</id>
    <status>4</status>
    <erro>0</erro>
  </limparPerfilOperacional>
</ResponseLimparPerfilOperacional>
```

4.9.3 Pontos de Controle

Pontos de Controle Existentes (Modelo) - (RequestPontoControle)

XML para requisição dos pontos de controle existentes.

```
<RequestPontoControle>
  <login>?</login>
  <senha>?</senha>
</RequestPontoControle>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponsePontoControle>
  <PontoControleModelo>
    <pcmlD>?</pcmlD>      [int]
    <nm>?</nm>            [varchar(50)]
    <des>?</des>          [varchar(500)]
```

```
        <lat>?</lat>           [varchar(9)]
        <lon>?</lon>           [varchar(9)]
        <raio>?</raio>        [real]
        <dt>?</dt>            [datetime]
    </PontoControleModelo>
</ResponsePontoControle>
```

Descrição:

pcmID: Identificador do ponto de controle modelo

nm: Nome do ponto de controle;

desc: descrição do ponto de controle;

lat: latitude;

lon: Longitude;

raio: raio do ponto a partir da latitude e longitude.

dt: Data de criação do Ponto de Controle

Exemplo

Requisição:

```
<RequestPontoControle>
    <login>12345678910</login>
    <senha>xxxx</senha>
</RequestPontoControle>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponsePontoControle>
    <PontoControle>
        <pcmID>1070</pcmID>
        <nome>ws</nome>
        <desc>web service</desc>
        <lat>-23,2873</lat>
        <lon>-51,1438</lon>
        <raio>150</raio>
        <dt>2010-05-13T15:09:40.653-03:00</dt>
```

```

</PontoControle>
<PontoControle>
  <pcmID>1086</pcmID>
  <nome>teste teste II</nome>
  <desc>testestsetset</desc>
  <lat>-23,3895</lat>
  <lon>-51,9163</lon>
  <raio>160</raio>
  <dt>2012-05-04T15:09:40.653-03:00</dt>
</PontoControle>
</ResponsePontoControle>

```

Criar Pontos de Controle (Modelo) - (RequestCriarPontoControle)

XML para criação dos pontos de controle.

```

<RequestCriarPontoControle login="12345678910" senha="xxxx">
  <criarPontoControle>
    <id>?</id>
    <nm>?</nm>
    <des>?</des>
    <lat>?</lat>
    <lon>?</lon>
    <raio>?</raio>
    <usuario>?</usuario>
  </criarPontoControle>
</RequestCriarPontoControle>

```

Descrição:

Id: Identificador seqüencial para criação dos pontos de controle. Este valor deve ser único, nunca deve ser repetido.

Nm: Nome do ponto de controle;

Des: Descrição do grupo;

Lat: Latitude (formato em graus, exemplo: -23,9087);

Lon: Longitude (formato em graus);

Raio: Raio (em metros, com valor mínimo de 150 m).

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCriarPontoControle>
  <criarPontoControle>
    <id>?</id>          [int]
    <status>?</status> [int]
    <erro>?</erro>     [int]
  </criarPontoControle>
</ResponseCriarPontoControle>
```

id: Identificador da criação do ponto de controle;

status: Status da criação :

2 – ponto de controle criado com sucesso;

3 – rejeitado (erro de sintaxe ou falta de credenciais);

Erro: [Código de erro](#), caso haja um erro na criação de um ponto de controle.

 **Exemplo****Requisição:**

```
<RequestCriarPontoControle login="12345678910" senha="xxx">
  <criarPontoControle>
    <id>5</id>
    <nome>teste ws</nome>
    <des>teste do pc ws</desc>
    <lat>-23,3845</lat>
    <lon>-51,9163</lon>
    <raio>189</raio>
    <usuario>carlos</usuario>
  </criarPontoControle>
</RequestCriarPontoControle>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCriarPontoControle>
  <criarPontoControle>
    <id>5</id>
    <status>2</status>
    <erro>0</erro>
  </criarPontoControle>
```

```
</ResponseCriarPontoControle>
```

■ Embarcar Ponto de Controle (Modelo) - (RequestEmbarcarPontoControle)

XML para embarque dos pontos de controle modelo

```
<RequestEmbarcarPontoControle login="?" senha="?">
  <veilID?</veilID> [int]
  <embarcarPontoControle tipo="1">
    <id?</id> [int]
    <pcmlID?</pcmlID> [int]
    <polDIn?</polDIn> [int]
    <polDOut?</polDOut> [int]
    <tgrIDIn?</tgrIDIn> (opc)[int]
    <tgrIDOut?</tgrIDOut> (opc)[int]
    <fixo?</fixo> [bit]
  </embarcarPontoControle>
  <usuario?</usuario> [varchar (20)]
</RequestEmbarcarPontoControle>
```

Descrição:

veilID: Identificador do veículo; pode ser embarcado para mais de um veículo ao mesmo tempo, para tanto, basta incluir mais uma tag veilID.

Id: Identificador seqüencial para embarque dos pontos de controle. Este valor deve ser único, nunca deve ser repetido.

pcmlID: Identificador do ponto de controle modelo;

polDIn: perfil operacional de entrada em ponto de controle, somente serão aceitos perfis já embarcados no equipamento.

polDOut: perfil operacional de saída do ponto de controle, somente serão aceitos perfis já embarcados no equipamento.

tgrIDIn: grupo de macro que será ativo na entrada do ponto de controle, somente serão aceitos grupos de macro já embarcados no equipamento.

tgrIDOut: grupo de macro que será ativo na saída do ponto de controle, somente serão aceitos grupos de macro já embarcados no equipamento.

fixo: se ele é fixo ou não, quando da requisição de limpeza de ponto de controle o ponto estiver marcado como fixo, não será excluído, sendo limpo apenas os pontos que não são fixos;

usuario: Usuário que está embarcando;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarPontoControle status="?">
  <id value="?" status="?">
    <veilID status="?">?</veilID>
  </id>
</ResponseEmbarcarGrupoMacro>
```

ResponseEmbarcarGrupoMacro status="?": [Código de erro](#), caso haja um erro na requisição.

id value: Identificador do embarque de ponto de controle;

status: Status do embarque :

2 – embarque processado com sucesso (servidor processou e embarcou o ponto no veículo);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – embarcando (embarque enviado, aguardando processamento do servidor (embarque ainda não foi finalizado no equipamento));

5 – embarque rejeitado no servidor;

veilID status: [Código de erro](#), caso haja um erro no envio de um embarque para um veículo. ;

OBS: como podemos embarcar um ponto de controle para mais de um veículo, para cada veiid teremos um status de erro.

Exemplo

Requisição:

```
<RequestEmbarcarPontoControle login="1234567890" senha="xxx">
  <veilID>45872</veilID>
  <veilID>00000</veilID>
  <embarcarPontoControle tipo="1">
    <id>1</id>
```

```

        <pcmlID>1070</pcmlID>
        <polDIn>595</polDIn>
        <polDOut>595</polDOut>
        <tgrlDIn>11696</tgrlDIn>
        <tgrlDOut>11696</tgrlDOut>
        <fixo>1</fixo>
    </embarcarPontoControle>
    <usuario>teste</usuario>
</RequestEmbarcarPontoControle>

```

Reposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarPontoControle status="0">
    <id value="1" status="4">
        <veilD status="0">45872</veilD>
        <veilD status="0">00000</veilD>
    </id>
</ResponseEmbarcarGrupoMacro>

```

■ Embarcar Ponto de Controle (Novo) - (RequestEmbarcarPontoControle)

XML para embarque dos pontos de controle novos. a criação de ponto é feita no momento do embarque.

```

<RequestEmbarcarPontoControle login="?" senha="?">
    <veilD?></veilD> [int]
    <embarcarPontoControle tipo="2">
        <id?></id> [int]
        <nm?></nm> [varchar (50)]
        <des?></des> [varchar (200)]
        <lat?></lat> [float]
        <lon?></lon> [float]
        <raio?></raio> [float]
        <polDIn?></polDIn> [int]
        <polDOut?></polDOut> [int]
        <tgrlDIn?></tgrlDIn> (opc)[int]
        <tgrlDOut?></tgrlDOut> (opc)[int]
        <fixo?></fixo> [bit]
    </embarcarPontoControle>
    <usuario?></usuario> [varchar (20)]
</RequestEmbarcarPontoControle>

```

Descrição:

veilD: Identificador do veículo; pode ser embarcado para mais de um veículo ao mesmo tempo, para tanto, basta incluir mais uma tag veilD.

Id: Identificador seqüencial para embarque dos pontos de controle. Este valor deve ser único, nunca deve ser repetido.

nm: nome do novo ponto de controle

des: descrição do novo ponto de controle;

lat: latitude de referencia para o novo ponto de controle;

lon: longitude referencia para o novo ponto de controle;

raio: raio desejado em metros;

poIDIn: perfil operacional de entrada em ponto de controle, somente serão aceitos perfis já embarcados no equipamento.

poIDOut: perfil operacional de saída do ponto de controle, somente serão aceitos perfis já embarcados no equipamento.

tgrIDIn: grupo de macro que será ativo na entrada do ponto de controle, somente serão aceitos grupos de macro já embarcados no equipamento.

tgrIDOut: grupo de macro que será ativo na saída do ponto de controle, somente serão aceitos grupos de macro já embarcados no equipamento.

fixo: se ele é fixo ou não, quando da requisição de limpeza de ponto de controle o ponto estiver marcado como fixo, não será excluído, sendo limpadado apenas os pontos que não são fixos;

usuario: Usuário que está embarcando;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarPontoControle status="?">
  <id value="?" status="?">
    <veilD status="?"?</veilD>
  </id>
```

</ResponseEmbarcarGrupoMacro>

ResponseEmbarcarGrupoMacro status="?": [Código de erro](#), caso haja um erro na requisição.

id value: Identificador do embarque de ponto de controle;

status: Status do embarque :

2 – embarque processado com sucesso (servidor processou e embarcou o ponto no veículo);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – embarcando (embarque enviado, aguardando processamento do servidor (embarque ainda não foi finalizado no equipamento));

5 – embarque rejeitado no servidor;

veilD status: [Código de erro](#), caso haja um erro no envio de um embarque para um veículo.

OBS: como podemos embarcar um ponto de controle para mais de um veículo, para cada veiid teremos um status de erro.

Exemplo

Requisição:

```
<RequestEmbarcarPontoControle login="1234567890" xxxx="?">
  <veilD>45872</veilD>
  <embarcarPontoControle tipo="2">
    <id>1</id>
    <nm>Teste WebService</nm>
    <des>Teste para integracao</des>
    <lat>-23,45687</lat>
    <lon>-47,38569</lon>
    <raio>30</raio>
    <polDIn>595</polDIn>
    <polDOut>595</polDOut>
    <tgrIDIn>11696</tgrIDIn>
    <tgrIDOut>11696</tgrIDOut>
    <fixo>1</fixo>
  </embarcarPontoControle>
  <usuario>teste</usuario>
</RequestEmbarcarPontoControle>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarPontoControle status="0">
  <id value="1" status="4">
    <veilD status="0">45872</veilD>
  </id>
</ResponseEmbarcarGrupoMacro>
```

Excluir Ponto de Controle (RequestExcluirPontoControle)

XML para exclusão de pontos de controle existentes.

```
<RequestExcluirPontoControle login="?" senha="?">
  <excluirPontoControle>
    <id>?</id>
    <pcmlD>?</pcmlD>
    <usuario>?</usuario>
  </excluirPontoControle>
</RequestExcluirPontoControle>
```

Descrição:

Id: Identificador seqüencial para exclusão dos pontos de controle. Este valor deve ser único, nunca deve ser repetido.

pcmlD: Identificador do ponto de controle;

usuario: Usuário que está embarcando;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseExcluirPontoControle>
  <excluirPontoControle>
    <id>?</id> [int]
    <status>?</status> [int]
    <erro>?</erro> [int]
  </excluirPontoControle>
</ResponseExcluirPontoControle>
```

Descrição:

id: Identificador da exclusão dos pontos de controle;

status: Status da exclusão :

2 – ponto de controle excluído com sucesso;

3 – rejeitado (erro de sintaxe ou falta de credenciais);

Erro: [Código de erro](#), caso haja um erro na criação de um ponto de controle.

Exemplo

Requisição:

```
<RequestExcluirPontoControle login="1234567890" senha="xxxx">
  <excluirPontoControle>
    <id>1</id>
    <pcmlID>100</pcmlID>
    <usuario>teste</usuario>
  </excluirPontoControle>
</RequestExcluirPontoControle>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseExcluirPontoControle>
  <excluirPontoControle>
    <id>1</id>
    <status>2</status>
    <erro>0</erro>
  </excluirPontoControle>
</ResponseExcluirPontoControle>
```

Limpeza dos Pontos de Controle (RequestLimparPontoControle)

XML para limpeza dos pontos de controle embarcados em um veículo.

```
<RequestLimparPontoControle login="?" senha="?">
  <limparPontoControle>
    <id>?</id> [int]
    <veilID>?</veilID> [int]
    <usuario>?</usuario> [varchar(20)]
  </limparPontoControle>
```

```
</RequestLimparPontoControle>
```

Descrição :

id: Identificador seqüencial para limpeza dos pontos de controle. Este valor deve ser único, nunca deve ser repetido para o limpeza dos pontos de controle, pois caso isto ocorra, ficará comprometido a limpeza dos pontos de controle;

veilD: identificador do veículo;

usuário: usuário que está limpando os pontos de controle;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparPontoControle>
  <limparPontoControle>
    <id>?</id>          [int]
    <status>?</status>  [int]
    <erro>?</erro>      [int]
  </limparPontoControle>
</ResponseLimparPontoControle>
```

Descrição :

id: Identificador da limpeza dos pontos de controle;

status: Status da limpeza:

2 – limpeza processada com sucesso (servidor processou e limpou os pontos do veículo);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – limpando (limpeza enviada, aguardando processamento do servidor (limpeza ainda não foi finalizado no equipamento));

5 – limpeza rejeitada no servidor;

Erro: [Código de erro](#), caso haja um erro no envio de uma limpeza de ponto de controle para um veículo.

 **Exemplo****Requisição:**

```
<RequestLimparPontoControle login="1234567890" senha="xxx">
```

```
<limparPontoControle>
  <id>1</id>
  <veilID>45872</veilID>
  <usuario>teste</usuario>
</limparPontoControle>
</RequestLimparPontoControle>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparPontoControle>
  <limparPontoControle>
    <id>1</id>
    <status>4</status>
    <erro>0</erro>
  </limparPontoControle>
</ResponseLimparPontoControle>
```

4.9.4 Cerca Eletrônica

Obs: A criação das Cercas Eletrônicas estão disponíveis apenas no Software New Enterprise, via integração está disponível apenas a manipulação das mesmas conforme descrito abaixo. As Cercas Eletrônicas para integração serão utilizadas, apenas as criadas no mapa da MapLink.

Cercas Eletrônicas Existentes (RequestCercaEletronica)

XML para requisição de cercas eletrônicas (no mapa da MapLink) do cliente.

```
<RequestCercaEletronica>
  <login>?</login>
  <senha>?</senha>
</RequestCercaEletronica>
```

Descrição:

Login: CGC/cpf do cliente integração ;

senha: senha do usuário do integração;

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCercaEletronica>
  <CercaEletronica>
    <cerID?</cerID>
    <nm?</nm>
    <km?</km>
    <dt?</dt>
    <dtmod?</dtmod>
  </CercaEletronica>
</ResponseCercaEletronica>
```

Descrição:

cerID: identificador da cerca eletrônica disponível, criadas no New Enterprise, no mapa da MapLink;

nm: nome da Cerca Eletrônica;

km: distância da Cerca em kilometros;

dt: data de criação da Cerca;

dtmod: data de modificação da Cerca;

 **Exemplo**

Requisição:

```
<RequestCercaEletronica>
  <login>1234567890</login>
  <senha>123456</senha>
</RequestCercaEletronica>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<ResponseCercaEletronica>
  <CercaEletronica>
    <cerID>123456</cerID>
    <nm>Cerca de Teste</nm>
    <km>14376.1201171875</km>
    <dt>2015-09-10T14:54:39.96-03:00</dt>
    <dtmod>2015-10-09T14:57:39.96-03:00</dtmod>
  </CercaEletronica>
</ResponseCercaEletronica>

```

Cerca Eletronica Embarcada (RequestCercaEletronicaEmbarcada)

XML para requisição das Cercas Eletronicas embarcadas por veiculo do cliente.

```

<RequestCercaEletronicaEmbarcada>
  <login>?</login>
  <senha>?</senha>
</RequestCercaEletronicaEmbarcada>

```

Descrição:

Login: CGC/cpf do cliente integração ;

senha: senha do usuário do integração;

Reposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseCercaEletronicaEmbarcada>
  <CercaEletronicaEmbarcada>
    <veilID>?</veilID>
    <cerID>?</cerID>
    <nm>?</nm>
    <km>?</km>
    <raio>?</raio>
    <polDIn>?</polDIn>
    <polDOut>?</polDOut>
    <ativo>?</ativo>
  </CercaEletronicaEmbarcada>
</ResponseCercaEletronicaEmbarcada>

```

Descrição:

veilD: id do veículo

cerID: identificador da cerca embarcada no veículo;

nm: nome da Cerca Eletrônica;

km: distancia da cerca em kilometros;

raio: a tolerância determinada no embarque, em metros;

polDIn: identificador do perfil operacional de entrada na Cerca Eletrônica;

polDOut: identificador do perfil operacional de saída na Cerca Eletrônica;

ativo*: indica se a Cerca Eletrônica está ativa.

*** 0 (zero) para Inativa; 1 (um) para Ativa. Apenas uma pode estar ativa, por veículo.**

 **Exemplo****Requisição:**

```
<RequestCercaEletronicaEmbarcada>
  <login>1234567890</login>
  <senha>123456</senha>
</RequestCercaEletronicaEmbarcada>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseCercaEletronicaEmbarcada>
  <CercaEletronicaEmbarcada>
    <veilD>123456</veilD>
    <cerID>123456</cerID>
    <nm>Cerca de Teste</nm>
    <km>14376.1201171875</km>
    <raio>500</raio>
    <polDIn>123</polDIn>
    <polDOut>321</polDOut>
    <ativo>1</ativo>
  </CercaEletronicaEmbarcada>
</ResponseCercaEletronicaEmbarcada>
```

■ Embarque de Cerca Eletrônica (RequestEmbarcarCercaEletronica)

XML para requisição de embarque de Cerca Eletrônica.

```
<RequestEmbarcarCercaEletronica login="?" senha="?">
  <embarcarCercaEletronica>
    <id>?</id>
    <veilD>?</veilD>
    <cerID>?</cerID>
    <polDIn>?</polDIn>
    <polDOut>?</polDOut>
    <raio>?</raio>
    <usuario>?</usuario>
  </embarcarCercaEletronica>
</RequestEmbarcarCercaEletronica>
```

Descrição:

id: Identificador seqüencial para embarque dos perfis de operação. Este valor deve ser único, nunca deve ser repetido.;

veilD: Identificador do equipamento ;

cerID: identificador da cerca embarcada no veículo;

polDIn: identificador do perfil operacional de entrada na Cerca Eletrônica;

polDOut: identificador do perfil operacional de saída na Cerca Eletrônica;

raio: raio de tolerância que deverá ser respeitado, dentro da cerca. Os limites disponíveis são: 500, 750, 1000, 1250, 1500, 1750, 2000;

usuário: usuário que está embarcando

OBS: Apenas uma Cerca Eletrônica pode estar ativa por veículo, porém pode ser embarcado mais de uma.

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ResponseEmbarcarCercaEletronica status="0">
  <id value="?" status="?">
    <veilD status="?">?</veilD>
  </id>
</ResponseEmbarcarCercaEletronica>
```

Descrição:

ResponseEmbarcarCercaEletronica status="?": [Código de erro](#), caso haja um erro na requisição.

id value: Identificador do embarque de Cerca Eletrônica;

status: Status do embarque :

- 2 – embarque processado com sucesso (servidor processou e embarcou a Cerca no veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – embarcando (embarque enviado, aguardando processamento do servidor (embarque ainda não foi finalizado no equipamento));
- 5 – embarque rejeitado no servidor;

veilD status: [Código de erro](#), caso haja um erro no envio de um embarque para um veículo.

Exemplo

Requisição:

```
<RequestEmbarcarCercaEletronica login="?" senha="?">
  <embarcarCercaEletronica>
    <id>1</id>
    <veilD>12345</veilD>
    <cerID>123456</cerID>
    <polDIn>123</polDIn>
    <polDOut>321</polDOut>
    <raio>1000</raio>
    <usuario>teste</usuario>
  </embarcarCercaEletronica>
</RequestEmbarcarCercaEletronica>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarCercaEletronica status="0">
```

```
<id value="1" status="4">
  <veiID status="12345"?></veiID>
</id>
</ResponseEmbarcarCercaEletronica>
```

Ativação e Inativação de Cerca Eletrônica

XML para ativação de Cerca Eletrônica.

```
<RequestAtivarCercaEletronica login="?" senha="?">
  <ativarCercaEletronica>
    <id?></id>
    <veiID?></veiID>
    <cerID?></cerID>
    <usuario?></usuario>
  </ativarCercaEletronica>
</RequestAtivarCercaEletronica>
```

Descrição:

id: Identificador seqüencial para ativação de Cerca Eletrônica. Este valor deve ser único, nunca deve ser repetido para ativação de Cerca Eletrônica, pois caso isto ocorra, ficará comprometido a ativação de Cerca Eletrônica;

veiID: identificador do veículo;

cerID: id de Cerca já embarcada, para ser ativada. Para recuperar cercas embarcadas, utilize a RequestCercaEletronica Embarcada;

usuário: usuário que está realizando a ativação;

Exemplo

Requisição:

```
<RequestAtivarCercaEletronica login="?" senha="?">
  <ativarCercaEletronica>
    <id>1</id>
    <veiID>12345</veiID>
    <cerID>123456</cerID>
    <usuario>teste</usuario>
  </ativarCercaEletronica>
```

```
</RequestAtivarCercaEletronica>
```

XML para inativação de Cerca Eletrônica.

```
<RequestInativarCercaEletronica login="?" senha="?">
  <inativarCercaEletronica>
    <id?</id>
    <veilD?</veilD>
    <cerID?</cerID>
    <usuario?</usuario>
  </inativarCercaEletronica>
</RequestInativarCercaEletronica>
```

Descrição:

id: Identificador seqüencial para ativação de Cerca Eletrônica. Este valor deve ser único, nunca deve ser repetido para ativação de Cerca Eletrônica, pois caso isto ocorra, ficará comprometido a ativação de Cerca Eletrônica;

veilD: identificador do veículo;

cerID: id de Cerca já embarcada, para ser ativada. Para recuperar cercas embarcadas, utilize a RequestCercaEletronica Embarcada;

usuário: usuário que está realizando a ativação;

Exemplo

Requisição:

```
<RequestInativarCercaEletronica login="?" senha="?">
  <inativarCercaEletronica>
    <id>1</id>
    <veilD>12345</veilD>
    <cerID>123456</cerID>
    <usuario>teste</usuario>
  </inativarCercaEletronica>
</RequestInativarCercaEletronica>
```

4.9.5 Outras Requisições

Limpeza da Inteligência Embarcada (RequestLimparIE)

XML para limpeza de toda a inteligência embarcada de um veículo (macros, ponto de controle, cerca, etc).

```
<RequestLimparIE login="?" senha="?">
  <limparIE>
    <id?></id>           [int]
    <veilD?></veilD>     [int]
    <usuario?></usuario> [varchar(20)]
  </limparIE>
</RequestLimparIE>
```

Descrição:

id: Identificador seqüencial para exclusão da I. E. . Este valor deve ser único, nunca deve ser repetido.

veilD: Identificador do veículo;

usuario: Usuário que está limpando;

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparIE>
  <limparIE>
    <id?></id>
    <status?></status>
    <erro?></erro>
  </limparIE>
</ResponseLimparIE>
```

Descrição:

id: Identificador da limpeza da I.E.;

status:Status da limpeza:

2 – limpeza processada com sucesso (servidor processou e limpou a i.e. veículo);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – limpando (limpeza enviada, aguardando processamento do servidor (limpeza ainda não foi finalizado no equipamento));

5 – limpeza rejeitada no servidor;

Erro: [Código de erro](#), caso haja um erro no envio de uma limpeza de ponto da i.e. para um veículo.

Exemplo

Requisição:

```
<RequestLimparE login="12345678910" senha="xxx">
  <limparE>
    <id>1</id>
    <veilD>49752</veilD>
    <usuario>teste</usuario>
  </limparE>
</RequestLimparE>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparE>
  <limparE>
    <id>1</id>
    <status>4</status>
    <erro>0</erro>
  </limparE>
</ResponseLimparE>
```

Senha e Contra-Senha (RequestSenhaContraSenha)

XML para requisitar a contra senha através da senha informada pelo motorista

```
<RequestSenhaContraSenha login="12345678910" senha="xxx">
  <senhaContraSenha>
    <id>?</id> [int]
    <veild>?</veild> [int]
```

```
<senha?</senha>
<motivo?</motivo>
<usuario?</usuario> [varchar(20)]
</senhaContraSenha>
</RequestSenhaContraSenha>
```

Descrição:

Id: Identificador seqüencial para requisitar a contra senha. Este valor deve ser único, nunca deve ser repetido para requisição de contra senha, pois caso isto ocorra, ficará comprometida a requisição da contra senha;

veiid: Identificador do veículo;

senha: Senha informada pelo motorista;

motivo: Motivo da solicitação da contra senha;

usuário: Usuário que está requisitando a contra senha

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseSenhaContraSenha>
  <senhaContraSenha>
    <id?</id>
    <contraSenha?</contraSenha>
    <lat?</lat>
    <lon?</lon>
    <erro?</erro>
  </senhaContraSenha>
</ResponseSenhaContraSenha>
```

Descrição:

id: Identificador da requisição;

contraSenha: Contra senha gerada;

latitude: Latitude da posição do veículo;

longitude: Longitude da posição do veículo;

erro: [Código de erro](#), caso haja um erro na solicitação da contra senha.

Exemplo

Requisição:

```
<RequestSenhaContraSenha login="12345678910" senha="xxxx">
  <senhaContraSenha>
    <id>1</id>
    <veilD>45872</veilD>
    <senha>6AC4E54B8D30A</senha>
    <motivo>teste web service</motivo>
    <usuario>teste</usuario>
  </senhaContraSenha>
</RequestSenhaContraSenha>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseSenhaContraSenha>
  <senhaContraSenha>
    <id>1</id>
    <contraSenha>3532</contraSenha>
    <lat>-23.302</lat>
    <lon>-51.1775</lon>
    <erro>0</erro>
  </senhaContraSenha>
</ResponseSenhaContraSenha>
```

4.10 Requisição de Telemetria

Nota

Para realizar o embarque de Telemetria v2.5, será necessário primeiramente criar um projeto de telemetria no New Enterprise (caso sua operação não saiba como fazê-lo, contactar a Consultoria Técnica). Após isto, você poderá obter os projetos de Telemetria criados, através da RequestTelemetriaProjeto. Com este resultado, você utilizará o tlpID desejado, para proceder com o Embarque através da RequestEmbarcarTelemetria.

☐ **Projetos de Telemetria (RequestTelemetriaProjetos)**

XML para requisição dos Projetos de Telemetria.

```
<RequestTelemetriaProjetos>
  <login>?</login>
  <senha>?</senha>
</RequestTelemetriaProjetos>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseTelemetriaProjetos>
  <TelemetriaProjetos>
    <tlpID>?</tlpID>
    <nome>?</nome>
    <dtCriacao>?</dtCriacao>
    <dtModif>?</dtModif>
  </TelemetriaProjetos>
  <TelemetriaProjetos>
    <tlpID>?</tlpID>
    <nome>?</nome>
    <descricao>?</descricao>
    <dtCriacao>?</dtCriacao>
    <dtModif>?</dtModif>
  </TelemetriaProjetos>
</ResponseTelemetriaProjetos>
```

Descrição:

tlpID: numero de identificação do Projeto de Telemetria;

nome: nome do Projeto de Telemetria;

descricao: breve descrição do Projeto de Telemetria;

dtCriacao: data/hora da criação do Projeto de Telemetria;

dtModif: data/hora de modificação do Projeto de Telemetria;

Exemplo

Requisição:

```
<RequestTelemetriaProjetos>
  <login>1234567890</login>
  <senha>123456</senha>
</RequestTelemetriaProjetos>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseTelemetriaProjetos>
  <TelemetriaProjetos>
    <tlpID>16</tlpID>
    <nome>Teste Projeto</nome>
    <dtCriacao>2015-10-18T10:00:00.953-02:00</dtCriacao>
    <dtModif>2015-10-18T10:00:00.953-02:00</dtModif>
  </TelemetriaProjetos>
  <TelemetriaProjetos>
    <tlpID>17</tlpID>
    <nome>Teste Projeto 1</nome>
    <descricao>Projeto criado como teste para integração</descricao>
    <dtCriacao>2015-10-18T10:30:00.655-02:00</dtCriacao>
    <dtModif>2015-10-18T13:00:15.057-02:00</dtModif>
  </TelemetriaProjetos>
</ResponseTelemetriaProjetos>
```

Embarque de Telemetria (RequestEmbarcarTelemetria)

XML para requisição de embarque de Telemetria.

```
<RequestEmbarcarTelemetria login="?" senha="?">
  <embarcarTelemetria>
    <id>?</id>
    <veilID>?</veilID>
    <tlpID>?</tlpID>
    <usuario>?</usuario>
  </embarcarTelemetria>
</RequestEmbarcarTelemetria>
```

Descrição:

id: Identificador seqüencial para embarque do Projeto de Telemetria. Este valor deve ser único, nunca deve ser repetido.;

veilD: Identificador do equipamento ;

tlpID: identificador do Projeto de Telemetria a ser embarcado;

usuário: usuário que está embarcando

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarTelemetria status="0">
  <id value="" status="">
    <veilD status="">?</veilD>
  </id>
</ResponseEmbarcarTelemetria>
```

Descrição:

ResponseEmbarcarTelemetria status="?": [Código de erro](#), caso haja um erro na requisição.

id value: Identificador do embarque de Projeto de Telemetria;

status: Status do embarque :

2 – embarque processado com sucesso (servidor processou e embarcou o Projeto de Telemetria);

3 – rejeitado (erro de sintaxe ou falta de credenciais);

4 – embarcando (embarque enviado, aguardando processamento do servidor (embarque ainda não foi finalizado no equipamento));

5 – embarque rejeitado no servidor;

veilD status: [Código de erro](#), caso haja um erro no envio de um embarque para um veículo.

 **Exemplo**

Requisição:

```
<RequestEmbarcarTelemetria login="?" senha="?">
  <embarcarTelemetria>
    <id?</id>
    <veilD?</veilD>
    <tlpID?</tlpID>
    <usuario?</usuario>
  </embarcarTelemetria>
</RequestEmbarcarTelemetria>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarTelemetria status="0">
  <id value="1" status="4">
    <veilD status="0">45872</veilD>
  </id>
</ResponseEmbarcarTelemetria>
```

Importante ressaltar que a configuração do relatório de Telemetria deverá ser feita exclusivamente no Software New Enterprise.

Telemetria Item (RequestTelemetrialtem)

XML para requisição dos itens de Telemetria.

Esta requisição envia os itens de Telemetria, que serão utilizados para representar cada item no Relatório de Telemetria.

```
<RequestTelemetrialtem>
  <login?</login>
  <senha?</senha>
</RequestTelemetrialtem>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseTelemetrialtem>
  <Telemetrialtem>
```

```

        <tiID?></tiID>          [int]
        <tiDesc?></tiDesc>      [varchar(50)]
    </Telemetrialtem>
    <Telemetrialtem>
        <tiID?></tiID>          [int]
        <tiDesc?></tiDesc>      [varchar(50)]
    </Telemetrialtem>
    <Telemetrialtem>
        <tiID?></tiID>          [int]
        <tiDesc?></tiDesc>      [varchar(50)]
    </Telemetrialtem>
</ResponseTelemetrialtem>

```

Descrição:

tiID: numero de identificação do item da Telemetria

tiDesc: descrição do item de Telemetria

Exemplo

Requisição:

```

<RequestTelemetrialtem>
  <login>05899105909</login>
  <senha>123456</senha>
</RequestTelemetrialtem>

```

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseTelemetrialtem>
  <Telemetrialtem>
    <tiID>10</tiID>
    <tiDesc>Excesso de Velocidade</tiDesc>
  </Telemetrialtem>
  <Telemetrialtem>
    <tiID>14</tiID>
    <tiDesc>Tempo Máximo de Veículo Parado com Motor Ligado</tiDesc>
  </Telemetrialtem>
  <Telemetrialtem>
    <tiID>15</tiID>
    <tiDesc>Excesso de Temperatura do Líquido de Arrefecimento</
tiDesc>

```

```

</Telemetrialtem>
<Telemetrialtem>
  <tiID>16</tiID>
  <tiDesc>Frenagem Brusca</tiDesc>
</Telemetrialtem>
<Telemetrialtem>
  <tiID>17</tiID>
  <tiDesc>Aceleração Brusca</tiDesc>
</Telemetrialtem>
<Telemetrialtem>
  <tiID>18</tiID>
  <tiDesc>Banguela</tiDesc>
</Telemetrialtem>
<Telemetrialtem>
  <tiID>19</tiID>
  <tiDesc>Excesso de RPM</tiDesc>
</Telemetrialtem>
<Telemetrialtem>
  <tiID>20</tiID>
  <tiDesc>Excesso de Velocidade na Chuva</tiDesc>
</Telemetrialtem>
</ResponseTelemetrialtem>

```

Telemetria Relatório (RequestTelemetria)

XML para requisição dos relatórios de Telemetria.

Esta requisição envia o relatório de ocorrências referente ao que foi previamente cadastrado na Telemetria do equipamento via software Enterprise.

```

<RequestTelemetria>
  <login>?</login>
  <senha>?</senha>
  <tID>?</tID>
</RequestTelemetria>

```

Descrição:

tID: Id do último relatório recebido de Telemetria, os próximos relatórios a serem recebidos virão a partir deste id; Neste campo sempre deve ser informado o tID do último relatório recebido

Resposta:

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseTelemetria>
  <Telemetria tID=?"> [bigint]
    <veilD?</veilD> [int]
    <odm?</odm> [int]
    <hfc?</hfc> [varchar(8)]
    <hinc?</hinc> [datetime]
    <item tilD=?"> [int]
      <qt?</qt> [int]
      <tt?</tt> [int]
      <hi?</hi> [varchar(8)]
      <hf?</hf> [varchar(8)]
    </item>
    <item tilD=?"> [int]
      <qt?</qt> [int]
      <tt?</tt> [int]
      <hi?</hi> [varchar(8)]
      <hf?</hf> [varchar(8)]
    </item>
  </Telemetria>
  <Telemetria tID=?"> [bigint]
    <veilD?</veilD> [int]
    <odm?</odm> [int]
    <hfc?</hfc> [varchar(8)]
    <hinc?</hinc> [datetime]
    <item tilD=?"> [int]
      <qt?</qt> [int]
      <tt?</tt> [int]
      <hi?</hi> [varchar(8)]
      <hf?</hf> [varchar(8)]
    </item>
    <item tilD=?"> [int]
      <qt?</qt> [int]
      <tt?</tt> [int]
      <hi?</hi> [varchar(8)]
      <hf?</hf> [varchar(8)]
    </item>
  </Telemetria>
</ResponseTelemetria>

```

Descrição:

Telemetria tID: ID do relatório de Telemetria

veilD: Id do veículo

odm: Hodometro do veículo no relatório

hfc: Horário do fechamento do relatório

hinc: Horário da inclusão do relatório

item tID: número de identificação do item de Telemetria presente no relatório

qt: quantidade de violações

tt: tempo total das violações (em segundos)

hi: hora inicial

hf: hora final

Exemplo

Requisição:

```
<RequestTelemetria>
  <login>05899105909</login>
  <senha>123456</senha>
  <tID>1000</tID>
</RequestTelemetria>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseTelemetria>
  <Telemetria tID="1734">
    <veilD>35379</veilD>
    <odm>0</odm>
    <hfc>01:00:00</hfc>
    <hinc>14/01/2011 01:08:04</hinc>
    <item tID="10">
      <qt>0</qt>
      <tt>0</tt>
      <hi>06:00:00</hi>
      <hf>18:00:00</hf>
    </item>
```

```
<item tID="19">
  <qt>0</qt>
  <tt>0</tt>
  <hi>06:00:00</hi>
  <hf>18:00:00</hf>
</item>
</Telemetria>
<Telemetria tID="1735">
  <veilD>85279</veilD>
  <odm>0</odm>
  <hfc>01:00:00</hfc>
  <hinc>14/01/2011 01:58:45</hinc>
  <item tID="15">
    <qt>2</qt>
    <tt>42380</tt>
    <hi>06:00:00</hi>
    <hf>20:00:00</hf>
  </item>
  <item tID="19">
    <qt>1</qt>
    <tt>22</tt>
    <hi>06:00:00</hi>
    <hf>20:00:00</hf>
  </item>
</Telemetria>
<Telemetria tID="1736">
  <veilD>85353</veilD>
  <odm>0</odm>
  <hfc>01:00:00</hfc>
  <hinc>14/01/2011 03:43:01</hinc>
  <item tID="19">
    <qt>7</qt>
    <tt>16</tt>
    <hi>06:00:00</hi>
    <hf>20:00:00</hf>
  </item>
  <item tID="15">
    <qt>2</qt>
    <tt>49228</tt>
    <hi>06:00:00</hi>
    <hf>20:00:00</hf>
  </item>
</Telemetria>
<Telemetria tID="1737">
  <veilD>21789</veilD>
  <odm>0</odm>
  <hfc>01:00:00</hfc>
```

```
<hinc>14/01/2011 07:14:03</hinc>
<item tilD="10">
  <qt>0</qt>
  <tt>0</tt>
  <hi>00:00:00</hi>
  <hf>23:59:00</hf>
</item>
<item tilD="14">
  <qt>4</qt>
  <tt>9620</tt>
  <hi>00:00:00</hi>
  <hf>23:59:00</hf>
</item>
<item tilD="16">
  <qt>0</qt>
  <tt>0</tt>
  <hi>00:00:00</hi>
  <hf>23:59:00</hf>
</item>
<item tilD="17">
  <qt>0</qt>
  <tt>0</tt>
  <hi>00:00:00</hi>
  <hf>23:39:00</hf>
</item>
</Telemetria>
</ResponseTelemetria>
```

4.11 Requisição do Horário do Servidor

Horário Servidor (RequestDataHoraServidor)

XML para requisição da data/hora do servidor de integração.

```
<RequestDataHoraServidor>
  <login>?</login>
  <senha>?</senha>
</RequestDataHoraServidor>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseDataHoraServidor>
  <DataHoraServidor>
    <dataHora>?</dataHora>
  </DataHoraServidor>
</ResponseDataHoraServidor>
```

Descrição:

dataHora: Data e hora do servidor de integração.

 **Exemplo**

Requisição:

```
<RequestDataHoraServidor>
  <login>1234567890</login>
  <senha>1234</senha>
</RequestDataHoraServidor>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseDataHoraServidor>
  <DataHoraServidor>
    <dataHora>2012-06-01T14:49:08.45-03:00</dataHora>
  </DataHoraServidor>
</ResponseDataHoraServidor>
```

4.12 Requisição de Motorista

 **Motoristas (RequestMotorista)**

XML para requisição da lista dos Motoristas cadastrados.

```
<RequestMotorista>
  <login>?</login>
  <senha>?</senha>
</RequestMotorista>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMotorista>
  <Motorista>
    <motID>?</motID>
    <mot>?</mot>
    <rg>?</rg>
    <cpf>?</cpf>
  </Motorista>
</ResponseMotorista>
```

Descrição:

motID: Id do Motorista;

mot: Nome do Motorista;

rg: Número do documento de RG do cadastro do Motorista;

cpf: Número do documento de CPF do cadastro do Motorista;

 **Exemplo****Requisição:**

```
<RequestAcessorio>
  <login>12345678910</login>
  <senha>xxxx</senha>
</RequestAcessorio>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMotorista>
  <Motorista>
    <motID>4</motID>
    <mot>TESTE</mot>
    <rg>49660839</rg>
    <cpf>2222222222</cpf>
  </Motorista>
</ResponseMotorista>
```

Embarque (RequestEmbarcarMotorista)

XML para requisição de embarque de motoristas

```
<RequestEmbarcarMotorista login="?" senha="?">
  <embarcarMotorista>
    <id>?</id>
    <veilID>?</veilID>
    <motID>?</motID>
  </embarcarMotorista>
  <usuario>?</usuario>
</RequestEmbarcarMotorista>
```

Descrição:

id: é um identificador seqüencial para o envio de comandos aos veículos, este valor nunca deve ser repetido, pois caso isto ocorra, ficará comprometido o status do comando;

veilID: identificação do veículo;

motID: identificação do motorista;

usuario: usuário que está realizando o embarque.

***Poderá ser embarcado até 30 motoristas por vez, lembrando que o equipamento aceita até 50 motoristas embarcados. Para enviar mais de um motorista, verifique o [Exemplo](#).**

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarMotorista>
  <embarcarMotorista>
    <id>?</id>
    <status>?</status>
    <erro>?</erro>
  </embarcarMotorista>
</ResponseEmbarcarMotorista>
```

Descrição:

id: Identificador seqüencial do envio do embarque;

status: Status do envio :

- 2 – embarque processado com sucesso (servidor processou e enviou o embarque ao veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – enviado (comando de embarque enviado, aguardando processamento do servidor (embarque ainda não foi enviado para o equipamento));
- 5 – embarque rejeitado no servidor;

erro: [Código de erro](#), caso haja um erro no envio do embarque.

Exemplo

Requisição:

```
<RequestEmbarcarMotorista login="0123456789" senha="123456">
  <embarcarMotorista>
    <id>1</id>
    <veilD>12345</veilD>
    <motID>10</motID>
  </embarcarMotorista>
  <embarcarMotorista>
    <id>2</id>
    <veilD>12346</veilD>
    <motID>20</motID>
  </embarcarMotorista>
  <embarcarMotorista>
    <id>3</id>
    <veilD>12347</veilD>
    <motID>30</motID>
  </embarcarMotorista>
  <usuario>teste</usuario>
</RequestEmbarcarMotorista>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseEmbarcarMotorista>
  <embarcarMotorista>
    <id>1</id>
    <status>4</status>
    <erro>0</erro>
  </embarcarMotorista>
  <embarcarMotorista>
    <id>2</id>
    <status>3</status>
    <erro>2</erro>
```

```
</embarcarMotorista>
<embarcarMotorista>
  <id>3</id>
  <status>4</status>
  <erro>0</erro>
</embarcarMotorista>
</ResponseEmbarcarMotorista>
```

Limpeza (RequestLimparMotorista)

XML para realizar a limpeza dos Motoristas embarcados no equipamento.

```
<RequestLimparMotorista login="?" senha="?">
  <limparMotorista>
    <id>?</id>
    <veilD>?</veilD>
  </limparMotorista>
  <usuario>?</usuario>
</RequestLimparMotorista>
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparMotorista>
  <limparMotorista>
    <id>?</id>
    <status>?</status>
    <erro>?</erro>
  </limparMotorista>
</ResponseLimparMotorista>
```

Descrição:

id: Identificador seqüencial do envio da limpeza;

status: Status do envio :

- 2 – limpeza processada com sucesso (servidor processou e enviou a limpeza ao veículo);
- 3 – rejeitado (erro de sintaxe ou falta de credenciais);
- 4 – enviado (comando de limpeza enviada, aguardando processamento do servidor (limpeza ainda não foi enviado para o equipamento));
- 5 – limpeza rejeitada no servidor;

erro: [Código de erro](#), caso haja um erro no envio da limpeza.

Exemplo

Requisição:

```
<RequestLimparMotorista login="0123456789" senha="123456">
  <limparMotorista>
    <id>1</id>
    <veilD>12345</veilD>
  </limparMotorista>
  <usuario>teste</usuario>
</RequestLimparMotorista>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseLimparMotorista>
  <limparMotorista>
    <id>1</id>
    <status>4</status>
    <erro>0</erro>
  </limparMotorista>
</ResponseLimparMotorista>
```

Embarcado (RequestMotoristaEmbarcado)

XML para requisição dos motoristas já embarcados em cada equipamento.

```
<RequestMotoristaEmbarcado>
  <login>?</login>
  <senha>?</senha>
</RequestMotoristaEmbarcado>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMotoristaEmbarcado>
```

```
<MotoristaEmbarcado>
  <veilD>?</veilD>
  <motID>?</motID>
</MotoristaEmbarcado>
</ResponseMotoristaEmbarcado>
```

Descrição:

veilD: identificação do veículo;

mtID: identificação do motorista;

 **Exemplo**

Requisição:

```
<RequestMotoristaEmbarcado>
  <login>0123456789</login>
  <senha>123456</senha>
</RequestMotoristaEmbarcado>
```

Reposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMotoristaEmbarcado>
  <MotoristaEmbarcado>
    <veilD>12345</veilD>
    <motID>10</motID>
  </MotoristaEmbarcado>
  <MotoristaEmbarcado>
    <veilD>12346</veilD>
    <motID>20</motID>
  </MotoristaEmbarcado>
</ResponseMotoristaEmbarcado>
```

Parte



5 Erros para Envio Incorreto do XML

Caso haja alguma inconsistência na envio de um XML, será retornada a seguinte resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
  <ErrorRequest>
    <codigo>14</codigo>
    <erro>Nenhum comando/macro a ser enviada, favor verificar as tags xml.</erro>
  </request>EnvioComando</request>
</ErrorRequest>
```

Código	Erro
1	Atributos para leitura de requisição inválidos. (Atributos, login e/ou senha incorretos)
2	Falha ao realizar solicitação / Falta de parâmetros
5	Usuário não credenciado para envio de comando / macro
7	Não atingiu o tempo mínimo para reenvio da requisição
8	Cliente não habilitado para referencia entrega
12	Falha ao realizar a formatação dos itens de macro
13	Falha ao processar comandos SPY
14	Nenhum comando / macro a ser enviado, favor verificar as tags xml

Ao enviar um XML inválido, será retornada a seguinte resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
  <ErrorRequest>
    <erro>Falha ao realizar o parse do xml.</erro>
  </ErrorRequest>
```

Ou

```
<?xml version="1.0" encoding="UTF-8"?>
  <ErrorRequest>
    <Erro>Tag xml invalida.</Erro>
  </ErrorRequest>
```

Parte



6 Relacionamentos

Indicativo de Requisições relacionadas.

Acessórios, Acessórios Veículos

O campo ACVID recebido na requisição de Acessórios Veículos é idêntico ao campo ACVID da requisição de Acessórios, identificando quais acessórios o veículo possui instalado.

Macro Grupo, Macro Item, Grupo Macro Embarcado, Embarque de Grupo

O campo TGRID, do macro grupo, identifica o código do grupo de macro. Este mesmo campo também é encontrado na requisição Macro Item, é o campo TGRID, identificando o grupo de macro do item em questão.

Na Macro Embarcado, os campos ATIVOCV e ATIVOVC equivalem ao campo TGRID do macro grupo, identificando qual grupo de macro está ativo para o veículo.

O campo TGRID também é o mesmo utilizado na hora do embarque de grupos de macro nos equipamentos.

Macro Item, Mensagens Veículos (RequestMensagemCB)

O campo TFRID recebido na requisição de Mensagens Veículos equivale ao campo TFRID da requisição de Macro Item.

Motoristas

O campo motID retornado na RequestMensagemCB faz menção ao ID do motorista que é recuperado através da RequestMotorista.

Veículos

Na requisição das informações dos veículos, recebe-se o código do equipamento (veilD), a placa do veículo (placa), o tipo de equipamento (eqp) e a versão do equipamento (versão). O campo veilD é a identificação do veículo, todas as mensagens são recebidas e enviadas com este código.

Ponto de Controle (Embarque), Perfil Operacional

Nas requisições para embarque de Pontos de Controle, é necessário que informe o polD (in e out), este polD é o mesmo encontrado nas requisições de Perfil Operacional. São

necessários para informar qual perfil o equipamento assumirá ao entrar e sair do Ponto de Controle.

SPY's

Na requisição das informações dos Spy's, recebe-se o código do equipamento (spyID), o número de serie do equipamento (serie), o tipo de equipamento (eqp), etc.

O campo spyID é a identificação do equipamento, todas as mensagens são recebidas e enviadas com este código.

Parte



7 Exemplo em C#



OBS: Possuimos alguns exemplos em outras linguagens, caso seja necessário entre em contato conosco.

- [Requisição](#)
- [Descompactação](#)

7.1 Requisição

Requisição em C#

```
private static HttpWebRequest CreateRequest()
{
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create("http://webservice.
onixsat.com.br");
    request.Method = "POST";
    request.ContentType = "text/xml";
    return request;
}

public static string RequestXml(string strRequest)
{
    string result = string.Empty;

    try
    {
        // requisição xml em bytes
```

```
byte[] sendData = UTF8Encoding.UTF8.GetBytes(strRequest);

// cria requisicao
HttpRequest request = CreateRequest();
Stream requestStream = request.GetRequestStream();

// envia requisição
requestStream.Write(sendData, 0, sendData.Length);
requestStream.Flush();
requestStream.Dispose();

// captura resposta
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
Stream responseStream = response.GetResponseStream();

MemoryStream output = new MemoryStream();

byte[] buffer = new byte[256];
int byteReceived = -1;

do
{
    byteReceived = responseStream.Read(buffer, 0, buffer.Length);
    output.Write(buffer, 0, byteReceived);
} while (byteReceived > 0);

responseStream.Dispose();
response.Close();

buffer = output.ToArray();
output.Dispose();

// transforma resposta em string para leitura xml
result = UTF8Encoding.UTF8.GetString(buffer);
}
catch (Exception ex)
{
    // tratar exceção
}
return result;
}
```

7.2 Descompactação

Zip

```
public override byte[] Decompress(byte[] data)
{
    if (IsValidDecompress(data))
    {
        try
        {
            // nome arquivo
            string nameDirectory = Guid.NewGuid().ToString();

            // cria diretorio na pasta temp
            DirectoryInfo dInfo = Directory.CreateDirectory(Path.GetTempPath() +
nameDirectory);

            // path arquivo zip
            string pathFileZip = dInfo.FullName + Path.DirectorySeparatorChar +
nameDirectory + ".zip";

            // CREATE FILE ZIP
            File.WriteAllBytes(pathFileZip, data);

            // DECOMPRESS
            FastZip fastZip = new FastZip();
            fastZip.ExtractZip(pathFileZip, dInfo.FullName, "");

            // nome do arquivo txt salvo
            string nameFileTxt = Directory.GetFiles(dInfo.FullName, "*.txt")[0];

            // arquivo descompactado em bytes
            byte[] buffer = File.ReadAllBytes(nameFileTxt);

            // exclui diretorio e arquivos da pasta temp
            if (Directory.Exists(dInfo.FullName))
                Directory.Delete(dInfo.FullName, true);

            return buffer;
        }
        catch (Exception ex)
        {
            throw new ZipLibraryException("Falha ao descompactar arquivo no formato .zip",
ex);
        }
    }
}
```

```
        return null;
    }
```

gZip

```
public override byte[] Decompress(byte[] data)
{
    if (IsValidDecompress(data))
    {
        try
        {
            MemoryStream input = new MemoryStream();
            input.Write(data, 0, data.Length);
            input.Position = 0;

            GZipStream gzip = new GZipStream(input, CompressionMode.Decompress,
true);

            byte[] buff = new byte[256];

            MemoryStream output = new MemoryStream();

            int read = gzip.Read(buff, 0, buff.Length);

            while (read > 0)
            {
                output.Write(buff, 0, read);
                read = gzip.Read(buff, 0, buff.Length);
            }

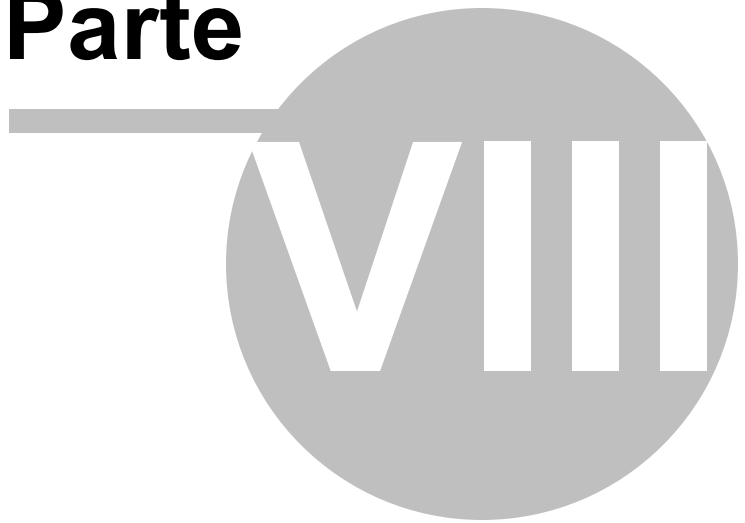
            gzip.Close();

            byte[] buffer = output.ToArray();

            output.Dispose();

            return buffer;
        }
        catch (Exception ex)
        {
            throw new ZipLibraryException("Falha ao descompactar arquivo no formato .
gzip", ex);
        }
    }
    return null;
}
```

Parte



8 Lista de Comandos para SPY

SpyTrack (eqp = 7)

Codigo	Descricao	Detalhe
1	Última Posição	Atualiza, após aproximadamente 3 minutos, a posição atual do equipamento no mapa. Obtém informações de velocidade.
2	Desligar SpyTrack	Desliga o equipamento.
3	Perfil 60/60 minutos	Frequência do temporizador do equipamento de 60 em 60 minutos em movimento. Previsão média de bateria de 4 a 10 dias.
4	Perfil 30/30 minutos	Frequência do temporizador do equipamento de 30 em 30 minutos em movimento. Previsão média de bateria de 4 a 10 dias.
5	Perfil 10/10 minutos	Frequência do temporizador do equipamento de 10 em 10 minutos em movimento. Previsão média de bateria de 2 a 5 dias.
6	Perfil 5/5 minutos	Frequência do temporizador do equipamento de 5 em 5 minutos em movimento. Previsão média de bateria de 1 a 3 dias.
7	Perfil 3/3 minutos	Frequência do temporizador do equipamento de 3 em 3 minutos em movimento. Previsão média de bateria de 2 dias.
9	Configuração de Velocidade	Configuração de velocidade máxima excedida. (Este comando está desabilitado)
10	Solicitar nível de bateria	Atualiza, após aproximadamente 3 minutos, o nível de bateria do equipamento.

SpyTrack2 (eqp = 14)

Codigo	Descricao	Detalhe
1	Última Posição	Atualiza, após aproximadamente 3 minutos, a posição atual do equipamento no mapa. Obtém informações de velocidade.
2	Desligar SpyTrack2	Desliga o equipamento.
3	Perfil 60/60 minutos	Frequência do temporizador do equipamento de 60 em 60 minutos em movimento. Previsão média de bateria de 4 a 10 dias.

4	Perfil 30/30 minutos	Frequência do temporizador do equipamento de 30 em 30 minutos em movimento. Previsão média de bateria de 4 a 10 dias.
5	Perfil 10/10 minutos	Frequência do temporizador do equipamento de 10 em 10 minutos em movimento. Previsão média de bateria de 2 a 5 dias.
6	Perfil 5/5 minutos	Frequência do temporizador do equipamento de 5 em 5 minutos em movimento. Previsão média de bateria de 1 a 3 dias.
7	Perfil 3/3 minutos	Frequência do temporizador do equipamento de 3 em 3 minutos em movimento. Previsão média de bateria de 2 dias.
9	Configuração de Velocidade	Configuração de velocidade máxima excedida. (Este comando está desabilitado)
10	Solicitar Nível de Bateria	Atualiza o nível de bateria do dispositivo
16	Ativar Modo Escuta	Permite que o cliente ative o modo escuta do dispositivo.
17	Desativar Modo Pânico	Desativa o modo pânico do dispositivo.

Parte



9 Lista de Comandos por Equipamento

☐ **Satelite System e Hybrid System**

Logo abaixo estão relacionados os comandos, versões de computadores de bordo e tipos de equipamentos que o sistema de integração suporta o envio.

Verifique a compatibilidade para cada veículo antes de enviar um comando.

Comandos suportados por tipos de equipamento **Satelite System (eqp = 1 / Puro Satelital)** e **Hybrid System (eqp = 2 / Híbrido)** com Computador de Bordo, nas versões:

Satélite: 1.01, 1.02, 1.03, 1.04, 2.00, 2.01, 2.02, 2.03, 2.04

Híbrido: 3.00, 3.01, 3.02, 3.03, 3.04, 3.05, 3.06, 3.07, 3.08, 3.09, 3.10, 3.11, 3.12, 3.13, 3.14, ou superior

DESCRICAO	COMANDO
Ligar Aviso de cabine	14
Desligar Aviso de cabine	01046000000000
Status Padrão	02
Ligar Bloqueio	10
Desligar Bloqueio	11
Ligar Sirene	12
Desligar Sirene	13
Última Posição	9C

Comandos suportados por tipos de equipamento **Satelite System (eqp = 1 / Puro Satelital)** e **Hybrid System (eqp = 2 / Híbrido)** com Computador de Bordo, nas versões:

Satélite: 2.00, 2.01, 2.02, 2.03, 2.04

Híbrido: 3.00, 3.01, 3.02, 3.03, 3.04, 3.05, 3.06, 3.07, 3.08, 3.09, 3.10, 3.11, 3.12, 3.13, 3.14, ou superior

DESCRICAO	COMANDO
Ligar Pisca Alerta	1A
Desligar Pisca Alerta	1B
Sensor de Temperatura	1C
Habilitar Botão de Abertura de Trava de Baú	1D
Desabilitar Botão de Abertura de Trava de Baú	1E
Status Extendido	1F
Habilitar sensor de porta do motorista	01048C80000060

Desabilitar sensor de porta do motorista	01048C00000000
Habilitar sensor de porta do carona	01048680000060
Desabilitar sensor de porta do carona	01048600000000
Habilitar sensor de desengate	0104508000000A
Desabilitar sensor de desengate	01045000000000

Comandos suportados por tipos de equipamento **Satelite System (eqp = 1 / Puro Satelital)** e **Hybrid System (eqp = 2 / Híbrido)** com Computador de Bordo, nas versões:

Satélite: 2.01, 2.02, 2.03, 2.04

Híbrido: 3.00, 3.01, 3.02, 3.03, 3.04, 3.05, 3.06, 3.07, 3.08, 3.09, 3.10, 3.11, 3.12, 3.13, 3.14, ou superior

DESCRICAO	COMANDO
Habilitar Botão da Trava da 5ª Roda	22
Desabilitar Botão da Trava da 5ª Roda	23

Comandos suportados por tipos de equipamento **Satelite System (eqp = 1 / Puro Satelital)** e **Hybrid System (eqp = 2 / Híbrido)** com Computador de Bordo, nas versões:

Satélite: 2.03, 2.04

Híbrido: 3.04, 3.05, 3.06, 3.07, 3.08, 3.09, 3.10, 3.11, 3.12, 3.13, 3.14, ou superior

DESCRICAO	COMANDO
Sensor de Temperatura 2	26
Sendor de Temperatura 3	27

Comandos suportados por tipos de equipamento Híbrido (eqp = 2) com Computador de Bordo, nas versões:

Híbrido: 3.00, 3.01, 3.02, 3.03, 3.04, 3.05, 3.06, 3.07, 3.08, 3.09, 3.10, 3.11, 3.12, 3.13, 3.14, ou superior

DESCRICAO	COMANDO
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3

Desabilita a geração de violação do sensor de temperatura, o sensor continuará informando o valor da temperatura, porém o sistema não identificará se houve violação. Para habilitar novamente o sensor, basta enviar um comando de configuração, descrito no Topico de Configuração de Sensor de Temperatura

Comandos de Alteração de Temporizador Satelital

Comandos de Temporizador são aceitos apenas por equipamentos dos tipos Híbridos e Satélite (eqp 1 e 2).

Segue relação de comandos de alteração de Temporizador aceitos pelo Sistema de Integração.

OBS: As possibilidades aceitas de retorno para a configuração padrão são: 1, 2, 4, 6, 12, 24, 48 horas. Exemplo: TT06-1

DESCRICAÇÃO	COMANDO
Temporizador 06-06 Retornar para configuração padrão após ? hora	TT06-?
Temporizador 10-10 Retornar para configuração padrão após ? hora	TT10-?
Temporizador 15-15 Retornar para configuração padrão após ? hora	TT15-?
Temporizador 20-20 Retornar para configuração padrão após ? hora	TT20-?
Temporizador 30-30 Retornar para configuração padrão após ? hora	TT30-?
Temporizador 60-60 Retornar para configuração padrão após ? hora	TT60-?

Ligth GSM 1

Comandos suportados por tipos de equipamento **Ligth GSM 1 (eqp = 3 / Puro GPRS)** com Computador de Bordo, nas versões 1.00, 1.01, ou superior

DESCRICAÇÃO	COMANDO
Ligar Setas	1A
Desligar Setas	1B
Ligar Bloqueio	10
Desligar Bloqueio	11
Ligar Sirene	12
Desligar Sirene	13
Última Posição	9C

Comandos suportados por tipos de equipamento **Ligth GSM 1 (eqp = 3 / Puro GPRS)** com Computador de Bordo, **APENAS** na versão 1.00

DESCRICAO	COMANDO
Abrir Baú	00CT1D
Fechar Baú	00CT1E

Comandos suportados por tipos de equipamento **Ligth GSM 1 (eqp = 3 / Puro GPRS)** com Computador de Bordo, **APENAS** na versão 1.01

DESCRICAO	COMANDO
Abrir Baú	00CT1A
Fechar Baú	00CT1B
Habilitar Botão de Abertura de Trava de Baú	00CT5671
Desabilitar Botão de Abertura de Trava de Baú	00CT5670

Satelite Sky

Comandos suportados por tipos de equipamento **Satelite Sky (eqp = 4 / Puro Satelital)**

DESCRICAO	COMANDO
Bloqueador Temporizado	00SK0642000905080A
Ligar Bloqueio	10
Desligar Bloqueio	11
Ligar Sirene	12
Desligar Sirene	13
Última Posição	9C

Slim GSM 1

Comandos suportados por tipos de equipamento **Slim GSM 1 (eqp = 9 / Puro GPRS)**.

DESCRICAO	COMANDO
Ligar Setas	14
Desligar Setas	15
Ligar Bloqueio	10
Desligar Bloqueio	11
Ligar Sirene	12
Desligar Sirene	13
Última Posição	9C

Smart Híbrido

Comandos suportados por tipos de equipamento **SMART HÍBRIDO (eqp = 6)** .

DESCRICAÇÃO	COMANDO
Ativar controle de velocidade	(comando configurável, está descrito no final)
Desativar controle de velocidade	CV0
Ativar controle de RPM	(comando configurável, está descrito no final)
Desativar controle de RPM	RPM0
Desabilitar Botão da Trava da 5ª Roda	T5R0
Desabilitar Botão de Abertura de Trava de Baú	TB0
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Habilitar Botão da Trava da 5ª Roda	T5R
Habilitar Botão da Trava da 5ª Roda. Opcional: Disparo único.	T5R*
Habilitar Botão de Abertura de Trava de Baú 1	TB1
Habilitar Botão de Abertura de Trava de Baú 1 e 2	TB12
Habilitar Botão de Abertura de Trava de Baú 1 e 2. Opcional: Disparo único.	TB12*
Habilitar Botão de Abertura de Trava de Baú 1 e 3	TB13
Habilitar Botão de Abertura de Trava de Baú 1 e 3. Opcional: Disparo único.	TB13*
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3	TB123
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3. Opcional: Disparo único.	TB123*
Habilitar Botão de Abertura de Trava de Baú 1. Opcional: Disparo único.	TB1*
Habilitar Botão de Abertura de Trava de Baú 2	TB2
Habilitar Botão de Abertura de Trava de Baú 2 e 3	TB23
Habilitar Botão de Abertura de Trava de Baú 2 e 3. Opcional: Disparo único.	TB23*
Habilitar Botão de Abertura de Trava de Baú 2. Opcional: Disparo único.	TB2*
Habilitar Botão de Abertura de Trava de Baú 3	TB3

3	
Habilitar Botão de Abertura de Trava de Baú	TB3*
3. Opcional: Disparo único.	
Ligar Aviso de Cabine	09
Desligar Aviso de Cabine	0A
Ligar Bloqueio	01
Ligar Setas	03
Ligar Sirene	05
Solicitar temperatura dos sensores	22
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3
Última Posição	1B
Habilitar sensor de porta do motorista	PMON
Desabilitar sensor de porta do motorista	PMOFF
Habilitar sensor de porta do carona	PPON
Desabilitar sensor de porta do carona	PPOFF
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF
Habilitar sensor de desengate 2	EN2ON
Desabilitar sensor de desengate 2	EN2OFF
Habilita Sensor de Janela do Motorista	JMON
Desabilita Sensor de Janela do Motorista	JMOFF
Habilita Sensor de Janela do Carona	JPON
Desabilita Sensor de Janela do Carona	JPOFF
Habilita Sensor de Painel	VPON
Desabilita Sensor de Painel	VPOFF
Reinicializa Teclado	C32

Comandos Configuráveis:

Ativar controle de velocidade	CV1 valor máximo da velocidade (0 - 220) tempo de ação em segundos (0 - 60) Exemplo: CV1 100 20 à após 20 segundos a 100 km/h será gerado um alerta de velocidade máxima excedida.
Ativar controle de RPM	RPM1 valor máximo da rotação (0 - 15000) tempo de ação em segundos (0 - 60) Exemplo: RPM1 10000 15 à após 15 segundos numa rotação acima de 10000 será gerado um alerta de valor máximo de RPM ultrapassado.

Comandos de Alteração de Temporizador Satelital

Comandos de Temporizador são aceitos pelos equipamentos Smart Híbrido

OBS: As possibilidades aceitas de retorno para a configuração padrão são: 1, 2, 4, 6, 12, 18 horas. Exemplo: TT06-1

DESCRICAO	COMANDO
Temporizador 06-06 Retornar para configuração padrão após ? hora	TT06-?
Temporizador 10-10 Retornar para configuração padrão após ? hora	TT10-?
Temporizador 15-15 Retornar para configuração padrão após ? hora	TT15-?
Temporizador 20-20 Retornar para configuração padrão após ? hora	TT20-?
Temporizador 30-30 Retornar para configuração padrão após ? hora	TT30-?
Temporizador 60-60 Retornar para configuração padrão após ? hora	TT60-?

Smart GPRS

Comandos suportados por tipos de equipamento **SMART GPRS (eqp = 8 / Puro GPRS)**.

DESCRICAO	COMANDO
Ativar controle de velocidade	(comando configurável, está descrito no final)
Desativar controle de velocidade	CV0
Ativar controle de RPM	(comando configurável, está descrito no final)
Desativar controle de RPM	RPM0
Desabilitar Botão da Trava da 5ª Roda	T5R0
Desabilitar Botão de Abertura de Trava de Baú	TB0
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Habilitar Botão da Trava da 5ª Roda	T5R
Habilitar Botão da Trava da 5ª Roda. Opcional: Disparo único.	T5R*

Habilitar Botão de Abertura de Trava de Baú 1	TB1
Habilitar Botão de Abertura de Trava de Baú 1 e 2	TB12
Habilitar Botão de Abertura de Trava de Baú 1 e 2. Opcional: Disparo único.	TB12*
Habilitar Botão de Abertura de Trava de Baú 1 e 3	TB13
Habilitar Botão de Abertura de Trava de Baú 1 e 3. Opcional: Disparo único.	TB13*
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3	TB123
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3. Opcional: Disparo único.	TB123*
Habilitar Botão de Abertura de Trava de Baú 1. Opcional: Disparo único.	TB1*
Habilitar Botão de Abertura de Trava de Baú 2	TB2
Habilitar Botão de Abertura de Trava de Baú 2 e 3	TB23
Habilitar Botão de Abertura de Trava de Baú 2 e 3. Opcional: Disparo único.	TB23*
Habilitar Botão de Abertura de Trava de Baú 2. Opcional: Disparo único.	TB2*
Habilitar Botão de Abertura de Trava de Baú 3	TB3
Habilitar Botão de Abertura de Trava de Baú 3. Opcional: Disparo único.	TB3*
Ligar Aviso de Cabine	09
Desligar Aviso de Cabine	0A
Ligar Bloqueio	01
Ligar Setas	03
Ligar Sirene	05
Solicitar temperatura dos sensores	22
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3
Última Posição	1B
Habilitar sensor de porta do motorista	PMON
Desabilitar sensor de porta do motorista	PMOFF
Habilitar sensor de porta do carona	PPON
Desabilitar sensor de porta do carona	PPOFF
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF
Habilitar sensor de desengate 2	EN2ON
Desabilitar sensor de desengate 2	EN2OFF
Habilita Sensor de Janela do Motorista	JMON

Desabilita Sensor de Janela do Motorista	JMOFF
Habilita Sensor de Janela do Carona	JPON
Desabilita Sensor de Janela do Carona	JPOFF
Habilita Sensor de Painel	VPON
Desbilita Sensor de Painel	VPOFF
Reinicializa Teclado	C32

Comandos Configuráveis:

Ativar controle de velocidade	CV1 valor máximo da velocidade (0 - 220) tempo de ação em segundos (0 - 60) Exemplo: CV1 100 20 à após 20 segundos a 100 km/h será gerado um alerta de velocidade máxima excedida.
Ativar controle de RPM	RPM1 valor máximo da rotação (0 - 15000) tempo de ação em segundos (0 - 60) Exemplo: RPM1 10000 15 à após 15 segundos numa rotação acima de 10000 será gerado um alerta de valor máximo de RPM ultrapassado.

☰ Slim GSM 2

Comandos suportados por tipos de equipamento **Slim GSM 2 (eqp = 11 / Puro GPRS)**.

DESCRICAO	COMANDO
Ligar Bloqueio	C10
Desligar Bloqueio	C11
Ligar Sirene	C14
Desligar Sirene	C15
Última Posição	9C

☰ Slim GSM 3

Comandos suportados por tipos de equipamento **Slim GSM 3 (eqp = 13 / Puro GPRS)**.

DESCRICAO	COMANDO
Ligar Bloqueio	BLQ1

Desligar Bloqueio	BLQ0
Ligar Sirene	SIR1
Desligar Sirene	SIR0
Última Posição	9C
Configurar Velocidade Maxima	(comando configurável, está descrito no final)
Desativar Violação de Painel	DP
Zerar Hodometro	ZH

Comandos Configuráveis:

Configurar Velocidade Maxima	CV (valor máximo da velocidade (0 - 220)) Exemplo: CV100 à a 100 km/h será gerado um alerta de velocidade máxima.
------------------------------	---

Slim GSM 4

Comandos suportados por tipos de equipamento **Slim GSM 4 (eqp = 33 / Puro GPRS)**.

DESCRICAO	COMANDO
Ligar Bloqueio	C10
Desligar Bloqueio	C11
Ligar Sirene	C14
Desligar Sirene	C15
Última Posição	9C

Trailer GSM System

Comandos suportados por tipos de equipamento **Trailer GSM System (eqp = 12 / Puro GPRS)**.

DESCRICAO	COMANDO
Última Posição	9C
Configurar Velocidade Maxima	(comando configurável, está descrito no final)

Comandos Configuráveis:

Configurar Velocidade Maxima	CV (valor máximo da velocidade (0 - 220))
------------------------------	---

	Exemplo: CV100 à a 100 km/h será gerado um alerta de velocidade máxima.
--	--

☐ Ligth GSM 2

Comandos suportados por tipos de equipamento Ligth GSM 2 (eqp = 10 / Puro GPRS).

DESCRICAO	COMANDO
Habilitar Botão de Abertura de Trava de Baú	TB
Desabilitar Botão de Abertura de Trava de Baú	TB0
Habilitar Botão de Abertura de Trava de Baú. Opcional disparo único.	TB*
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Ligar Bloqueio	01
Ligar Setas	03
Ligar Sirene	05
Última Posição	1B
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF

☐ Rail Patrol

Comandos suportados por tipos de equipamento Rail Patrol (eqp = 29)

DESCRICAO	COMANDO
Última Posição	UPFCA
Mensagem de alerta, informa ao operador da mochila que há uma composição (locomotiva) em proximidade.	MAFCA

☐ Smart 2 Híbrido

Comandos suportados por tipos de equipamento SMART 2 HÍBRIDO (eqp = 35) .

DESCRICAO	COMANDO
-----------	---------

Ativar controle de velocidade	(comando configurável, está descrito no final)
Desativar controle de velocidade	CV0
Ativar controle de RPM	(comando configurável, está descrito no final)
Desativar controle de RPM	RPM0
Desabilitar Botão da Trava da 5ª Roda	T5R0
Desabilitar Botão de Abertura de Trava de Baú	TB0
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Habilitar Botão da Trava da 5ª Roda	T5R
Habilitar Botão da Trava da 5ª Roda. Opcional: Disparo único.	T5R*
Habilitar Botão de Abertura de Trava de Baú 1	TB1
Habilitar Botão de Abertura de Trava de Baú 1 e 2	TB12
Habilitar Botão de Abertura de Trava de Baú 1 e 2. Opcional: Disparo único.	TB12*
Habilitar Botão de Abertura de Trava de Baú 1 e 3	TB13
Habilitar Botão de Abertura de Trava de Baú 1 e 3. Opcional: Disparo único.	TB13*
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3	TB123
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3. Opcional: Disparo único.	TB123*
Habilitar Botão de Abertura de Trava de Baú 1. Opcional: Disparo único.	TB1*
Habilitar Botão de Abertura de Trava de Baú 2	TB2
Habilitar Botão de Abertura de Trava de Baú 2 e 3	TB23
Habilitar Botão de Abertura de Trava de Baú 2 e 3. Opcional: Disparo único.	TB23*
Habilitar Botão de Abertura de Trava de Baú 2. Opcional: Disparo único.	TB2*
Habilitar Botão de Abertura de Trava de Baú 3	TB3
Habilitar Botão de Abertura de Trava de Baú 3. Opcional: Disparo único.	TB3*
Ligar Aviso de Cabine	09
Desligar Aviso de Cabine	0A
Ligar Bloqueio	01
Ligar Setas	03

Ligar Sirene	05
Solicitar temperatura dos sensores	22
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3
Última Posição	1B
Habilitar sensor de porta do motorista	PMON
Desabilitar sensor de porta do motorista	PMOFF
Habilitar sensor de porta do carona	PPON
Desabilitar sensor de porta do carona	PPOFF
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF
Habilitar sensor de desengate 2	EN2ON
Desabilitar sensor de desengate 2	EN2OFF
Habilita Sensor de Janela do Motorista	JMON
Desabilita Sensor de Janela do Motorista	JMOFF
Habilita Sensor de Janela do Carona	JPON
Desabilita Sensor de Janela do Carona	JPOFF
Habilita Sensor de Painel	VPON
Desabilita Sensor de Painel	VPOFF
Reinicializa Teclado	C32

Comandos Configuráveis:

Ativar controle de velocidade	CV1 valor máximo da velocidade (0 - 220) tempo de ação em segundos (0 - 60) Exemplo: CV1 100 20 à após 20 segundos a 100 km/h será gerado um alerta de velocidade máxima excedida.
Ativar controle de RPM	RPM1 valor máximo da rotação (0 - 15000) tempo de ação em segundos (0 - 60) Exemplo: RPM1 10000 15 à após 15 segundos numa rotação acima de 10000 será gerado um alerta de valor máximo de RPM ultrapassado.

Comandos de Alteração de Temporizador Satelital

Comandos de Temporizador são aceitos pelos equipamentos Smart 2 Híbrido

OBS: As possibilidades aceitas de retorno para a configuração padrão são: 1, 2, 4, 6,

12, 18 horas. Exemplo: TT06-1

DESCRICAO	COMANDO
Temporizador 06-06 Retornar para configuração padrão após ? hora	TT06-?
Temporizador 10-10 Retornar para configuração padrão após ? hora	TT10-?
Temporizador 15-15 Retornar para configuração padrão após ? hora	TT15-?
Temporizador 20-20 Retornar para configuração padrão após ? hora	TT20-?
Temporizador 30-30 Retornar para configuração padrão após ? hora	TT30-?
Temporizador 60-60 Retornar para configuração padrão após ? hora	TT60-?


Smart 2 GSM

Comandos suportados por tipos de equipamento **SMART 2 GSM (eqp = 36 / Puro GPRS)**.

DESCRICAO	COMANDO
Ativar controle de velocidade	(comando configurável, está descrito no final)
Desativar controle de velocidade	CV0
Ativar controle de RPM	(comando configurável, está descrito no final)
Desativar controle de RPM	RPM0
Desabilitar Botão da Trava da 5ª Roda	T5R0
Desabilitar Botão de Abertura de Trava de Baú	TB0
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Habilitar Botão da Trava da 5ª Roda	T5R
Habilitar Botão da Trava da 5ª Roda. Opcional: Disparo único.	T5R*
Habilitar Botão de Abertura de Trava de Baú 1	TB1
Habilitar Botão de Abertura de Trava de Baú 1 e 2	TB12
Habilitar Botão de Abertura de Trava de Baú 1 e 2. Opcional: Disparo único.	TB12*
Habilitar Botão de Abertura de Trava de Baú	TB13

1 e 3	
Habilitar Botão de Abertura de Trava de Baú 1 e 3. Opcional: Disparo único.	TB13*
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3	TB123
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3. Opcional: Disparo único.	TB123*
Habilitar Botão de Abertura de Trava de Baú 1. Opcional: Disparo único.	TB1*
Habilitar Botão de Abertura de Trava de Baú 2	TB2
Habilitar Botão de Abertura de Trava de Baú 2 e 3	TB23
Habilitar Botão de Abertura de Trava de Baú 2 e 3. Opcional: Disparo único.	TB23*
Habilitar Botão de Abertura de Trava de Baú 2. Opcional: Disparo único.	TB2*
Habilitar Botão de Abertura de Trava de Baú 3	TB3
Habilitar Botão de Abertura de Trava de Baú 3. Opcional: Disparo único.	TB3*
Ligar Aviso de Cabine	09
Desligar Aviso de Cabine	0A
Ligar Bloqueio	01
Ligar Setas	03
Ligar Sirene	05
Solicitar temperatura dos sensores	22
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3
Última Posição	1B
Habilitar sensor de porta do motorista	PMON
Desabilitar sensor de porta do motorista	PMOFF
Habilitar sensor de porta do carona	PPON
Desabilitar sensor de porta do carona	PPOFF
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF
Habilitar sensor de desengate 2	EN2ON
Desabilitar sensor de desengate 2	EN2OFF
Habilita Sensor de Janela do Motorista	JMON
Desabilita Sensor de Janela do Motorista	JMOFF
Habilita Sensor de Janela do Carona	JPON
Desabilita Sensor de Janela do Carona	JPOFF
Habilita Sensor de Painel	VPON
Desbilita Sensor de Painel	VPOFF
Reinicializa Teclado	C32

Comandos Configuráveis:

Ativar controle de velocidade	CV1 valor máximo da velocidade (0 - 220) tempo de ação em segundos (0 - 60) Exemplo: CV1 100 20 à após 20 segundos a 100 km/h será gerado um alerta de velocidade máxima excedida.
Ativar controle de RPM	RPM1 valor máximo da rotação (0 - 15000) tempo de ação em segundos (0 - 60) Exemplo: RPM1 10000 15 à após 15 segundos numa rotação acima de 10000 será gerado um alerta de valor máximo de RPM ultrapassado.

 **Smart Mid Híbrido**

Comandos suportados por tipos de equipamento **SMART MID HÍBRIDO (eqp = 45)**.

DESCRICAO	COMANDO
Ativar controle de velocidade	(comando configurável, está descrito no final)
Desativar controle de velocidade	CV0
Ativar controle de RPM	(comando configurável, está descrito no final)
Desativar controle de RPM	RPM0
Desabilitar Botão da Trava da 5ª Roda	T5R0
Desabilitar Botão de Abertura de Trava de Baú	TB0
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Habilitar Botão da Trava da 5ª Roda	T5R
Habilitar Botão da Trava da 5ª Roda. Opcional: Disparo único.	T5R*
Habilitar Botão de Abertura de Trava de Baú 1	TB1
Habilitar Botão de Abertura de Trava de Baú 1 e 2	TB12
Habilitar Botão de Abertura de Trava de Baú 1 e 2. Opcional: Disparo único.	TB12*

Habilitar Botão de Abertura de Trava de Baú 1 e 3	TB13
Habilitar Botão de Abertura de Trava de Baú 1 e 3. Opcional: Disparo único.	TB13*
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3	TB123
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3. Opcional: Disparo único.	TB123*
Habilitar Botão de Abertura de Trava de Baú 1. Opcional: Disparo único.	TB1*
Habilitar Botão de Abertura de Trava de Baú 2	TB2
Habilitar Botão de Abertura de Trava de Baú 2 e 3	TB23
Habilitar Botão de Abertura de Trava de Baú 2 e 3. Opcional: Disparo único.	TB23*
Habilitar Botão de Abertura de Trava de Baú 2. Opcional: Disparo único.	TB2*
Habilitar Botão de Abertura de Trava de Baú 3	TB3
Habilitar Botão de Abertura de Trava de Baú 3. Opcional: Disparo único.	TB3*
Ligar Aviso de Cabine	09
Desligar Aviso de Cabine	0A
Ligar Bloqueio	01
Ligar Setas	03
Ligar Sirene	05
Solicitar temperatura dos sensores	22
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3
Última Posição	1B
Habilitar sensor de porta do motorista	PMON
Desabilitar sensor de porta do motorista	PMOFF
Habilitar sensor de porta do carona	PPON
Desabilitar sensor de porta do carona	PPOFF
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF
Habilitar sensor de desengate 2	EN2ON
Desabilitar sensor de desengate 2	EN2OFF
Habilita Sensor de Janela do Motorista	JMON
Desabilita Sensor de Janela do Motorista	JMOFF
Habilita Sensor de Janela do Carona	JPON
Desabilita Sensor de Janela do Carona	JPOFF
Habilita Sensor de Painel	VPON
Desbilita Sensor de Painel	VPOFF
Reinicializa Teclado	C32

Comandos Configuráveis:

Ativar controle de velocidade	CV1 valor máximo da velocidade (0 - 220) tempo de ação em segundos (0 - 60) Exemplo: CV1 100 20 à após 20 segundos a 100 km/h será gerado um alerta de velocidade máxima excedida.
Ativar controle de RPM	RPM1 valor máximo da rotação (0 - 15000) tempo de ação em segundos (0 - 60) Exemplo: RPM1 10000 15 à após 15 segundos numa rotação acima de 10000 será gerado um alerta de valor máximo de RPM ultrapassado.

 **Smart Mid GSM**

Comandos suportados por tipos de equipamento **SMART MID GSM (eqp = 46 / Puro GPRS)**.

DESCRICAO	COMANDO
Ativar controle de velocidade	(comando configurável, está descrito no final)
Desativar controle de velocidade	CV0
Ativar controle de RPM	(comando configurável, está descrito no final)
Desativar controle de RPM	RPM0
Desabilitar Botão da Trava da 5ª Roda	T5R0
Desabilitar Botão de Abertura de Trava de Baú	TB0
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Habilitar Botão da Trava da 5ª Roda	T5R
Habilitar Botão da Trava da 5ª Roda. Opcional: Disparo único.	T5R*
Habilitar Botão de Abertura de Trava de Baú 1	TB1
Habilitar Botão de Abertura de Trava de Baú 1 e 2	TB12

Habilitar Botão de Abertura de Trava de Baú 1 e 2. Opcional: Disparo único.	TB12*
Habilitar Botão de Abertura de Trava de Baú 1 e 3	TB13
Habilitar Botão de Abertura de Trava de Baú 1 e 3. Opcional: Disparo único.	TB13*
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3	TB123
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3. Opcional: Disparo único.	TB123*
Habilitar Botão de Abertura de Trava de Baú 1. Opcional: Disparo único.	TB1*
Habilitar Botão de Abertura de Trava de Baú 2	TB2
Habilitar Botão de Abertura de Trava de Baú 2 e 3	TB23
Habilitar Botão de Abertura de Trava de Baú 2 e 3. Opcional: Disparo único.	TB23*
Habilitar Botão de Abertura de Trava de Baú 2. Opcional: Disparo único.	TB2*
Habilitar Botão de Abertura de Trava de Baú 3	TB3
Habilitar Botão de Abertura de Trava de Baú 3. Opcional: Disparo único.	TB3*
Ligar Aviso de Cabine	09
Desligar Aviso de Cabine	0A
Ligar Bloqueio	01
Ligar Setas	03
Ligar Sirene	05
Solicitar temperatura dos sensores	22
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3
Última Posição	1B
Habilitar sensor de porta do motorista	PMON
Desabilitar sensor de porta do motorista	PMOFF
Habilitar sensor de porta do carona	PPON
Desabilitar sensor de porta do carona	PPOFF
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF
Habilitar sensor de desengate 2	EN2ON
Desabilitar sensor de desengate 2	EN2OFF
Habilita Sensor de Janela do Motorista	JMON
Desabilita Sensor de Janela do Motorista	JMOFF
Habilita Sensor de Janela do Carona	JPON
Desabilita Sensor de Janela do Carona	JPOFF
Habilita Sensor de Painel	VPON

Desabilita Sensor de Painel	VPOFF
Reinicializa Teclado	C32

Comandos Configuráveis:

Ativar controle de velocidade	CV1 valor máximo da velocidade (0 - 220) tempo de ação em segundos (0 - 60) Exemplo: CV1 100 20 à após 20 segundos a 100 km/h será gerado um alerta de velocidade máxima excedida.
Ativar controle de RPM	RPM1 valor máximo da rotação (0 - 15000) tempo de ação em segundos (0 - 60) Exemplo: RPM1 10000 15 à após 15 segundos numa rotação acima de 10000 será gerado um alerta de valor máximo de RPM ultrapassado.

Connect Smart Híbrido

Comandos suportados por tipos de equipamento **Connect Smart Híbrido (eqp = 55)** .

DESCRICAO	COMANDO
Ativar controle de velocidade	(comando configurável, está descrito no final)
Desativar controle de velocidade	CV0
Ativar controle de RPM	(comando configurável, está descrito no final)
Desativar controle de RPM	RPM0
Desabilitar Botão da Trava da 5ª Roda	T5R0
Desabilitar Botão de Abertura de Trava de Baú	TB0
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Habilitar Botão da Trava da 5ª Roda	T5R
Habilitar Botão da Trava da 5ª Roda. Opcional: Disparo único.	T5R*
Habilitar Botão de Abertura de Trava de Baú 1	TB1
Habilitar Botão de Abertura de Trava de Baú	TB12

1 e 2	
Habilitar Botão de Abertura de Trava de Baú 1 e 2. Opcional: Disparo único.	TB12*
Habilitar Botão de Abertura de Trava de Baú 1 e 3	TB13
Habilitar Botão de Abertura de Trava de Baú 1 e 3. Opcional: Disparo único.	TB13*
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3	TB123
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3. Opcional: Disparo único.	TB123*
Habilitar Botão de Abertura de Trava de Baú 1. Opcional: Disparo único.	TB1*
Habilitar Botão de Abertura de Trava de Baú 2	TB2
Habilitar Botão de Abertura de Trava de Baú 2 e 3	TB23
Habilitar Botão de Abertura de Trava de Baú 2 e 3. Opcional: Disparo único.	TB23*
Habilitar Botão de Abertura de Trava de Baú 2. Opcional: Disparo único.	TB2*
Habilitar Botão de Abertura de Trava de Baú 3	TB3
Habilitar Botão de Abertura de Trava de Baú 3. Opcional: Disparo único.	TB3*
Ligar Aviso de Cabine	09
Desligar Aviso de Cabine	0A
Ligar Bloqueio	01
Ligar Setas	03
Ligar Sirene	05
Solicitar temperatura dos sensores	22
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3
Última Posição	1B
Habilitar sensor de porta do motorista	PMON
Desabilitar sensor de porta do motorista	PMOFF
Habilitar sensor de porta do carona	PPON
Desabilitar sensor de porta do carona	PPOFF
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF
Habilitar sensor de desengate 2	EN2ON
Desabilitar sensor de desengate 2	EN2OFF
Habilita Sensor de Janela do Motorista	JMON
Desabilita Sensor de Janela do Motorista	JMOFF
Habilita Sensor de Janela do Carona	JPON
Desabilita Sensor de Janela do Carona	JPOFF

Habilita Sensor de Painel	VPON
Desabilita Sensor de Painel	VPOFF
Reinicializa Teclado	C32

Comandos Configuráveis:

Ativar controle de velocidade	CV1 valor máximo da velocidade (0 - 220) tempo de ação em segundos (0 - 60) Exemplo: CV1 100 20 à após 20 segundos a 100 km/h será gerado um alerta de velocidade máxima excedida.
Ativar controle de RPM	RPM1 valor máximo da rotação (0 - 15000) tempo de ação em segundos (0 - 60) Exemplo: RPM1 10000 15 à após 15 segundos numa rotação acima de 10000 será gerado um alerta de valor máximo de RPM ultrapassado.

Comandos de Alteração de Temporizador Satelital

Comandos de Temporizador são aceitos pelos equipamentos Connect Smart Híbrido

OBS: As possibilidades aceitas de retorno para a configuração padrão são: 1, 2, 4, 6, 12, 18 horas. Exemplo: TT06-1

DESCRICAÇÃO	COMANDO
Temporizador 06-06 Retornar para configuração padrão após ? hora	TT06-?
Temporizador 10-10 Retornar para configuração padrão após ? hora	TT10-?
Temporizador 15-15 Retornar para configuração padrão após ? hora	TT15-?
Temporizador 20-20 Retornar para configuração padrão após ? hora	TT20-?
Temporizador 30-30 Retornar para configuração padrão após ? hora	TT30-?
Temporizador 60-60 Retornar para configuração padrão após ? hora	TT60-?

☐ Connect Smart GSM

Comandos suportados por tipos de equipamento **Connect Smart GSM (eqp = 54 / Puro GPRS)**.

DESCRICAO	COMANDO
Ativar controle de velocidade	(comando configurável, está descrito no final)
Desativar controle de velocidade	CV0
Ativar controle de RPM	(comando configurável, está descrito no final)
Desativar controle de RPM	RPM0
Desabilitar Botão da Trava da 5ª Roda	T5R0
Desabilitar Botão de Abertura de Trava de Baú	TB0
Desligar Bloqueio	02
Desligar Setas	04
Desligar Sirene	06
Habilitar Botão da Trava da 5ª Roda	T5R
Habilitar Botão da Trava da 5ª Roda. Opcional: Disparo único.	T5R*
Habilitar Botão de Abertura de Trava de Baú 1	TB1
Habilitar Botão de Abertura de Trava de Baú 1 e 2	TB12
Habilitar Botão de Abertura de Trava de Baú 1 e 2. Opcional: Disparo único.	TB12*
Habilitar Botão de Abertura de Trava de Baú 1 e 3	TB13
Habilitar Botão de Abertura de Trava de Baú 1 e 3. Opcional: Disparo único.	TB13*
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3	TB123
Habilitar Botão de Abertura de Trava de Baú 1, 2 e 3. Opcional: Disparo único.	TB123*
Habilitar Botão de Abertura de Trava de Baú 1. Opcional: Disparo único.	TB1*
Habilitar Botão de Abertura de Trava de Baú 2	TB2
Habilitar Botão de Abertura de Trava de Baú 2 e 3	TB23
Habilitar Botão de Abertura de Trava de Baú 2 e 3. Opcional: Disparo único.	TB23*
Habilitar Botão de Abertura de Trava de Baú 2. Opcional: Disparo único.	TB2*
Habilitar Botão de Abertura de Trava de Baú 3	TB3

3	
Habilitar Botão de Abertura de Trava de Baú	TB3*
3. Opcional: Disparo único.	
Ligar Aviso de Cabine	09
Desligar Aviso de Cabine	0A
Ligar Bloqueio	01
Ligar Setas	03
Ligar Sirene	05
Solicitar temperatura dos sensores	22
Desabilitar Sensor de Temperatura 1	TPT1
Desabilitar Sensor de Temperatura 2	TPT2
Desabilitar Sensor de Temperatura 3	TPT3
Última Posição	1B
Habilitar sensor de porta do motorista	PMON
Desabilitar sensor de porta do motorista	PMOFF
Habilitar sensor de porta do carona	PPON
Desabilitar sensor de porta do carona	PPOFF
Habilitar sensor de desengate 1	EN1ON
Desabilitar sensor de desengate 1	EN1OFF
Habilitar sensor de desengate 2	EN2ON
Desabilitar sensor de desengate 2	EN2OFF
Habilita Sensor de Janela do Motorista	JMON
Desabilita Sensor de Janela do Motorista	JMOFF
Habilita Sensor de Janela do Carona	JPON
Desabilita Sensor de Janela do Carona	JPOFF
Habilita Sensor de Painel	VPON
Desbilita Sensor de Painel	VPOFF
Reinicializa Teclado	C32

Comandos Configuráveis:

Ativar controle de velocidade	CV1 valor máximo da velocidade (0 - 220) tempo de ação em segundos (0 - 60) Exemplo: CV1 100 20 à após 20 segundos a 100 km/h será gerado um alerta de velocidade máxima excedida.
Ativar controle de RPM	RPM1 valor máximo da rotação (0 - 15000) tempo de ação em segundos (0 - 60) Exemplo: RPM1 10000 15 à após 15 segundos numa rotação acima de 10000 será gerado um alerta de valor máximo de RPM ultrapassado.

Configuração de Sensor de Temperatura

Para veículos que permitem utilização de sensores de temperatura, existem os comandos de configuração dos mesmos. Veja o exemplo abaixo de como deve ser gravado no campo mensagem a string do comando para configuração dos sensores de temperatura.

Ex.: TP1+25+10300

TP = Este parâmetro indica que o comando é de configuração do sensor de temperatura. Caso não haja esta identificação o comando será rejeitado.

1 = Este parâmetro indica qual sensor será configurado, existem três sensores, portanto na string do comando não serão aceitos números diferentes de 1, 2, 3 ou 0 (1 indica o sensor de temperatura 1, 2 indica o sensor de temperatura 2, 3 indica o sensor de temperatura 3 e 0 indica todos os sensores. Ao enviar o 0, o comando solicitará os valores de todos os sensores de temperatura instalados).

+25 = Nesta posição, atribui-se o valor do limite superior da temperatura, serão aceitos, somente, valores inteiros que variam de -99 à +99. Valores fora desta faixa, farão com que o comando seja rejeitado.

+10 = Nesta posição, atribui-se o valor do limite inferior da temperatura, serão aceitos, somente, valores inteiros que variam de -99 à +99. Valores fora desta faixa, farão com que o comando seja rejeitado.

300 = Este parâmetro da string de configuração do sensor indica o tempo, em segundos, de violação dos limites estipulados para o sensor. Serão aceitos somente valores que variam de 0 à 65535 segundos.

IMPORTANTE:

1. Para valores de temperatura que não possuem duas casas decimais, deve-se acrescentar um zero a esquerda. Exemplos.: Ao invés de "+6" grave "+06", ao invés de "-4" grave "-04".
2. A temperatura "zero graus" deve ser gravada com um sinal de mais ("+") ou menos ("-"), indiferentemente, seguidos de dois. "+00" ou "-00".
3. Se o limite superior for menor ou igual ao limite inferior, o comando será rejeitado.
4. Se o sistema não localizar o sinal da temperatura ("+" ou "-") na posição adequada da string do comando de configuração, este será rejeitado.
5. As temperaturas provindas do veículo, devido a alguma violação serão enviadas nos campo st1, st2 e st3 de acordo com a quantidade de sensores instalados no veículo.

Funcionamento:

Quando os sensores estiverem configurados da forma desejada eles terão o seguinte funcionamento: Quando a temperatura exceder um dos limites (superior ou inferior) no tempo estipulado na configuração, o computador de bordo enviará um sinal para a central informando qual é a temperatura atual do sensor. Quando a temperatura retornar para a faixa de limites estipulada na configuração, o computador de bordo também enviará um sinal informando a temperatura.

Fica claro que o sistema não irá gerar alertas referente à violação dos limites de temperaturas, tal fato será implementado pela aplicação que usará os dados fornecidos pela Integração.

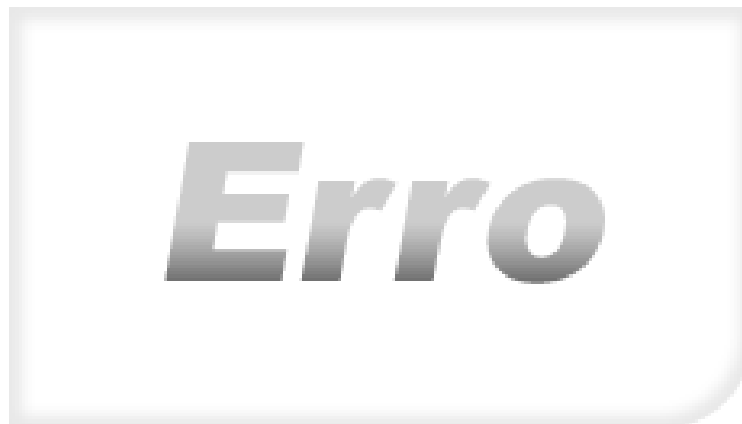
Obs.: Somente nos casos relacionados abaixo este comando deve ser utilizado:

- Veículos do tipo "Satélite" nas versões 2.03 e 2.04 podem possuir até três sensores de temperatura.
- Veículo do tipo "Híbrido" nas versões 3.00, 3.01, 3.02, 3.03 podem possuir apenas um sensor, em quanto versões superiores, tais como 3.04, 3.05, 3.06, 3.07 e 3.08 podem possuir até três sensores de temperatura.
- Veículos do tipo Smart;

Parte



10 Erros Retornados



- [Envio de Comandos](#)
- [Envio de Comandos SPY](#)

10.1 Envio de Comandos

Erros retornados na integração no envio de comandos via requisição.

CÓDIGO	DESCRIÇÃO
*	
1	Veiculo não habilitado para operação. Cliente não possui direitos de envio
2	Erro ao decodificar. Sintaxe incorreta ou falta de parâmetros.
3	Grupo de equipamento não permite envio de configuração de temperatura
4	Valor de configuração de temperatura inválido
5	Sensor destino de temperatura invalida
6	Envio de comando/macro não autorizado. Comando/Macro enviado em menos de 5 minutos ou já existe comando/macro sendo processado.
7	Grupo de equipamento não permite envio de configuração de temporizador
8	Versão (Payload) de comando invalida para grupo de veiculo
9	Grupo de equipamento não permite envio de macro
10	Veiculo não possui teclado de macro
14	Veículo não espelhado
17	Macro inválida para esse equipamento
18	Falha ao processar comando temporizador.
19	Falha ao reconfigurar temporizador.
20	Temporizador já possui configuração default.

21	Formato incorreto de comando temporizador.
22	Comando ou período de temporizador inválido.
23	Sintaxe de comando sensor de temperatura incorreta
25	Veículo não possui sensor referente a esse comando
28	Operação Inválida! O veículo/senha informado são inválidos
29	Parâmetros de contra senha não gerados
30	É necessário definir a senha gerada pelo veículo e apresentada no teclado de macro
31	A senha informada parece ser incorreta, toda senha gerada pelo Computador de Bordo possui 13 dígitos
32	O motivo da geração da contra-senha deve ser informado
33	Índice de grupo de macro a ser embarcado inválido, valores possíveis são entre 1 e 4
34	Inteligência embarcada somente para equipamentos Smart
35	Falha ao recuperar informações computador de bordo
36	Grupo central veículo já embarcado e ativo, para embarcar um novo grupo é necessário limpar os grupos de macros embarcados no cb
37	Grupos veículo central já embarcados (máximo 4), para embarcar um novo grupo é necessário limpar os grupos de macros embarcados no cb
38	Grupo não embarcado (ativação) ou não pertencente ao cliente
39	Já embarcado grupo de macro nessa posição
40	Grupo de macro já embarcado/embarcando
41	Cliente não possui Inteligência Embarcada
42	Grupo de macro já ativo
43	Já existe uma requisição de ativação para esse grupo de macro
44	Grupo de macro não está ativo (requisição de inativação)
45	Requisição para inativação já efetuada para esse grupo de macro
46	Não permitido inativação do grupo, 0 ou 1 grupos VC embarcado
47	Nenhum grupo de macro embarcado
48	Formato senha inválido
49	Intersecção com outro ponto de controle
50	Ponto de Controle incluído em outro ponto de controle (relação contem/contido)
51	Número máximo (400) de ponto de controle embarcado excedido
52	Perfil operacional inválido
53	Ponto de controle inválido (não pertencente ao cliente)
54	Não é possível excluir o ponto de controle (embarcado no veículo)
55	Nenhum ponto de controle embarcado
56	Informações inválidas de espelhamento, não é proprietário ou equipamento não existe.
57	Requisição/espelhamento já efetuada
58	Espelhamento efetuada com sucesso, mas não foi possível realizar a transferência de IE
59	Cancelamento de espelhamento não foi possível pois o servidor está manipulando IE
60	Cliente já está com IE
61	Já existe uma requisição de Transferencia de IE para esse veículo

62	Não existe uma requisição de Transferencia de IE para esse veículo
63	Espelhamento sem direito a IE
64	Ultrapassou o numero de tags enviadas permitidas
65	CNPJ ou CPF informado para Transferência de IE para o cliente inválido.
66	Não foi possível realizar a Operação, o servidor está manipulando IE.
67	Não existe cliente para o CNPJ informado.
68	Não existe espelhamento pendente
69	Equipamento não está espelhado para o cliente informado.
70	Não é permitido o proprietário cancelar o próprio espelhamento.
71	Numero Maximo (50) de Perfil Operacional já embarcado
72	Perfil Operacional já embarcado
73	Perfil Operacional embarcando
74	Perfil Operacional substituindo
75	Perfil Operacional não embarcado
76	Nenhum Perfil Operacional Embarcado para a requisição de limpeza dos perfis
77	Não foi permitido limpar Perfil Operacional, pois contém Ponto de Controle Embarcado
78	Grupo de Macro contém macro preenchível, inválido para alguns equipamentos
79	Valor para configuração de velocidade inválido. 0 - 220. Trailer GSM System e Slim GSM 3
80	Não é possível enviar comando, veículo em manutenção
81	Grupo de macro inválido para esse tipo/versão de equipamento.
82	Grupo de macro não embarcado no equipamento. Operação de Embarque de Ponto de Controle.
83	Tipo de Grupo de Macro inválido (não é Veículo Central). Operação de Embarque de Ponto de Controle.
84	O N° máximo de motoristas embarcados para este veículo foi atingido, para concluir o processo é necessário antes remover um motorista
85	Motorista já embarcado para este veículo.
86	Já existe um comando de embarque e ativação de grupo de macro para esse equipamento.
87	A placa do Cavallo informada, não está espelhada para o cliente.
88	A placa da Carreta informada, não está espelhada para o cliente ou não está cadastrada como Carreta.
89	Esta Carreta já possui vínculo ativo a um Cavallo.
90	Esta Carreta não está vinculada a este Cavallo.
91	Não foi possível realizar o procedimento de Vincular Cavallo/Carreta.
92	A Cerca Eletrônica solicitada, já está ativa neste veículo.
93	A ativação desta Cerca Eletrônica, já está sendo processada.
94	A Cerca Eletrônica solicitada, já está inativa neste veículo.
95	A inativação desta Cerca Eletrônica, já está sendo processada.
96	Cerca Eletrônica não embarcada ou não existente.
97	Perfil Operacional não suportado pelo equipamento.

98	Cerca Eletrônica inexistente ou não é do cliente.
99	Não foi possível recuperar os dados da Cerca Eletrônica.
100	A Cerca Eletrônica, já está embarcada no Veículo.
101	O Projeto de Telemetria não foi encontrado.
102	Não é possível o embarque de Cerca Eletrônica, para este tipo de Equipamento.
103	Proprietário não pode cancelar este espelhamento.
104	Já existe uma viagem habilitada para este veículo.
105	Não é possível realizar o espelhamento com exclusividade, pois esta configuração já foi efetuada.
106	Não é possível enviar comando, pois este veículo já foi espelhado com exclusividade de comando para outro cliente.
107	O campo Identificador não pode ter mais do que 50 caracteres.
108	O campo Localizacao não pode ter mais do que 100 caracteres.
109	O campo Nome não pode ter mais do que 50 caracteres.
110	Este equipamento não suporta o envio de comando de alterar posicionamento com veículo parado (PPCOFF).
111	Comando PPCOFF não autorizado. Cliente não possui direito de envio deste tipo de comando

*** Referente ao código de erro do campo "erro".**

10.2 Envio de Comandos SPY

Erros retornados na integração no envio de comandos via requisição nos dispositivos.

CÓDIGO*	MENSAGEM
1	Spy não habilitado para operação.
2	Erro ao decodificar comando
12	Erro ao processar comando
15	Comando inválido para tipo de spy
16	Informações complementares inválidas de spy

*** Referente ao código de erro do campo "erro".**

Parte



11 O que há de Novo ?

Novidades Versão 5.3 - Documentação Integração

Lançamento: 31 de janeiro de 2019

NOVAS FUNCIONALIDADES

■ Novas Informações da Requisição de Mensagens - RequestMensagemCB

Foram incluídos novos eventos.



Parte



12 Baixar versão em pdf

Para realizar o download deste manual em pdf, [clique aqui](#).