



Instalação , Comandos XML -RPC e Estrutura da base de dados de Integração

Michelin connected fleet Powered by Sascar

REVISION

Date	Version	Author	Description
2011-02-25	3.0	Suporte de Produtos	New revision of document. See changelog.
2001-03-04	3.1	Suporte de Produtos	Inclusion of reset_undo command. See changelog.
2011-04-07	3.2	Suporte de Produtos	Correction name reset_undo command. See changelog.
2011-07-19	3.3	Suporte de Produtos	Alteration topic 2.6 & 2.7. See changelog.
2012-04-24	3.4	Suporte de Produtos	Alteration topic 3.24, 4.3 & 4.4. See changelog.
2012-01-11	3.5	Desenvolvimento de sistemas	Adicionado item 4.5.19. Removido as seções 2 & 3 e suas subseções.
2013-01-15	3.6	Desenvolvimento de sistemas	Retirada seções 2 e 3.
2013-04-24	3.7	Desenvolvimento de sistemas	Atualização de exemplos de retorno XML nas seções 2.4 e 2.5.30.
2013-06-05	3.8	Desenvolvimento de Sistemas	Correção de endereço de conexão para alguns métodos específicos.
2013-08-08	3.9	Desenvolvimento de Sistemas	Atualização de endereço para o comando de modoSeguro.
2013-09-18	4.0	Desenvolvimento de Sistemas	Atualização do retorno para o comando de modoSeguro
2013-10-04	4.1	Desenvolvimento de Sistemas	Atualização do retorno dos comandos
2015-05-02	4.2	Desenvolvimento de Sistemas	Atualização do retorno do comando status_ticket para comandos expirados
2015-05-02	4.3	Suporte de Produtos	Inclusão do método: "embarcar_sequenciamento_macro_sasmdt".
2018-06-15	4.4	CSA	Remoção Embarque de sequenciamento de macros, ASM= 218536, e alteração do nome do método INIBIÇÃO E DESINIBIÇÃO DE SENSORES E ATUADORES EM MODO SEGURO para Ativação e Desativação de Modo Seguro.
2019-09-10	4.5	Desenvolvimento de Sistemas	Inclusão dos métodos "embarcar_layout_grupo_area_avd" e "desembarcar_layout_grupo_area_avd"; remoção dos métodos "embarcar_layout_td40_tipo", "embarcar_layout_td50_tipo" e "embarcar_layout_tmcd_tipo".

2020-06-12	4.6	CSA	Inclusão de Logs de erros e alterado o Intervalo de transmissão satelital para 5 minutos
2021-03-21	4.7	Desenvolvimento de Sistemas	Inclusão dos métodos “vincular_alerta_avd”, “desvincular_alerta_avd”, “inicializar_operacao”, “finalizar_operacao” e “vincular_rota”. Incluído endpoint de “operacao” para possibilitar envio em massa de veículos.
2021-12-20	4.8	CSA	Mudança de HTTP para HTTPS com TLSv1.2, vide item “2.2” e “2.3”

CONFIDENTIALITY TERM

The information contained herein is confidential and the property of SASCAR Tecnologia e Segurança Automotiva S/A (SASCAR). No data be used for other purposes rather than development authorized by SASCAR and cannot be shared with other parties outside your company without previous written authorization from SASCAR. It is forbidden to generate photocopies of this document, along with any kind of reproduction or distribution, partially or in its integrity, by any means, including graphics, magnetic, photographic or electronic. SASCAR reserves its right to alter this document without prior notice and to require the devolution of this document at the end of development process.

As informações contidas neste documento são confidenciais e de propriedade da SASCAR Tecnologia e Segurança Automotiva S/A (SASCAR). Nenhuma informação pode ser usada para outros propósitos exceto o desenvolvimento com autorização da SASCAR, não podendo ser compartilhado com outros fora de sua empresa sem prévia autorização escrita da SASCAR. É proibido fazer fotocópias deste documento, bem como sua reprodução ou distribuição, parcialmente ou em sua inteireza, por qualquer meio, incluindo gráficos, magnéticos, fotografias ou meios eletrônicos. SASCAR se reserva ao direito de alterar este documento sem prévio aviso e também de exigir a devolução deste documento ao final do processo de desenvolvimento.

Sumário

1.	INTRODUÇÃO.....	7
2.	COMANDOS XML-RPC.....	8
2.1.	OBJETIVO	8
2.2.	DESENVOLVIMENTO	8
2.3.	ENDEREÇOS	9
2.4.	RETORNO DO XML	9
2.4.1.	LOG DE ERROS	11
2.5.	COMANDOS	14
2.5.1.	POSIÇÃO	14
2.5.2.	BLOQUEIO.....	15
2.5.3.	DESBLOQUEIO.....	16
2.5.4.	ATUAÇÃO DE SAÍDAS	17
2.5.5.	ENVIO DE MENSAGEM DE TEXTO	18
2.5.6.	CONFIGURA TRANSMISSÃO COM IGNIÇÃO DESLIGADA	19
2.5.7.	LISTAGEM DE COMANDOS ENVIADOS	20
2.5.8.	RESET DE ALARME DE SEQUENCIAMENTO DE MACROS	23
2.5.9.	GERAR CONTRA SENHA TECLADO TD40/TMCD	24
2.5.10.	GERAR CONTRA SENHA TECLADO TD50	25
2.5.11.	INTERVALO DE ANÁLISE SATELITAL	27
2.5.12.	INTERVALO DE TRANSMISSÃO SATELITAL	28
2.5.13.	ALTERAÇÃO DE TEMPO DE TRANSMISSÃO GPRS	29
2.5.14.	INIBIÇÃO DE SENSORES.....	30
2.5.15.	ATIVAÇÃO E DESATIVAÇÃO DE MODO SEGURO DOS SENSORES E ATUADORES	31
2.5.16.	EMBARCAR LAYOUT DE AÇÃO EMBARCADA AVD	33
2.5.17.	DESEMBARCAR LAYOUT DE AÇÃO EMBARCADA AVD	34
2.5.18.	EMBARCAR LAYOUT DE GRUPO DE PONTOS.....	35
2.5.19.	DESEMBARCAR LAYOUT DE GRUPO DE PONTOS.....	36
2.5.20.	EMBARCAR MOTORISTA	37
2.5.21.	DESEMBARCAR MOTORISTA	38
2.5.22.	EMBARCAR LAYOUT TMCD	39
2.5.23.	EMBARCAR LAYOUT TD40.....	40
2.5.24.	EMBARCAR LAYOUT TD50.....	41
2.5.25.	EMBARCAR SEQUENCIAMENTO DE MACROS	42

2.5.26.	EMBARCAR SEQUENCIAMENTO DE MACROS SASMDT.....	43
2.5.27.	EMBARCAR LAYOUT DE GRUPO DE ÁREA AVD.....	44
2.5.28.	DESEMBARCAR LAYOUT DE GRUPO DE ÁREA AVD.....	46
2.5.29.	CONSULTA STATUS DE UM COMANDO ENVIADO	47
2.5.30.	VINCULAR ALERTA AVD AO VEÍCULO	51
2.5.31.	DESVINCULAR ALERTA AVD AO VEÍCULO	54
2.5.32.	INICIALIZAR OPERAÇÃO	56
2.5.33.	FINALIZAR OPERAÇÃO	58
2.5.34.	VINCULAR ROTA AO VEÍCULO	60
2.6.	SUPORTE.....	62
2.7.	RELEASE NOTES INTEGRAÇÃO	62
2.8.	CHANGE LOG DOCUMENTAÇÃO INTEGRAÇÃO	62

1. Introdução

Este documento tem por finalidade documentar a estrutura da base de dados de integração, bem como definir sua respectiva utilização. Aborda desde a instalação do servidor até a documentação relativa a envio de comandos para os rastreadores instalados nos veículos. Essas informações serão úteis durante o desenvolvimento de aplicações que utilizem a base de dados de integração.

A SASCAR se limita a criar a estrutura de tabelas de banco de dados e enviar as informações para o servidor informado em formulário específico para tal. Não é responsável por como as informações serão utilizadas pelo INTEGRADOR. As informações de rastreamento somente são disponibilizadas ao INTEGRADOR mediante autorização expressa do cliente, passando o INTEGRADOR a responder por qualquer uso indevido de informação privilegiada do CLIENTE. A SASCAR não se responsabiliza pela manutenção e segurança do servidor de banco de dados que a ela foi concedido acesso. Tais medidas são de responsabilidade do INTEGRADOR, ficando o mesmo também designado como responsável pela guarda das informações dos CLIENTES a ele direcionados.

2. Comandos XML-RPC

A SASCAR disponibiliza um webservice onde é possível enviar comandos aos veículos, como bloqueio, sirene, acionamento e desativação de travas, etc. Os comandos possuem estrutura XML-RPC (<http://www.xmlrpc.com>). Este documento tem por finalidade registrar os comandos possíveis de envio por XML-RPC, bem como definir sua respectiva utilização. Tais dados devem ser utilizados para o desenvolvimento de aplicações com o objetivo de integração de dados. Novas implementações foram feitas, sendo necessário, portanto, um estudo cuidadoso das mudanças realizadas para que a aplicação integradora atenda as novas especificações. A metodologia de envio de comandos continua a mesma. Para evitarmos ao máximo a quantidade de mensagens de erro, fizemos alterações na parte de validação do XML-RPC, para garantir que o mesmo corresponde ao que está no veículo. Outra mudança é que, a partir de agora, os comandos terão como resposta um valor de um ticket interno. O que é isso? A partir de agora, se o comando for válido e aceito pelo servidor, ele irá gerar um ticket de retorno. Se o desenvolvedor quiser saber se o comando foi executado ou não, ele deverá enviar uma requisição de XML-RPC conforme descrito adiante para consulta do status do comando. Ou seja, se o ticket for retornado, significa que o comando é válido e foi aceito pelo servidor, mas para saber se o mesmo foi executado no veículo, o desenvolvedor deve aguardar a tabela comando do banco de integração ser atualizada ou enviar essa requisição XML-RPC, que deverá conter o ticket que foi enviado no comando e o ticket interno.

IMPORTANTE: Não poderão existir tickets repetidos para um mesmo veículo. Visto que o ticket interno é apenas um retorno XML, ele não será gravado na tabela de integração. É de responsabilidade de o desenvolvedor armazenar esse valor para consulta. Abaixo, um descritivo dos comandos XML-RPC disponíveis para envio.

2.1. Objetivo

Criar um webservice de comunicação que permita o envio de comandos utilizando protocolo XML-RPC de forma a evitar atrasos no retorno do comando aos integradores.

2.2. Desenvolvimento

Para realização de testes de desenvolvimento ou conexão com o servidor, pode-se utilizar a seguinte configuração de autenticação:

host: http://xmlrpc.sascar.com.br/xmlrpc/enviar_comando

host: https://xmlrpc.sascar.com.br/xmlrpc/enviar_comando

Login: teste_login

Senha: teste_senha

Placa: 0000AAA

ticket: qualquer número entre 1 e 2.147.483.647. **Não pode ser repetido.**

2.3. Endereços

O XML do comando desejado deve ser enviado via POST para o seguinte endereço:

https://xmlrpc.sascar.com.br/xmlrpc/enviar_comando

!! Atenção!!

A Sascar, prezando a segurança nas informações de seus clientes, está descontinuando o protocolo HTTP para o serviço "XMLRPC.SASCAR.COM.BR".

Seguindo as recomendações do IETF (Internet Engineering Task Force), a comunicação se dará através do protocolo HTTPS com TLS v1.2.

Devido a essa mudança, os clientes deverão validar se suas aplicações suportam a comunicação através deste protocolo, e caso necessário, realizar os ajustes para suportar HTTPS com TLS v1.2.

2.4. Retorno do XML

Para cada comando XML enviado ao servidor, um ticket interno será retornado. Esse ticket não será retornado quando:

- * Algum parâmetro do XML estiver incorreto;
- * Houver algum retorno com faultString.
- * Não for um comando com execução no módulo rastreador.

Exemplo: Funções do banco, consulta de comandos.

Abaixo temos um exemplo de retorno do XML, para um comando que foi executado com sucesso no servidor:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>ticketServidor</name>
            <value>
              <string>12132678</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

OBS: Outra possível forma de retorno, encontrada exclusivamente no comando **modoSeguro**, é do seguinte modelo abaixo:

```
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>statusComando</name>
            <value>
              <string>7</string>
            </value>
          </member>
          <member>
            <name>ticketServidor</name>
            <value>
              <string>12132678</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

OBS: Uma outra possível forma de retorno, encontrada exclusivamente no comando **inibir_sensor**, é do seguinte modelo abaixo:

```
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>successCode</name>
            <value>
              <string>103</string>
            </value>
          </member>
          <member>
            <name>successString</name>
            <value>
              <string>Funcoes executadas</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

Caso o servidor encontre algum erro na execução do comando, será retornado um XML do tipo **faultCode** que representa o tipo de erro ocorrido (ver seção 2.4.1) e a descrição do erro ocorrido em **faultString**.

A estrutura de erro retornada é apresentada abaixo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<fault>
  <value>
    <struct>
      <member>
        <name>faultCode</name>
        <value>
          <int>14</int>
        </value>
      </member>
      <member>
        <name>
          faultString</name>
        <value>
          <string>Erro ao ler xml</string>
        </value>
      </member>
    </struct>
  </value>
</fault>
```

2.4.1. Log de erros

A lista de erros que podem ser retornados no campo **faultCode** e as suas causas são apresentadas na tabela abaixo:

CODIGO	MENSAGEM	MOTIVO
1	Layout Acao Embarcada Avd Nao Encontrado	Layout Ação Embarcada Avd Não Encontrado
1	Layout Sem Acoes cadastradas	Layout Ação Embarcada Avd sem ações cadastradas
1	Layout Sem pontos ou Inexistente	Caso layout não tenha nenhum ponto de referência cadastrado ou não pertença a gerenciadora
1	Erro ao acessar a base remota de dados da gerenciadora	Se no comando <code>inibir_sensor</code> houver algum erro ao acessar a base da Integração
1	Erro ao gravar log de inibicao	Se no comando <code>inibir_sensor</code> ocorrer algum erro ao atualizar a base da Integração ou na base da Sascar

1	Erro ao executar comando	Se houver algum erro ao executar o comando para o cliente
1	Erro ao gerar ticket	Se houver algum erro ao gerar ticket do comando para o cliente
4	Comando nao suportado	Se o equipamento contratado possui restrição de envio de comando
6	Erro na autenticacao	Se os dados da integração não forem encontrados na base
7	Veiculo nao disponivel para esse login	Se o veículo não estiver direcionado para a integração

8	layout nao preenchido	Se o codigo do layout (Teclado, Grupo de Pontos ou Ação Embarcada AVD) não estiver preenchido
8	Nenhum resultado retornado para a consulta efetuada	No comando listar_comandos se não existir comandos a serem listados na data pesquisada
8	acao	Se no comando inibir_sensor o campo ação estiver nulo, vazio e for diferente de 0 ou 1
8	configuracao	Se no comando inibir_sensor o valor do campo configuracao for inválido
8	intervalo invalido	Se nos comandos analise_satelital ou relatorio_satelital o campo intervalo for nulo ou ≤ 0
8	quantidade invalida	Se no comando listar_comandos o campo quantidade for nulo
8	data inicial invalida	Se no comando listar_comandos a data inicial for inválida
8	data final invalida	Se no comando listar_comandos a data final for inválida
8	comando atuador	Se no comando atuador o campo configuração estiver com valores invalidos
8	configuracao	Se no comando atuador for informado parametros diferentes de on ou off
8	comando transmissao desligada	Se no comando transmissao_ignicao_desligada o parametro for diferente de on ou off
8	Erro nos parametros	Se o usuário, senha, placa ou o campo ticket estiver nulo
14	Erro ao ler xml	Se o xml enviado for nulo, vazio ou divergente do formato esperado
50	Erro, dispositivo inexistente para o veiculo	Dispositivo inexistente para o veículo
50	Erro, dispositivo inexistente para o veiculo	Se no comando texto o veículo não possuir teclado
50	Erro, dispositivo inexistente para o veiculo	Se no comando atuador for informado um dispositivo inexistente no veículo
52	O veiculo nao possui o sensor	Se no comando inibir_sensor o veículo não possuir o sensor informado
255	Erro nao esperado ao recuperar mensagem	Se houver um erro interno ao validar os dados da integração
14	Placa não localizada	Placa informada não está vinculado a está gerenciadora ou não existe nos cadastros.
2	Equipamento offline	Equipamento esta offline
8	Atuador/Sensor não encontrado para o veiculo.	Se no comando atuador/Sensor for informado um dispositivo inexistente no veiculo

2.5. Comandos

Abaixo listamos os comandos disponíveis até a criação da versão corrente deste documento.

2.5.1. Posição

Aplicabilidade: Requisitar uma posição.

Comando: posicao

Parâmetros: Sem parâmetros

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>posicao</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.2. Bloqueio

Aplicabilidade: Envia um comando de bloqueio ao veículo.

Comando: bloqueio

Parâmetros: Sem parâmetros

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>bloqueio</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.3. Desbloqueio

Aplicabilidade: Desbloqueia o veículo.

Comando: desbloqueio

Parâmetros: Sem parâmetros

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>desbloqueio</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.4. Atuação de saídas

Aplicabilidade: Altera o estado de atuadores.

Comando: atuador

Parâmetros: Código do atuador: Ex.: 240 (sirene) [esse código é encontrado na tabela atuador_grupo]

Estado do atuador: On ou Off

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>atuador</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>configuracao</name>
            <value>
              <array>
                <data>
                  <value><int>240</int></value>
                  <value><string>On</string></value>
                </data>
              </array>
            </value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.5. Envio de mensagem de texto

Aplicabilidade: Envia texto livre para teclado.

Comando: texto

Parâmetros: Sequência de caracteres ASCII

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>texto</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>texto</name>
            <value><string>sequencia de caracteres asc2</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.6. Configura transmissão com ignição desligada

Aplicabilidade: Habilita a transmissão de posições com a ignição desligada. Comando disponível somente para MTC401. **IMPORTANTE: O envio desse comando poderá ultrapassar o pacote contratado pelo cliente, gerando custos adicionais. Somente enviar esse comando com o consentimento do cliente.**

Comando: transmissao_ignicao_desligada

Parâmetros: Estado: On para ligar, Off para desligar.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>transmissao_ignicao_desligada</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
          <member>
            <name>estado</name>
            <value><string>Off</string></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.7. Listagem de comandos enviados

Aplicabilidade: Requisitar uma listagem dos comandos enviados para um veículo. O retorno será o comando juntamente com o status do comando no momento que foi requisitado. A quantidade é partindo da data final para a data inicial. **IMPORTANTE: Para esse comando não deve ser enviado o parâmetro ticket, pois é uma consulta dos comandos enviados.**

Comando: listar_comandos

Parâmetros: Quantidade de itens (1 a 999)

Data Inicial – Data inicial de envio do comando. Formato: mm/dd/aaaa hh:mm.

Data Final – Data final de envio do comando. Formato: mm/dd/aaaa hh:mm.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>listar_comandos</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
          <member>
            <name>quantidade</name>
            <value><int>1</int></value>
          </member>
          <member>
            <name>data_inicial</name>
            <value><string>01/03/2011 00:00</string></value>
          </member>
          <member>
            <name>data_final</name>
            <value><string>01/03/2011 23:59</string></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Michelin connected fleet Powered by Sascar

Listagem de comandos enviados (continuação):

Estrutura de resposta: A resposta é feita através de um array de struct. O struct é dividido em dois membros: Indicador de item e dados. O membro dados é um array com os seguintes elementos:

- Gerenciadora ou cliente que enviou o comando
- Login usado para envio de comando
- Comando
- Parâmetros do comando, caso existam
- Meio de transmissão: GPRS, SATELITAL, etc.
- Data Envio do comando
- Data da Execução do comando, caso tenha sido executado.
- Data da Expiração do comando, caso o comando não tenha sido executado e foi descartado pelo servidor de envio.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
<params>
  <param>
    <value>
      <struct>
        <member>
          <name>successCode</name>
          <value>
            <int>0</int>
          </value>
        </member>
        <member>
          <name>successString</name>
          <value>
            <array>
              <data>
                <value>
                  <struct>
                    <member>
                      <name>item</name>
                      <value>
                        <int>1</int>
                      </value>
                    </member>
                    <member>
                      <name>dados</name>
                      <value>
                        <array>
                          <data>
                            <value>
                              <string>NOMEDAGERENCIADORA</string>
                            </value>
                            <value>
                              <string>LOGINQUEENVIOUCOMANDO</string>
                            </value>
                            <value>
                              <string>Horimetro</string>
                            </value>
                          </data>
                        </array>
                      </value>
                    </member>
                  </struct>
                </value>
              </data>
            </array>
          </value>
        </member>
      </struct>
    </value>
  </param>
</params>
</methodResponse>
```

Michelin connected fleet Powered by Sascar

```
</value>
<value>
  <string>Tempo: 500000 minutos</string>
</value>
<value>
  <string>GPRS</string>
</value>
<value>
  <string>01/03/2011 07:00:10</string>
</value>
<value>
  <string>01/03/2011 07:02:14</string>
</value>
<value>
  <string/>
</value>
</data>
</array>
</value>
</member>
</struct>
</value>
</data>
</array>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

2.5.8. Reset de alarme de sequenciamento de macros

Aplicabilidade: Envia o comando de reset de alarme mantendo o perfil de segurança ou reset de alarme limpando o perfil de segurança do sequenciamento de macros.

Comando: reset_undo_alarme

Parâmetros: set (0 = Manter perfil de segurança / 2 = Limpar perfil de segurança)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>reset_undo_alarme</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>set</name>
            <value><int>0</int></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.9. Gerar contra senha teclado TD40/TMCD

Aplicabilidade: Gerar uma contra senha para manipulação de saídas pelo teclado TD40 ou TMCD.

Comando: gerar_contra_senha

Parâmetros: Senha (essa informação é passada pelo condutor através da geração de senha no teclado)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>gerar_contra_senha</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
          <member>
            <name>senha</name>
            <value><int>002</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Exemplo de resposta para comando gerar_contra_senha:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
<params>
  <param>
    <value>
      <struct>
        <member>
          <name>successCode</name>
          <value>
            <int>9</int>
          </value>
        </member>
        <member>
          <name>successString</name>
          <value><string>12345</string></value>
        </member>
      </struct>
    </value>
  </param>
</params>
</methodResponse>
```

2.5.10. Gerar contra senha teclado TD50

Aplicabilidade: Gera uma contra senha para liberar ativação de atuadores via teclado TD50.

Comando: gerar_contra_senha_mtc600

Parâmetros: senha alfanumérica de 15 dígitos fornecida pelo condutor.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>gerar_contra_senha_mtc600</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAAA</string></value>
          </member>
          <member>
            <name>senha</name>
            <value><string>ABCD1-1234B-AAAA0</string></value>
          </member>
          <member>
            <name>seguranca_entrada</name>
            <value><string>off</string></value>
          </member>
          <member>
            <name>seguranca_saida</name>
            <value><string>off</string></value>
          </member>
          <member>
            <name>acoes</name>
            <value><string>off</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Michelin connected fleet Powered by Sascar

Exemplo de resposta para comando gerar_contra_senha_mtc600:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
<params>
  <param>
    <value>
      <struct>
        <member>
          <name>successCode</name>
          <value>
            <int>13</int>
          </value>
        </member>
        <member>
          <name>successString</name>
          <value>
            <string>A12E37</string>
          </value>
        </member>
      </struct>
    </value>
  </param>
</params>
</methodResponse>
```

2.5.11. Intervalo de Análise Satelital

Aplicabilidade: Altera o tempo de análise para entrar em modo satelital. Para saber se o veículo possui antena satelital, veja a informação da coluna *veisatelital* na tabela *veiculo*. **IMPORTANTE: O envio desse comando poderá implicar em ultrapassar o pacote contratado pelo cliente, gerando custos adicionais. Somente enviar esse comando com o consentimento do cliente.**

Comando: `analise_satelital`

Parâmetros: Intervalo (em segundos) maior que 0.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>analise_satelital</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
          <member>
            <name>intervalo</name>
            <value><int>600</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.12. Intervalo de transmissão satelital

Aplicabilidade: Altera o tempo de transmissão do equipamento quando em modo satelital.

IMPORTANTE: O envio desse comando poderá implicar em ultrapassar o pacote contratado pelo cliente, gerando custos adicionais. Somente enviar esse comando com o consentimento do cliente.

Comando: relatorio_satelital

Parâmetros: Intervalo (em segundos) maior que 300.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>relatorio_satelital</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
          <member>
            <name>intervalo</name>
            <value><int>600</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.13. Alteração de tempo de transmissão GPRS

Aplicabilidade: Altera o tempo de envio de posições quando o equipamento estiver em GPRS.

Comando: relatorio

Parâmetros: intervalo (em segundos) maior que 0.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>relatorio</methodName>
  <params>
    <param>
      <param>
        <value>
          <struct>
            <member>
              <name>login</name>
              <value><string>teste_login</string></value>
            </member>
            <member>
              <name>password</name>
              <value><string>teste_senha</string></value>
            </member>
            <member>
              <name>placa</name>
              <value><string>0000AAA</string></value>
            </member>
            <member>
              <name>intervalo</name>
              <value><int>120</int></value>
            </member>
            <member>
              <name>ticket</name>
              <value><int>123456</int></value>
            </member>
          </struct>
        </value>
      </param>
    </params>
  </methodCall>
```

2.5.14. Inibição de Sensores

Aplicabilidade: Insere um registro na tabela mascara_dispositivo. Se um sensor estiver 'inibido', a chegada de alertas na tabela de eventos para os sensores inibidos não ocorre. O retorno em caso de sucesso na execução do comando, será de "Funções Executadas", SuccessCode 103, porém, não é enviado nenhum comando ao equipamento instalado no veículo, sendo somente alterado um registro em banco de dados. Esse comando é somente para SENSORES. Não há comando para inibir ATUADORES.

Comando: inibir_sensor

Parâmetros: Ação (0 para desinibir um sensor e 1 para inibir o sensor);

Configuração (atgoid do sensor que deseja inibir/desinibir). Se enviado 0 executa a ação (0 ou 1) para todos os sensores.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>inibir_sensor</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>acao</name>
            <value><string>0</string></value>
          </member>
          <member>
            <name>configuracao</name>
            <value>
              <array>
                <data>
                  <value><int>231</int></value>
                  <value><int>241</int></value>
                  <value><int>248</int></value>
                </data>
              </array>
            </value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.15. Ativação e Desativação de Modo Seguro dos Sensores e Atuadores

Aplicabilidade: Com este comando pode-se inibir determinado sensor ou atuador do veículo, ou todos os sensores e atuadores do veículo de uma única vez.

Comando: modoSeguro

Parâmetros:

Configuração: (atgoid do sensor que deseja inibir/desinibir). Se enviado 0 executa a ação (0 ou 1) para todos os sensores. On / Off é para ativar ou desativar os sensores.

TipoComando: sensor vai atuar nos sensores do veículo. Já o parâmetro atuador o comando será enviado para os atuadores do veículo.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>modoSeguro</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>configuracao</name>
            <value>
              <array>
                <data>
                  <value><int>0</int></value>
                  <value><string>On</string></value>
                </data>
              </array>
            </value>
          </member>
          <member>
            <name>tipoComando</name>
            <value><string>sensor</string></value>
          </member>
          <member>
            <name>ticket</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```


2.5.16. Embarcar Layout de Ação Embarcada AVD

Aplicabilidade: Permite enviar a um equipamento o layout com suas devidas ações embarcadas criadas anteriormente.

Comando: embarcar_layout_acao_embarcada_avd

Parâmetros: laemoid (id dos layouts, dado disponível na tabela layout_acao_embarcada_mtc600).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>embarcar_layout_acao_embarcada_avd</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>layout</name>
<value>
<array>
<data>
<value><int>12345</int></value>
</data>
</array>
</value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.17. Desembarcar Layout de Ação Embarcada AVD

Aplicabilidade: Comando efetua o desembarque de qualquer layout que esteja embarcado no veículo.

Comando: desembarcar_layout_acao_embarcada_avd

Parâmetros: Sem parâmetros.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>desembarcar_layout_acao_embarcada_avd</methodName>
<params>
<param>
<value>
  <struct>
    <member>
      <name>login</name>
      <value><string>teste_login</string></value>
    </member>
    <member>
      <name>password</name>
      <value><string>teste_senha</string></value>
    </member>
    <member>
      <name>placa</name>
      <value><string>AAA0000</string></value>
    </member>
    <member>
      <name>ticket</name>
      <value><int>9999999</int></value>
    </member>
  </struct>
</value>
</param>
</params>
</methodCall>
```

2.5.18. Embarcar Layout de Grupo de Pontos

Aplicabilidade: Efetua o embarque do grupo de pontos no veículo informado.

Comando: embarcar_layout_grupo_ponto

Parâmetros: Igpoid (Id do layout, dado disponível na tabela layout_grupo_ponto)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>embarcar_layout_grupo_ponto</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>layout</name>
<value><int>1</int></value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.19. Desembarcar Layout de Grupo de Pontos

Aplicabilidade: Efetua o desembarque de qualquer grupo de pontos que esteja embarcado no veículo.

Comando: desembarcar_layout_grupo_ponto

Parâmetros: Sem parâmetros.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>desembarcar_layout_grupo_ponto</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.20. Embarcar Motorista

Aplicabilidade: Embarca motoristas no veículo.

Comando: embarcar_motorista

Parâmetros: motooid (id dos motoristas, esse dado está disponível na tabela motorista)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>embarcar_motorista</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>motoristas</name>
<value>
<array>
<data>
<value><int>12345</int></value>
<value><int>11111</int></value>
<value><int>54321</int></value>
</data>
</array>
</value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.21. Desembarcar Motorista

Aplicabilidade: Efetua o desembarque de motoristas no veículo.

Comando: desembarcar_motorista

Parâmetros: motooid (id dos motoristas, esse dado está disponível na tabela motorista)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>desembarcar_motorista</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>motoristas</name>
<value>
<array>
<data>
<value><int>12345</int></value>
<value><int>11111</int></value>
<value><int>54321</int></value>
</data>
</array>
</value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.22. Embarcar Layout TMCD

Aplicabilidade: Embarcar um layout de teclado para um veículo com o teclado TMCD. Essa informação está disponível nas colunas veiserial0 e veiserial1 da tabela veículo.

Comando: embarcar_layout_tmcd

Parâmetros: Id do layout (Esse dado está disponível no campo tltmcdcodigogc que pode ser encontrado na tabela tipo_layout_teclado_tmcd).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>embarcar_layout_tmcd</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>layout</name>
<value><int>1</int></value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.23. Embarcar Layout TD40

Aplicabilidade: Embarcar um layout de teclado para um veículo com o teclado TD40. Essa informação está disponível nas colunas veiserial0 e veiserial1 da tabela veículo.

Comando: embarcar_layout_td40

Parâmetros: Id do layout (Esse dado está disponível no campo tlccodigogc que pode ser encontrado na tabela tipo_layout_teclado).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>embarcar_layout_td40</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>layout</name>
<value><int>1</int></value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.24. Embarcar Layout TD50

Aplicabilidade: Embarcar um layout de teclado para um veículo com o teclado TD50. Essa informação está disponível nas colunas veiserial0 e veiserial1 da tabela veículo.

Comando: embarcar_layout_td50

Parâmetros: Id do layout (Esse dado está disponível no campo tltd50codigogc que pode ser encontrado na tabela tipo_layout_teclado_td50).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>embarcar_layout_td50</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>layout</name>
<value><int>1</int></value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.25. Embarcar Sequenciamento de Macros

Aplicabilidade: Permite enviar a um veículo equipamento com teclado TD50 o sequenciamento de macros. O sequenciamento de macros precisa ter sido criado e ter sido enviado ao SASGC.

Comando: embarcar_sequenciamento_td50

Parâmetros: Id do layout (Esse dado está disponível no campo tsmttd50codigogc que pode ser encontrado na tabela tipo_sequenciamento_macro_td50).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>embarcar_sequenciamento_td50</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>layout</name>
<value><int>1</int></value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.26. Embarcar Sequenciamento de Macros SASMDT

Aplicabilidade: Permite enviar a um veículo equipamento com teclado SASMDT o sequenciamento de macros. O sequenciamento de macros precisa ter sido criado e ter sido enviado ao SASGC.

Comando: embarcar_sequenciamento_macro_sasmdt

Parâmetros: Id do layout que será enviado ao veículo.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>embarcar_sequenciamento_macro_sasmdt</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>login</name>
<value><string>teste_login</string></value>
</member>
<member>
<name>password</name>
<value><string>teste_senha</string></value>
</member>
<member>
<name>placa</name>
<value><string>AAA0000</string></value>
</member>
<member>
<name>layout</name>
<value><int>1</int></value>
</member>
<member>
<name>ticket</name>
<value><int>9999999</int></value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

2.5.27. Embarcar Layout de Grupo de Área AVD

Aplicabilidade: Requisita envio de um layout de grupo de área a um determinado veículo. O Layout de Grupo de Área AVD deve estar criado no SASGC. Os id de layouts existentes são obtidos via Sasintegra.

Comando: embarcar_layout_grupo_area_avd

Parâmetros: Id do layout que será enviado ao veículo.

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

<methodCall>

  <methodName>embarcar_layout_grupo_area_avd</methodName>

  <params>

    <param>

      <value>

        <struct>

          <member>

            <name>login</name>

            <value>

              <string>teste_login</string>

            </value>

          </member>

          <member>

            <name>password</name>

            <value>

              <string>teste_senha</string>

            </value>

          </member>

          <member>

            <name>placa</name>

            <value>
```

Michelin connected fleet Powered by Sascar

```
<string>AAA0000</string>

</value>

</member>

<member>

  <name>layout</name>

  <value>

    <int>111</int>

  </value>

</member>

<member>

  <name>ticket</name>

  <value>

    <int>99999999</int>

  </value>

</member>

</struct>

</value>

</param>

</params>

</methodCall>
```

2.5.28. Desembarcar Layout de Grupo de Área AVD

Aplicabilidade: Requisita o desembarque de grupo de área AVD de um veículo.

Comando: desembarcar_layout_grupo_area_avd

Parâmetros: sem parâmetros.

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>desembarcar_layout_grupo_area_avd</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value>
              <string>teste_login</string>
            </value>
          </member>
          <member>
            <name>password</name>
            <value>
              <string>teste_senha</string>
            </value>
          </member>
          <member>
            <name>placa</name>
            <value>
              <string>AAA0000</string>
            </value>
          </member>
          <member>
            <name>ticket</name>
            <value>
              <int>9999999</int>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

2.5.29. Consulta status de um comando enviado

Aplicabilidade: Consultar o status de um comando que já foi enviado. Com esse comando, pode-se consultar o status de um comando que já tenha sido enviado. Os parâmetros de consulta devem ser os que foram enviados no comando. **IMPORTANTE: Para esse comando não deve ser enviado ticket, pois se trata de uma consulta aos comandos enviados.**

Comando: status_ticket

Parâmetros: Ticketconsulta (Qual o ticket enviado que será consultado).

Ticketinterno (Retorno do servidor – Campo ticketServidor).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>status_ticket</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>login</name>
            <value><string>teste_login</string></value>
          </member>
          <member>
            <name>password</name>
            <value><string>teste_senha</string></value>
          </member>
          <member>
            <name>placa</name>
            <value><string>0000AAA</string></value>
          </member>
          <member>
            <name>ticketconsulta</name>
            <value><int>123456</int></value>
          </member>
          <member>
            <name>ticketinterno</name>
            <value><int>123456</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Exemplo de retorno do comando `status_ticket` executado com sucesso no Servidor:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>successCode</name>
            <value>
              <int>1</int>
            </value>
          </member>
          <member>
            <name>successString</name>
            <value>
              <string>Comando enviado para execucao.</string>
            </value>
          </member>
          <member>
            <name>returnServerData</name>
            <value>
              <string>0</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

OBS: Outra possível forma de retorno, é do seguinte modelo abaixo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>faultCode</name>
            <value>
              <int>2</int>
            </value>
          </member>
          <member>
            <name>faultString</name>
            <value>
              <string>Equipamento offline.</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

Michelin connected fleet Powered by Sascar

```

</member>
<member>
  <name>returnServerData</name>
  <value>
    <string>2</string>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
    
```

O parâmetro **successCode** ou **faultCode** pode retornar 1 – Sucesso e 2 – Erro.

O parâmetro **returnServerData** retorna o código referente a mensagem da execução do comando, conforme é apresentado na tabela abaixo.

CODIGO	TIPO	MENSAGEM
0	COMANDO_EXECUTADO	Comando foi enviado para execução
1	COMANDO_ARMAZENADO	Comando foi armazenado
2	COMANDO_EXPIRADO	Equipamento esta offline
3	COMANDO_ENFILEIRADO	Resposta que mostra qual comando está na fila
4	COMANDO_NAOSUPORTADO	Comando não suportado
5	COMANDO_TIMEOUT	Comando não executado por timeout
6	COMANDO_RECUSADO	Comando foi recusado
7	COMANDO_ENVIADO	Comando foi enviado para execução
8	COMANDO_SMS_ENVIADO	Comando via SMS foi enviado
9	COMANDO_ONLINE	Resposta de ping
10	COMANDO_NAOARMazenado	Não armazenado na fila
11	COMANDO_ERRO_GENERICO	Erro com descrição
12	COMANDO_GERENCIADORA	Existe algum comando enfileirado pela gerenciadora
13	COMANDO_RESPOSTA	Comando seguido de string resposta. a string depende do comando requisitado
14	COMANDO_NAOEXECUTADO	Comando não executado, Nack do Equipamento

Exemplo de retorno do comando `status_ticket` não executado no Servidor, devido a um erro:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<fault>
  <value>
    <struct>
      <member>
        <name>faultCode</name>
        <value>
          <int>1</int>
        </value>
      </member>
      <member>
        <name>faultString</name>
        <value>
          <string>Erro ao buscar status do ticket</string>
        </value>
      </member>
    </struct>
  </value>
</fault>
```

2.5.30. Vincular Alerta AVD ao Veículo

Aplicabilidade: Requisita o vínculo de um alerta AVD a uma lista de veículos. **IMPORTANTE:** Para esse comando deve ser utilizado o endereço <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Endereço: <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Comando: vincular_alerta_avd

Parâmetros: Placas: Lista de placas a serem vinculadas ao alerta AVD
id_alerta_avd: Identificador do alerta AVD a ser vinculado.

```
<?xmlversion="1.0" encoding="ISO-8859-1"?>
<comando>
<methodName>vincular_alerta_avd</methodName>
<login>teste_login</login>
<password>teste_senha</password>
<placas>
<placa>0000AAA</placa>
<placa>0000BBB</placa>
</placas>
<params>
<param>ticket</param>
<param>9999999</param>
<param>id_alerta_avd</param>
<param>11111</param>
</params>
</comando>
```

Exemplo de retorno do comando vincular_alerta_avd executado com sucesso no Servidor:

```
<?xmlversion="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>1</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>1</string>
</value>
</member>
</struct>
</value>
</param>
```

Michelin connected fleet Powered by Sascar

```
</params>
</methodResponse>
```

Exemplo de retorno do comando vincular_alerta_avd executado com erro no Servidor:

```
<?xmlversion="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>2</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>2</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Códigos de retorno do comando vincular_alerta_avd:

CODIGO	TIPO	MENSAGEM
1	COMANDO_EXECUTADO	Comando foi executado com sucesso.
2	COMANDO_RECUSADO	Comando falhou ao executar.

Mensagens de retorno do comando vincular_alerta_avd:

MENSAGEM	MENSAGEM
Veiculo nao suporta o Alerta AVD	Quando o veículo possui o suporte ao Alerta AVD.
Alerta AVD invalido	Quando o identificador do Alerta enviado não é valido ou o alerta não é AVD.
Veiculo nao pertence a gerenciadora	Quando o veículo não está vinculado ao login da gerenciadora.
Informe ou placas, ou veiculos para executar a acao	Quando é enviado ao mesmo tempo placas e veoid (identificador do veículo).
Quantidade de veiculos maior que	Quando a quantidade de placas ou veoid (identificador do veículo)

o limite permitido

enviados é superior ao permitido.

2.5.31. Desvincular Alerta AVD ao Veículo

Aplicabilidade: Requisita o desvínculo de um alerta AVD a uma lista de veículos. **IMPORTANTE:** Para esse comando deve ser utilizado o endereço <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Endereço: <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Comando: desvincular_alerta_avd

Parâmetros: Placas: Lista de placas a serem vinculadas ao alerta AVD

id_alerta_avd: Identificador do alerta AVD a ser vinculado.

```
<?xmlversion="1.0" encoding="ISO-8859-1"?>
<comando><methodName>desvincular_alerta_avd</methodName>
<login>teste_login</login>
<password>teste_senha</password>
<placas>
<placa>0000AAA</placa>
<placa>0000BBB</placa>
</placas>
<params>
<param>ticket</param>
<param>9999999</param>
<param>id_alerta_avd</param>
<param>11111</param>
</params>
</comando>
```

Exemplo de retorno do comando desvincular_alerta_avd executado com sucesso no Servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>1</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>1</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Exemplo de retorno do comando desvincular_alerta_avd executado com erro no Servidor:

Michelin connected fleet Powered by Sascar

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>2</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>2</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Códigos de retorno do comando desvincular_alerta_avd:

CODIGO	TIPO	MENSAGEM
1	COMANDO_EXECUTADO	Comando foi executado com sucesso.
2	COMANDO_RECUSADO	Comando falhou ao executar.

Mensagens de retorno do comando desvincular_alerta_avd:

MENSAGEM	MENSAGEM
Veiculo nao suporta o Alerta AVD	Quando o veículo possui o suporte ao Alerta AVD.
Alerta AVD invalido	Quando o identificador do Alerta enviado não é valido ou o alerta não é AVD.
Veiculo nao pertence a gerenciadora	Quando o veículo não está vinculado ao login da gerenciadora.
Informe ou placas, ou veiculos para executar a acao	Quando é enviado ao mesmo tempo placas e veioid (identificador do veículo).
Quantidade de veiculos maior que o limite permitido	Quando a quantidade de placas ou veioid (identificador do veículo) enviados é superior ao permitido.

2.5.32. Inicializar Operação

Aplicabilidade: Requisita o início da operação de veículos. **IMPORTANTE:** Para esse comando deve ser utilizado o endereço <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Endereço: <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Comando: inicializar_operacao

Parâmetros: Placas: Lista de placas para inciar a operação.

```
<?xmlversion="1.0"encoding="ISO-8859-1"?>
<comando>
<methodName>inicializar_operacao</methodName>
<login>teste_login</login>
<password>teste_senha</password>
<placas>
<placa>0000AAA</placa>
<placa>0000BBB</placa>
</placas>
<params>
<param>ticket</param>
<param>9999999</param>
</params>
</comando>
```

Exemplo de retorno do comando inicializar_operacao executado com sucesso no Servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>1</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>1</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Exemplo de retorno do comando inicializar_operacao executado com erro no Servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>2</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>2</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Códigos de retorno do comando inicializar_operacao:

CODIGO	TIPO	MENSAGEM
1	COMANDO_EXECUTADO	Comando foi executado com sucesso.
2	COMANDO_RECUSADO	Comando falhou ao executar.

Mensagens de retorno do comando inicializar_operacao:

MENSAGEM	MENSAGEM
Veiculo nao pertence a gerenciadora	Quando o veículo não está vinculado ao login da gerenciadora.
Informe ou placas, ou veiculos para executar a acao	Quando é enviado ao mesmo tempo placas e veioid (identificador do veículo).
Quantidade de veiculos maior que o limite permitido	Quando a quantidade de placas ou veioid (identificador do veículo) enviados é superior ao permitido.

2.5.33. Finalizar Operação

Aplicabilidade: Requisita o fim da operação de veículos. **IMPORTANTE:** Para esse comando deve ser utilizado o endereço <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Endereço: <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Comando: finalizar_operacao

Parâmetros: Placas: Lista de placas para finalizar a operação.

```
<?xmlversion="1.0"encoding="ISO-8859-1"?>
<comando>
<methodName>finalizar_operacao</methodName>
<login>teste_login</login>
<password>teste_senha</password>
<placas>
<placa>0000AAA</placa>
<placa>0000BBB</placa>
</placas>
<params>
<param>ticket</param>
<param>9999999</param>
</params>
</comando>
```

Exemplo de retorno do comando finalizar_operacao executado com sucesso no Servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>1</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>1</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Exemplo de retorno do comando finalizar_operacao executado com erro no Servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>2</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>2</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Códigos de retorno do comando finalizar_operacao:

CODIGO	TIPO	MENSAGEM
1	COMANDO_EXECUTADO	Comando foi executado com sucesso.
2	COMANDO_RECUSADO	Comando falhou ao executar.

Mensagens de retorno do comando finalizar_operacao:

MENSAGEM	MENSAGEM
Veiculo nao pertence a gerenciadora	Quando o veículo não está vinculado ao login da gerenciadora.
Informe ou placas, ou veiculos para executar a acao	Quando é enviado ao mesmo tempo placas e veioId (identificador do veículo).
Quantidade de veiculos maior que o limite permitido	Quando a quantidade de placas ou veioId (identificador do veículo) enviados é superior ao permitido.

2.5.34. Vincular Rota ao Veículo

Aplicabilidade: Requisita o vínculo da rota ao veículo. **IMPORTANTE:** Para esse comando deve ser utilizado o endereço <https://xmlrpc.sascar.com.br/xmlrpc/operacao>. O desvinculo da rota é no momento da finalização da operação.

Endereço: <https://xmlrpc.sascar.com.br/xmlrpc/operacao>.

Comando: vincular_rota

Parâmetros: Placas: Lista de placas a serem vinculadas a rota.

Rota: Identificador da rota a ser vinculada aos veículos.

Tipo: Sentido da rota: 0 – Sentido Oposto.

1 – Sentido Normal.

```
<?xmlversion="1.0"encoding="ISO-8859-1"?>
<comando>
<methodName>vincular_rota</methodName>
<login>teste_login</login>
<password>teste_senha</password>
<placas>
<placa>0000AAA</placa>
<placa>0000BBB</placa>
</placas>
<params>
<param>ticket</param>
<param>9999999</param>
<param>rota</param>
<param>11111</param>
<param>tipo</param>
<param>1</param>
</params>
</comando>
```

Exemplo de retorno do comando vincular_rota executado com sucesso no Servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>1</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>1</string>
</value>
```

```
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Exemplo de retorno do comando vincular_rota executado com erro no Servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
<params>
<param>
<value>
<struct>
<member>
<name>0000AAA</name>
<value>
<string>2</string>
</value>
</member>
<member>
<name>0000BBB</name>
<value>
<string>2</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodResponse>
```

Códigos de retorno do comando vincular_rota:

CODIGO	TIPO	MENSAGEM
1	COMANDO_EXECUTADO	Comando foi executado com sucesso.
2	COMANDO_RECUSADO	Comando falhou ao executar.

Mensagens de retorno do comando vincular_rota:

MENSAGEM	MENSAGEM
Veiculo nao pertence a gerenciadora	Quando o veículo não está vinculado ao login da gerenciadora.
Informe ou placas, ou veiculos para executar a acao	Quando é enviado ao mesmo tempo placas e veioid (identificador do veículo).
Quantidade de veiculos maior que o limite permitido	Quando a quantidade de placas ou veioid (identificador do veículo) enviados é superior ao permitido.

2.6. Suporte

Esperamos que as informações contidas neste documento sejam suficientes para a instalação do servidor de integração e desenvolvimento de sua aplicação, incluindo envio de comandos por XML-RPC.

Contudo, se surgirem dúvidas que não tenham sido abordadas neste documento, encaminhe para nossa Central de Relacionamento através do telefone 4002-6004. Sugestões para melhoria contínua de nossos serviços e elogios também podem ser direcionados para a Central de Relacionamento. Desde já agradecemos a preferência por nossa tecnologia.

2.7. Release notes Integração

Versão 2.0

- Implementação da chegada de alarmes de sequenciamento de macros.
- Implementação do embarque de sequenciamento de macros via XML-RPC.
- Implementação de tabela contendo os eventos de sequenciamento de macros.

2.8. Change log Documentação Integração

Versão 3.0

- Inclusão da tabela sequenciamento_alarme.
- Inclusão da tabela sequenciamento_evento.
- Inclusão da tabela tipo_sequenciamento_macro_td50.
- Inclusão da tabela comando_tipo contendo o nome dos comandos.
- Inclusão de comando para embarque do sequenciamento de macros via XML-RPC.
- Correção na struct de alguns comandos
- Mudança dos documentos de integração que passam a ser um único documento de instalação do servidor, estrutura de tabelas e estrutura de comandos XML-RPC.

Versão 3.1

- Inclusão do comando de Reset de alarme mantendo o perfil de segurança e Reset de Alarme limpando o perfil de segurança (sequenciamento de macros).

Versão 3.2

- Correção do nome do comando de Reset de alarme mantendo o perfil de segurança e Reset de Alarme mantendo o perfil de segurança para reset_undo_alarme.

Versão 3.3

- Mudança nos IPs para liberação, conforme tópicos 2.6 e 2.7.

Versão 3.4

- Atualização do logotipo da Sascar.
- Alterado host de envio de comandos XML-RPC no tópico 4.3.
- Remoção de coluna duplicada upacentrada8 na tabela ultimo_pacote tópico 3.24.
- Remoção dos códigos de erros XML-RPC do tópico 4.4.

Versão 4.7

- Inclusão de novos métodos para vincular, desvincular alertas AVD, inicializar operação, finalizar operação e vincular rota ao veículo.
- Inclusão de novo endpoint para envio em massa de veículos.

Michelin connected fleet Powered by Sascar

Versão 4.8

- Mudança dos dois end-points de HTTP para HTTPS com TLSv1.2
- https://xmlrpc.sascar.com.br/xmlrpc/enviar_comando
- <https://xmlrpc.sascar.com.br/xmlrpc/operacao>