



Web Service de Integração Sascar

MICHELIN Connected Fleet

2026

Sumário

1. Termos de Confidencialidade	13
2. Introdução	14
2.1. Suporte	15
2.2. URL de conexão	16
2.3. Autenticação	17
3. Métodos – Comandos e Atuadores	18
3.1. AtualizarSenha	18
3.2. ObterAlertasAVDVinculados	20
3.3. ObterGrupoAtuadores	22
3.4. ObterCadastroAlertasAVD	27
3.5. ObterStatusComando	29
3.6. ObterStatusComandoTicketSascar	31
3.7. ObterTipoComando	33
3.8. ObterSequenciamentoEvento	35
3.9. ComandoEmbarquePontoDiario	37
4. Métodos – Veículos e Clientes	39
4.1. ObterClientes	39
4.2. ObterClientesV2	41
4.3. ObterVeiculos	43
4.4. ObterVeiculosJSON	48
4.5. ObterVeiculosRFNacional	53
4.6. getVehiclesJSON	56
4.7. ObterDadosAdicionais	61
4.8. ObterDadosAdicionaisCliente	63
4.9. ObterPontosReferencia	65
4.10. ObterRotas	67
4.11. ObterEnderecoPosicao	69
5. Métodos – Posição	71
5.1. ObterPacotePosicoes	71
5.2. ObterPacotePosicoesMotorista	79
5.3. ObterPacotePosicoesMotoristaComPlaca	87
5.4. ObterPacotePosicoesMotoristaRestricao	95
5.5. ObterPacotePosicaoMotoristaPorRange	103
5.6. ObterPacotePosicaoMotoristaHistorico	111
5.7. ObterPacotePosicaoMotoristaPorRangeJSON	119
5.8. ObterPacotePosicoesMotoristaJSON	128
5.9. ObterPacotePosicoesRestricao	137
5.10. ObterPacotePosicaoPorRange	144
5.11. ObterPacoteLocalizacao	152
5.12. ObterPacotePosicaoHistorico	155
5.13. ObterPacotePosicoesJSON	163
5.14. ObterPacotePosicoesJSONComPlaca	172
5.15. ObterPacotePosicaoPorRangeJSON	181
5.16. ObterPacotePosicoesRFNacional	190
5.17. getPositionsPacketJSON	192
5.18. getDriverPositionPacketJSON	201

5.19. getPositionPacketByRangeJSON	210
5.20. getDriverPositionPacketByRangeJSON	219
5.21. getPositionPacketWithLicensePlateJSON	228
6. Métodos – Layout e Macros	236
6.1. ObterMacroTd50Tmcd	236
6.2. ObterMacroTd50TmcdDetalhado	238
6.3. ObterMascaraDispositivo	240
6.4. ObterMacroTD40	242
6.5. ObterLayout	244
6.6. ObterLayoutDetalhado	246
6.7. ObterLayoutAcaoEmbarcadaAVD	248
6.8. ObterLayoutAreaAvd	250
6.9. ObterLayoutData	252
6.10. ObterLayoutTecladoVeiculos	254
6.11. ObterLayoutGrupoPontos	256
7. Métodos – Telemetria e Câmeras	258
7.1. ObterEventosTempoDirecao	258
7.2. ObterEventosTempoDirecaoDataChegada	262
7.3. ObterDeltaTelemetriaIntegracaoInercia	266
7.4. ObterDeltaTelemetriaIntegracaoDataChegada	273
7.5. ObterDeltaTelemetriaIntegracaoInerciaDataChegada	276
7.6. ObterEventoTelemetriaIntegracaoDataChegada	282
7.7. getSmartCamerasEvents	285
7.8. SolicitarEventosCaixaPreta	289
7.9. RecuperarEventosCaixaPreta	291
8. Métodos – Motoristas	295
8.1. ObterMotoristas	295
8.2. ObterMotoristasJson	298
8.3. ObterMotoristasVeiculos	301
9. Referência – Estrutura de Eventos e Exceções	303
9.1. Estrutura Atual	303
9.2. Nova Estrutura de Eventos	305
9.3. Exemplos da Nova Estrutura de Eventos	306
9.4. Mensagens de Exceção	307
10. Anexo	308
10.1. Criação de Cliente para Conexão com o Web Service SasIntegra	308



Nova versão disponível: Este documento corresponde à versão **3.01** do Manual Técnico do Web Service SasIntegra. Consulte a versão mais recente do manual para obter informações atualizadas sobre novos métodos e funcionalidades.

REVISION

Data	Versão	Autor	Descrição
2012-06-01	1.0	Desenvolvimento de Sistemas	Criação do documento.
2012-06-15	1.1	Desenvolvimento de Sistemas	Atualização dos itens de retorno do pacote de posição.
2012-07-10	1.2	Desenvolvimento de Sistemas	Adicionado tópico sobre a nova estrutura de eventos; alterada descrição do campo de eventos do pacote de posição.
2012-07-27	1.3	Desenvolvimento de Sistemas	Alteração do meio de autenticação.
2012-08-27	1.4	Desenvolvimento de Sistemas	Alteração do envio de pacotes de posição informando os dados de endereço e pontos de referência para todos os pacotes.
2012-09-24	1.5	Desenvolvimento de Sistemas	Adicionado exemplo descritivo da nova estrutura de Eventos.
2013-01-08	1.6	Desenvolvimento de Sistemas	Inclusão de novos serviços de layouts e macros: • obterLayoutDetalhado • obterMacroTd50TmcdDetalhado
2013-02-08	1.7	Desenvolvimento de Sistemas	Alteração do método <code>obterGrupoAtuador</code> (envia apenas informações de Eventos, Entradas e Saídas). Remoção do campo <code>tipoMensagem</code> e inclusão do campo <code>jamming</code> no método <code>obterPacotePosições</code> .
2013-03-25	1.8	Desenvolvimento de Sistemas	Inclusão de novos métodos: • Consultar Motoristas • Consultar Motoristas Vinculados a Veículos • Consultar Layout de Teclado em Veículo • Consultar Layout de Grupo de Pontos Inclusão de regra que permite retornar apenas o registro desejado através de id para os métodos <code>obterClientes</code> , <code>obterVeiculos</code> e <code>obterMotoristas</code> .
2013-06-05	1.9	Desenvolvimento de Sistemas	Inclusão do Anexo A com exemplo de criação de um cliente para uso dos métodos do Web Service SasIntegra.
2013-08-01	1.10	Desenvolvimento de Sistemas	Atualização de novo campo em <code>obterMacroTd50Tmcd</code> e de itens e exemplos em métodos diversos.

Data	Versão	Autor	Descrição
2013-10-18	1.11	Desenvolvimento de Sistemas	Inclusão do novo método obterLayoutAcaoEmbarcadaAVD .
2014-01-31	1.12	Desenvolvimento de Sistemas	Alteração da URL de chamada do serviço de integração. Inclusão dos novos métodos e campos: • Identificador do pacote (idPacote) em obterPacotePosicoes() • atualizarSenha() • obterPacotePosicaoPorRange() • Campos idMacro e layout no retorno de ObterLayoutDetalhado
2014-05-21	1.13	Desenvolvimento de Sistemas	Inclusão de novos métodos de Telemetria: • obterEventosTelemetria • obterDeltasTelemetria • enviarParametrizacaoTelemetria
2014-10-23	1.14	Desenvolvimento de Sistemas	Inclusão do novo método comandoEmbarquePontoDiario .
2015-03-06	1.15	Desenvolvimento de Sistemas	Inclusão do novo método obterStatusComandoTicketSascar . Inclusão de dois novos status (Comando Cancelado e Comando Pendente) nos retornos dos métodos: • obterStatusComandoTicketSascar • obterStatusComando
2015-04-06	1.16	Desenvolvimento de Sistemas	Inclusão do novo método obterPacoteLocalizacao .
2016-01-14	1.17	Desenvolvimento de Sistemas	Inclusão do novo método obterDadosAdicionais .
2015-09-03	1.18	Desenvolvimento de Sistemas	Alteração nos itens: • ObterGrupoAtuadores • ObterSequenciamentoEvento
2016-04-22	1.19	Desenvolvimento de Sistemas	Inclusão de mensagens de Exceções.
2016-05-27	1.20	Desenvolvimento de Sistemas	Atualização do Manual da V3.
2016-07-28	1.21	Desenvolvimento de Sistemas	Atualização do Manual da V3 — Observação em Inibir Comando no método ObterStatusComando .

Data	Versão	Autor	Descrição
2016-08-03	1.22	Desenvolvimento de Sistemas	Inclusão de regra de bloqueio de clientes com restrição. Alteração nos itens: • obterPacotePosicoes • obterPacotePosicoesJSON • obterPacotePosicaoPorRange • obterPacotePosicaoPorRangeJSON • obterPacotePosicaosHistorico • obterMotorista • obterVeiculos Inclusão do novo método obterPacotePosicoesRestricao.
2016-09-14	1.23	Desenvolvimento de Sistemas	Correção do XML no método obterPacotePosicaoHistorico e da quantidade de registros em obterPacotePosicoes .
2016-10-03	1.24	Desenvolvimento de Sistemas	Inclusão dos Ids 674 e 675 (eventos da trava de 5ª roda inteligente) na lista de atuadores suportados pela Sascar.
2017-02-06	1.25	Desenvolvimento de Sistemas	Inclusão das Restrições Financeiras.
2017-02-09	1.26	Desenvolvimento de Sistemas	Atualização do manual referente aos valores de retorno do tipoTeclado .
2017-04-19	1.27	Desenvolvimento de Sistemas	Inclusão do método ObterPacotePosicaoRFNacional .
2017-07-06	1.28	Desenvolvimento de Sistemas	Inclusão do novo valor 2 (Satelital Texto Livre) para o campo satelite. Impacta os métodos: • ObterPacotePosicoes • ObterPacotePosicoesRestricao • ObterPacotePosicaoPorRange • obterPacotePosicaoHistorico • obterPacotePosicoesJSON • obterPacotePosicaoPorRangeJSON
2017-08-03	1.29	Desenvolvimento de Sistemas	Atualização do método obterPacotePosicaoHistorico .
2017-09-18	1.30	Desenvolvimento de Sistemas	Ajuste no número de consultas simultâneas.

Data	Versão	Autor	Descrição
2017-12-29	1.31	Desenvolvimento de Sistemas	Atualização no retorno de <code>ObterGrupoAtuadores</code> com inclusão de novos teclados: • SASMDT • TMCD • TD50 • TD-40 Inclusão dos campos <code>velocidadeMaximaEvento</code> e <code>velocidadeReferencia</code> no retorno de <code>obterEventoTelemetriaIntegracao</code>.
2018-02-01	1.32	Desenvolvimento de Sistemas	Criação dos novos métodos de posição com informações de motorista e limpador de para-brisa: • <code>obterPacotePosicoesMotorista</code> • <code>obterPacotePosicoesMotoristaRestricao</code> • <code>obterPacotePosicoesMotoristaJSON</code> • <code>obterPacotePosicaoMotoristaPorRangeJSON</code> • <code>obterPacotePosicaoMotoristaPorRange</code> • <code>obterPacotePosicaoMotoristaHistorico</code>
2018-05-08	1.33	Desenvolvimento de Sistemas	Inclusão das informações de ID do evento, Velocidade de referência, Velocidade máxima do evento e Duração do evento nos métodos de posição: • <code>obterPacotePosicoesMotorista</code> • <code>obterPacotePosicoesMotoristaJSON</code> • <code>ObterPacotePosicoesMotoristaRestricao</code> • <code>ObterPacotePosicaoMotoristaHistorico</code> • <code>ObterPacotePosicaoMotoristaPorRange</code> • <code>ObterPacotePosicaoMotoristaPorRangeJSON</code>
2018-05-09	1.34	Desenvolvimento de Sistemas	Criação do método <code>obterEventoTelemetriaDescricao</code>, que retorna lista com todos os Eventos disponíveis, suas descrições, IDs e Tipos de Evento.
2018-05-22	1.35	Desenvolvimento de Sistemas	Inclusão da informação de Consumo de Combustível no método <code>obterDeltaTelemetriaIntegracao</code>.
2019-04-11	1.36	Desenvolvimento de Sistemas	Atualização da informação de Velocidade e Odômetro nos métodos de posição: por padrão é enviada a do GPS; se o veículo possuir Telemetria, são enviadas as da Telemetria.

Data	Versão	Autor	Descrição
2019-07-18	1.37	Desenvolvimento de Sistemas	Inclusão de parâmetros de freada brusca no método <code>enviarParametrizacaoTelemetria</code> : pedal de freio acionado e velocidade mínima no início do evento.
2019-07-20	1.38	CSA	Alteração no método <code>ObterVeiculos</code> , retornando dados no campo <code>descricao</code> e incluído o campo <code>idEquipamentoDesc</code> .
2019-07-20	1.39	CSA	Criado o método <code>ObterVeiculosJson</code> .
2019-07-25	1.40	CSA	Correção nas descrições dos itens <code>idReferencia</code> , <code>pontoReferencia</code> , <code>anguloReferencia</code> e <code>distanciaReferencia</code> .
2019-07-06	1.41	CSA	Inclusão dos métodos de posições que retornam placa como atributo do response: • <code>obterPacotePosicoesComPlaca</code> • <code>obterPacotePosicoesMotoristaComPlaca</code> • <code>obterPacotePosicoesJSONComPlaca</code> Inclusão do método <code>obterVeiculosRFNacional</code> .
2019-09-02	1.42	Desenvolvimento de Sistemas	Inclusão do novo método <code>ObterLayoutAreaAvd</code> .
2019-10-09	1.43	Desenvolvimento de Sistemas	Inclusão do novo método <code>obterEventosTempoDirecao</code> .
2019-10-22	1.44	Desenvolvimento de Sistemas	Inclusão do novo método <code>obterLayoutData</code> .
2019-11-12	1.45	Desenvolvimento de Sistemas	Inclusão dos novos métodos em inglês: • <code>getPositionsPacketJSON</code> • <code>getDriverPositionPacketJSON</code> • <code>getPositionPacketByRangeJSON</code> • <code>getDriverPositionPacketByRangeJSON</code> • <code>getVehiclesJSON</code> • <code>getPositionPacketWithLicensePlateJSON</code>
2020-04-14	1.46	Desenvolvimento de Sistemas	Inclusão dos campos <code>aceleracaoLateralForcaG</code> , <code>direcaoForcaG</code> e <code>limiteMovLateralForcaG</code> no método <code>obterEventoTelemetriaIntegracao</code> para eventos de Força G lateral (214, 215, 216).
2020-04-30	1.47	Desenvolvimento de Sistemas	Inclusão do novo método <code>obterDadosAdicionaisCliente</code> .

Data	Versão	Autor	Descrição
2020-08-25	1.48	Desenvolvimento de Sistemas	Inclusão do campo país nos métodos JSON: • obterPacotePosicaoMotoristaPorRangeJSON • obterPacotePosicaoPorRangeJSON • obterPacotePosicoesJSON • obterPacotePosicoesJSONComPlaca • obterPacotePosicoesMotoristaJSON • getPositionsPacketJSON • getDriverPositionPacketJSON • getPositionPacketByRangeJSON • getDriverPositionPacketByRangeJSON • getPositionPacketWithLicensePlateJSON Correção da requisição e retorno, e formato de data e hora no método obterEventoTelemetriaIntegracao.
2020-09-23	1.49	Desenvolvimento de Sistemas	Inclusão do novo método obterDeltaTelemetriaIntegracaoInercia. Descontinuação do método obterDeltaTelemetriaIntegracao.
2020-09-30	1.50	Desenvolvimento de Sistemas	Inclusão dos campos nivelCombustivel/fuelLevel nos métodos: • obterPacotePosicaoMotoristaPorRangeJSON • obterPacotePosicoesMotoristaJSON • obterPacotePosicoesJSON • obterPacotePosicoesJSONComPlaca • obterPacotePosicaoPorRangeJSON • getPositionsPacketJSON • getDriverPositionPacketJSON • getPositionPacketByRangeJSON • getDriverPositionPacketByRangeJSON • getPositionPacketWithLicensePlateJSON
2020-10-26	1.51	CSA	Atualização do período de dados disponível para consumo: alterado de 48 horas para D0 e D-1.
2020-11-11	1.52	Desenvolvimento de Sistemas	Inclusão de novos parâmetros no método obterDeltaTelemetriaIntegracaoInercia.

Data	Versão	Autor	Descrição
2021-01-02	1.53	Desenvolvimento de Sistemas	Inclusão do campo litrometro/lithometer (litros, 3 casas decimais) nos métodos JSON: • obterPacotePosicaoMotoristaPorRangeJSON • obterPacotePosicoesMotoristaJSON • obterPacotePosicoesJSON • obterPacotePosicoesJSONComPlaca • obterPacotePosicaoPorRangeJSON • getPositionsPacketJSON • getDriverPositionPacketJSON • getPositionPacketByRangeJSON • getDriverPositionPacketByRangeJSON • getPositionPacketWithLicensePlateJSON
2021-03-29	1.54	Desenvolvimento de Sistemas	Criação dos métodos: • ObterRotas • ObterAlertasAVDVinculados • ObterCadastroAlertasAVD
2021-05-11	1.55	Desenvolvimento de Sistemas	Inclusão de novos campos no método obterDeltaTelemetriaIntegracaoInercia e novos eventos no método obterEventoTelemetriaIntegracao .
2021-06-17	1.56	Desenvolvimento de Sistemas	Criação do método getSmartCamerasEvents .
2021-07-08	1.57	Desenvolvimento de Sistemas	Inclusão de novos campos e eventos no método obterEventoTelemetriaIntegracao .
2021-08-27	1.58	Desenvolvimento de Sistemas	Inclusão de novos Eventos no método obterGrupoAtuadores .
2021-11-16	1.59	CSA	Mudança de HTTP para HTTPS com TLSv1.2 (vide itens "3.1" e "Anexo").

Data	Versão	Autor	Descrição
2022-01-03	1.60	CSA	Atualização do Manual sem inclusão de novos métodos ou campos. Removido registro duplicado de "ObterPacotePosicoesMotorista". Movido o item "Suporte" para o início do manual. Complemento à revisão 1.28: comportamento do campo satelite com valor 2 (Satelital Texto Livre) nos métodos: • ObterPacotePosicoes • ObterPacotePosicoesMotorista • ObterPacotePosicoesMotoristaComPlaca • ObterPacotePosicoesMotoristaRestricao • ObterPacotePosicaoMotoristaPorRange • obterPacotePosicaoMotoristaHistorico • obterPacotePosicaoMotoristaPorRangeJSON • obterPacotePosicoesMotoristaJSON • ObterPacotePosicoesRestricao • ObterPacotePosicaoPorRange • obterPacotePosicaoHistorico • obterPacotePosicoesJSON • obterPacotePosicoesJSONComPlaca • obterPacotePosicaoPorRangeJSON • getPositionsPacketJSON • getDriverPositionPacketJSON • getPositionPacketByRangeJSON • getDriverPositionPacketByRangeJSON • getPositionPacketWithLicensePlateJSON
2022-03-21	1.61	Desenvolvimento de Sistemas	Inclusão dos novos métodos: • obterDeltaTelemetriaIntegracaoDataChegada • obterDeltaTelemetriaIntegracaoInerciaDataChegada • obterEventoTelemetriaIntegracaoDataChegada
2022-06-18	1.62	Desenvolvimento de Sistemas	Inclusão do novo método obterEventosTempoDirecaoDataChegada.

Data	Versão	Autor	Descrição
2022-08-18	1.63	Desenvolvimento de Sistemas	Atualização do Manual com inclusão do campo statusAncora nos métodos: • obterPacotePosicaoMotoristaPorRangeJSON • obterPacotePosicaoPorRangeJSON • obterPacotePosicoesJSON • obterPacotePosicoesJSONComPlaca • obterPacotePosicoesMotoristaJSON O método obterSequenciamentoEvento passará a retornar as informações do Ancora.
2024-07-25	1.64	Engenharia	Inclusão da informação de Envio de Horímetro = 0 quando o método de envio for satelital para a tecnologia MSC de rastreadores.
2024-07-25	1.65	Desenvolvimento de Sistemas	Ajustes de informações dos métodos: • ObterLayoutAreaAvd • obterEventosTempoDirecaoDataChegada • ObterEventosTempoDirecao
2024-10-14	1.66	Desenvolvimento de Sistemas	Inclusão do campo acessórios nos métodos: • obterPacotePosicoesJSON • obterPacotePosicoesJSONComPlaca • obterPacotePosicoesMotoristaJSON • obterPacotePosicaoMotoristaPorRangeJSON • obterPacotePosicaoPorRangeJSON • getPositionsPacket.JSON • getPositionPacketByRangeJSON • getDriverPositionPacketByRangeJSON • getPositionPacketWithLicensePlateJSON • getDriverPositionPacketJSON
2025-01-14	2.05	Desenvolvimento de Sistemas	Adaptação do layout ao padrão institucional. Adequação à LGPD.
2026-04-07	2.06	Desenvolvimento de Sistemas	Inclusão do novo método para suporte a CNPJ alfanumérico.
2026-05-05	2.07	Desenvolvimento de Sistemas	Alteração do método ObterClientes para manter compatibilidade do protocolo e funcionamento em virtude do novo CNPJ alfanumérico.

Data	Versão	Autor	Descrição
2026-08-20	3.00	Desenvolvimento de Sistemas	Inclusão do novo método <code>obterMotoristasJson</code> : • Retorno equivalente ao método <code>obterMotoristas</code> em formato JSON. • Inclusão dos campos <code>ativo</code> e <code>dataInativacao</code>. • O método <code>obterMotoristas</code> permanece sem alterações.
2026-09-28	3.01	Desenvolvimento de Sistemas	O campo <code>acessórios</code> foi adicionado ao exemplo de resposta nos métodos: • <code>obterPacotePosicoesJSON</code> • <code>obterPacotePosicoesJSONComPlaca</code> • <code>obterPacotePosicoesMotoristaJSON</code> • <code>obterPacotePosicaoMotoristaPorRangeJSON</code> • <code>obterPacotePosicaoPorRangeJSON</code> • <code>getPositionPacketJSON</code> • <code>getDriverPositionPacketJSON</code>

1. Termos de Confidencialidade

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 can not be shared with others 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.

2. Introdução

Este documento tem por finalidade documentar a utilização do Web Service de Integração Sascar. As informações aqui apresentadas, serão úteis durante o desenvolvimento de aplicações que utilizem o Web Service Integração como fonte de informações. A integração de dados é a disponibilização de informações de rastreamento dos equipamentos instalados em veículos. Todos os dados, como mensagens, status de sensores, atuadores, informações de posicionamento, estarão disponíveis para consumo através do Web Service. As informações deste manual são exclusivamente de cunho técnico. Para compreensão do material descrito a seguir, é necessário que o leitor possua conhecimentos sobre Web Service, comunicação de rede TCP/IP, protocolo SOAP e XML. A SASCAR se limita a disponibilizar as informações em D0 e D-1, sendo o INTEGRADOR o responsável pelo consumo dessas informações dentro do prazo estipulado, desta forma, a Sascar não se responsabiliza em 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, além de ser o responsável pela guarda das informações dos clientes a ele direcionados.

2.1. Suporte

Esperamos que as informações contidas neste documento sejam suficientes para a utilização do Web Service de Integração Sascar. Contudo, se surgirem dúvidas que não tenham sido abordadas neste documento, encaminhe para nossa central de relacionamento através dos telefones 4002-6004 (Capitais e Regiões Metropolitanas) ou 0800-648-6004 (Demais Regiões) selecionando a opção 2 da URA. Sugestões para melhoria contínua de nossos serviços e elogios também podem ser direcionados para a central de relacionamento. Desde já agradecemos pela preferência por nossa tecnologia.



Este manual está em constante atualização, sempre verifique no site da SASCAR se você está usando a última versão.

2.2. URL de conexão

URL SasIntegra

<https://sasintegra.sascar.com.br/SasIntegra/SasIntegraWSService?wsdl>



A Sascar, prezando a segurança nas informações de seus clientes, está descontinuando o protocolo HTTP para o serviço “SASINTEGRA.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.

2.3. Autenticação

A autenticação com o Web Service será realizada através do envio do usuário e senha do integrador como parâmetros dos métodos que forem consumidos.

Consultas Simultâneas

O Web Service de Integração possui um controle sobre a quantidade de consultas simultâneas que poderão ser realizadas.



O limite atual é de **1 solicitação simultânea por integradora**. As consultas que excederem esse limite serão recusadas pelo Web Service.

Os métodos sujeitos a essa restrição são:

- `obterPacotePosicoes`
- `obterPacotePosicoesRestricao`
- `obterPacotePosicoesRFNacional`
- `obterPacotePosicoesJSON`
- `obterPacotePosicaoPorRangeJSON`
- `obterPacotePosicaoPorRange`
- `obterPacotePosicaoHistorico`
- `obterPacotePosicoesMotorista`
- `obterPacotePosicoesMotoristaRestricao`
- `obterPacotePosicoesMotoristaJSON`
- `obterPacotePosicaoMotoristaPorRangeJSON`
- `obterPacotePosicaoMotoristaPorRange`
- `obterPacotePosicaoMotoristaHistorico`
- `getPositionPacketJSON`
- `getDriverPositionPacketJSON`
- `getPositionPacketByRangeJSON`
- `getDriverPositionPacketByRangeJSON`
- `getPositionPacketWithLicensePlateJSON`

Para os demais métodos não há limite de requisições simultâneas. O Web Service permite realizar uma consulta por gerenciadora, podendo retornar até 3.000 pacotes.

3. Métodos – Comandos e Atuadores

3.1. AtualizarSenha

Descrição

Método que possibilita ao integrador alterar ou atualizar sua senha de acesso aos métodos de consumo da integração web service.

Declaração do Método

```
String atualizarSenha(String usuario, String senhaAtual, String novaSenha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senhaAtual	String	Sim	Senha atual do integrador para acesso ao sistema.
novaSenha	String	Sim	Nova senha a substituir a senha atual.

Retorno

Caso true Retorna uma mensagem informando que a senha foi alterada: **"Login Atualizado com Sucesso! Agora o webservice já está pronto para utilização."**

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:atualizarSenha>
      <usuario>Usuário</usuario>
      <senhaAtual>ValorSenhaAtual</senhaAtual>
      <novaSenha>ValorSenhaNova</novaSenha>
    </web:atualizarSenha>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:atualizarSenhaResponse xmlns:ns2="http://webservice.web.integracao.sascar.com.br/">
      <return>
        Login Atualizado com Sucesso! Agora o webservice já esta pronto para utilizacao!
      </return>
    </ns2:atualizarSenhaResponse>
  </S:Body>
</S:Envelope>
```

```
</return>  
</ns2:atualizarSenhaResponse>  
</S:Body>  
</S:Envelope>
```


3.2. ObterAlertasAVDVinculados

Descrição

Método que possibilita ao integrador consultar quais alertas AVD estão vinculados aos veículos.

Declaração do Método

```
String obterAlertasAVDVinculados(String usuario, String senha, String veiplaca, String veioId)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuário	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha Atual do integrador para acesso ao sistema
veiplaca	String	Sim	Placa para consulta
veioId	String	Sim	Identificação da placa de consulta

Retorno

Caso true Retorna uma lista de objetos do tipo “ObterAlertasAVDVinculados” com as propriedades (Acoes, idAlerta, Login, nomeAlerta) - como tabela abaixo:

Nome	Tipo	Descrição
acoes	String	Lista de ações associadas ao alerta
idAlerta	Integer	Id do alerta vinculado
login	String	Usuário responsável pela associação do alerta ao veículo
nomeAlerta	String	Descrição do alerta associado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:ObterAlertasAVDVinculados>
      <usuario>Usuário</usuario>
      <senha>ValorSenha</senha>
      <!--Optional:-->
      <veiplaca>ValorVeiculoPlaca</veiplaca>
      <!--Optional:-->
      <veioId>ValorVeiculoID</veioId>
    </web:ObterAlertasAVDVinculados>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</web:ObterAlertasAVDVinculados>  
</soapenv:Body>  
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">  
  <S:Body>  
    <ns0:obterAlertasAVDVinculadosResponse  
      xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">  
      <return>  
        <acoes>513 I oi</acoes>  
        <idAlerta>36031</idAlerta>  
        <login>ADM</login>  
        <nomeAlerta>517 E 1 </nomeAlerta>  
      </return>  
    </ns0:obterAlertasAVDVinculadosResponse>  
  </S:Body>  
</S:Envelope>
```

3.3. ObterGrupoAtuadores

Descrição

Método para consulta da relação de sensores, atuadores e eventos disponibilizados pela Sascar.

Declaração do Método

```
List obterGrupoAtuadores(String usuario, String senha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuário	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema

Retorno

Retorna uma lista de objetos do tipo 'GrupoAtuador' com as propriedades abaixo:

Nome	Tipo	Descrição
idAtuador	Integer	Código do atuador.
descricao	String	Nome do atuador, sensor, acessório ou evento.
tipoPorta	String	Tipo do atuador: • S — Saída • E — Entrada • V — Evento



As informações obtidas podem ser utilizadas para alimentar a tabela 'atuador_grupo' da base de dados de integração.

Abaixo segue a lista de dados enviados pelo método em questão, os quais são utilizados para sinalizar a ocorrência de eventos no método ObterPacotePosicoes.

Código	Descrição	Tipo	Observação
251	Sensor Bau Traseiro	Entrada	

Código	Descrição	Tipo	Observação
541	Falha Teclado TD50	Entrada	
249	Sensor Bau	Entrada	
246	Sensor Portas	Entrada	
250	Sensor Bau Lateral	Entrada	
241	Desengate	Entrada	
218	Sensor Porta Intermediária	Entrada	
217	Sensor Limpador	Entrada	
216	Sensor Lanterna	Entrada	
215	Sensor Cofre de Motor	Entrada	
227	Alimentação Carreta	Entrada	
214	Sensor de Betoneira	Entrada	
206	Sensor Tampa de Combustível	Entrada	
207	Sensor Janelas	Entrada	
229	Violação Sascarreta	Entrada	
517	Pânico	Entrada	Indica ocorrência de botão de pânico.

Código	Descrição	Tipo	Observação
248	Sensor Porta Carona	Entrada	
231	Violação Painel	Entrada	
247	Sensor Porta Motorista	Entrada	
232	Buzzer	Saída	
252	Trava Bau	Saída	
253	Trava Bau Lateral	Saída	
254	Trava Bau Traseiro	Saída	
211	Buzzer - Segunda Instalação	Saída	
245	Trava 5 Roda	Saída	
240	Sirene	Saída	
659	Pre Sleep	Evento	
658	Ancora	Evento	Indica entrada ou saída de cerca.
509	Excesso de Tempo Parado	Evento	
1002	Pontos de referência	Evento	
563	Excesso Tempo Parado	Evento	
562	Motorista Coação	Evento	
555	Grupo de Pontos	Evento	Indica ocorrência de entrada ou saída de um grupo de pontos.
553	Area	Evento	Indica entrada ou saída de área.
652	Chegada Ponto	Evento	

Código	Descrição	Tipo	Observação
653	Saida Ponto	Evento	
662	Prorrogação de excesso de tempo parado	Evento	
660	Senha Motorista	Evento	
674	Falha na ativação automática da Trava 5ª Roda	Evento	Indica que houve uma falha ao tentar reativar (Travar) a Trava de 5ª Roda. Obs.: Evento enviado somente pela Trava de 5ª Roda Inteligente.
675	Violação 5ª Roda	Evento	Indica que houve uma tentativa de violação da Trava de 5ª Roda. Obs.: Evento enviado somente pela Trava de 5ª Roda inteligente.
238	Teclado TMCD	Evento	Indica o tipo do teclado.
224	Teclado TD50	Evento	Indica o tipo do teclado.
550	Teclado TD50	Evento	Indica o tipo do teclado.
202	Teclado SasMDT	Evento	Indica o tipo do teclado.
235	Teclado TD-40	Evento	Indica o tipo do teclado.
200	Teclado SasMDT	Evento	Indica o tipo de teclado.
204	Sensor de umidade e temperatura	Entrada	

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ws="http://ws.integra.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
```

```
<ws:obterGrupoAtuadores>
  <usuario>teste</usuario>
  <senha>teste</senha>
</ws:obterGrupoAtuadores>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:obterGrupoAtuadoresResponse xmlns:ns2="http://ws.integra.sascar.com.br/">
      <return>
        <descricao>Panico Desativado</descricao>
        <idAtuador>517</idAtuador>
        <tipoPorta>E</tipoPorta>
      </return>
    </ns2:obterGrupoAtuadoresResponse>
  </S:Body>
</S:Envelope>
```

3.4. ObterCadastroAlertasAVD

Descrição

Método de consulta que retorna o cadastro de alertas AVD dos veículos.

Declaração do Método

```
String obterCadastroAlertasAVD(String usuario, String senha, String dataInicio)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema
dataInicio	String	Não	Data de inicio para consulta

Retorno

Retorna uma lista de objetos do tipo “ObterCadastroAlertasAVD” com as propriedades abaixo:

Nome	Tipo	Descrição
Evento	String	Descrição do alerta vinculado.
Id	String	Id do alerta vinculado.
Login	String	Login responsável pela associação do alerta ao veículo.
Timezone	String	Fuso horário do alerta

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterCadastroAlertasAvd>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <dataInicio>2021-10-01 00:00:00</dataInicio>
    </web:obterCadastroAlertasAvd>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:


```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterCadastroAlertasAvdResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <evento>254 0 1</evento>
        <id>62228</id>
        <login>ADM</login>
        <timezone>America/Sao_Paulo</timezone>
      </return>
    </ns0:obterCadastroAlertasAvdResponse>
  </S:Body>
</S:Envelope>
```

3.5. ObterStatusComando

Descrição

Método para consultar o status dos comandos enviados através do Web Service XML-RPC.

Declaração do Método

```
StatusComando obterStatusComando(String usuario, String senha, Integer ticket)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
ticket	Integer	Sim	Número do ticket "EXTERNO" do comando enviado. Este ticket é informado pela integradora na hora do envio do comando.

Retorno

Retorna uma lista de objetos do tipo 'StatusComando' com as propriedades abaixo:

Nome	Tipo	Descrição
idStatusComando	Integer	Id do status do comando. Esse id é o mesmo cadastrado na tabela 'status_comando' onde estão os possíveis status de um comando: • 1 — Comando expirado • 2 — Erro geral • 100 — Execução do comando via GPRS • 101 — Execução do comando via SATELITAL • 102 — Comando cancelado • 103 — Comando pendente
dataExec	Calendar	Data de execução/cancelamento/expiração do comando.



As informações obtidas podem ser utilizadas para alimentar a tabela 'comando' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterStatusComando>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <ticket>NumeroTicket</ticket>
    </web:obterStatusComando>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterStatusComandoResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <dataExec>2023-11-06T11:35:23.0</dataExec>
        <idStatusComando>100</idStatusComando>
      </return>
      <return>
        <dataExec>2023-11-06T11:34:09.0</dataExec>
        <idStatusComando>100</idStatusComando>
      </return>
      <return>
        <dataExec>2023-11-06T11:33:53.0</dataExec>
        <idStatusComando>100</idStatusComando>
      </return>
      <return>
        <dataExec>2023-11-06T11:33:27.0</dataExec>
        <idStatusComando>100</idStatusComando>
      </return>
      <return>
        <dataExec>2023-11-06T11:21:34.0</dataExec>
        <idStatusComando>100</idStatusComando>
      </return>
    </ns0:obterStatusComandoResponse>
  </S:Body>
</S:Envelope>
```



Quando enviado o comando XML ***inibir_sensor*** não irá retornar o ***idStatusComando***, visto que o comando não vai para o servidor de comandos, já que se trata de um status de análise de quando a posição do equipamento chega no BD. Esse comando não é enviado para o equipamento.

3.6. ObterStatusComandoTicketSascar

Descrição

Método para consultar o status dos comandos enviados através do Web Service XML-RPC.

Declaração do Método

```
StatusComando obterStatusComando(String usuario, String senha, Integer ticket)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
ticket	Integer	Sim	Número do ticket "INTERNO" do comando enviado. Este ticket é gerado e informado pela Sascar no momento do envio de um comando.

Retorno

Retorna uma lista de objetos do tipo 'StatusComando' com as propriedades abaixo:

Nome	Tipo	Descrição
idStatusComando	Integer	Id do status do comando. Esse id é o mesmo cadastrado na tabela 'status_comando' onde estão os possíveis status de um comando: • 1 — Comando expirado • 2 — Erro geral • 100 — Execução do comando via GPRS • 101 — Execução do comando via SATELITAL • 102 — Comando cancelado • 103 — Comando pendente
dataExec	Calendar	Data de execução/cancelamento/expiração do comando.



As informações obtidas podem ser utilizadas para alimentar a tabela 'comando' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterStatusComandoTicketSascar>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <ticket>Ticket</ticket>
    </web:obterStatusComandoTicketSascar>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterStatusComandoTicketSascarResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      </S:Body>
    </S:Envelope>
```

3.7. ObterTipoComando

Descrição

Método para consultar a relação de comandos que o INTEGRADOR pode enviar via XMLRPC para os seus veículos.

Declaração do Método

```
List obterTipoComando(String usuario, String senha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema

Retorno

Retorna uma lista de objetos do tipo 'TipoComando' com as propriedades abaixo:

Nome	Tipo	Descrição
idTipoComando	Integer	Código do comando.
nome	String	Nome do comando.
descricao	String	Descrição resumida do comando.



As informações obtidas podem ser utilizadas para alimentar a tabela 'comando_tipo' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterTipoComando>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
    </web:obterTipoComando>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterTipoComandoResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <descricao>Bloqueio de Veículo</descricao>
        <idTipoComando>2</idTipoComando>
        <nome>Bloqueio</nome>
      </return>
    </ns0:obterTipoComandoResponse>
  </S:Body>
</S:Envelope>
```

3.8. ObterSequenciamentoEvento

Descrição

Método para consultar a relação dos possíveis eventos de sequenciamento.

Declaração do Método

```
List obterSequenciamentoEvento(String usuario, String senha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do usuário para acesso ao sistema.
senha	String	Sim	Senha do usuário para acesso ao sistema

Retorno

Retorna uma lista de objetos do tipo 'SequenciamentoEvento' com as propriedades abaixo:

Nome	Tipo	Descrição
idSequenciamentoEvento	Integer	Identificação do evento de sequenciamento de macros.
atuador	Integer	Atuador que se refere (quando aplicável).
descricao	String	Descrição do evento de sequenciamento de macros



As informações obtidas podem ser utilizadas para alimentar a tabela 'sequenciamento_evento' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterSequenciamentoEvento>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
    </web:obterSequenciamentoEvento>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:


```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterSequenciamentoEventoResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <atuador>248</atuador>
        <descricao>Sensor Porta Carona</descricao>
        <idSequenciamentoEvento>2031</idSequenciamentoEvento>
      </return>
    </ns0:obterSequenciamentoEventoResponse>
  </S:Body>
</S:Envelope>
```

3.9. ComandoEmbarquePontoDiario

Descrição

Método utilizado para cadastrar pontos de referencia, vincula-los com um layout de grupo de pontos e em seguida, este novo layout é embarcado no veículo.

Declaração do Método

```
LogComando comandoEmbarquePontoDiario(String usuario, String senha, Integer idVeiculo, String pontosRef)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idVeiculo	Integer	Sim	Id do veículo que será enviado o comando.
pontosRef	String	Sim	Pontos de referência a cadastrar. Ver formato abaixo.



Formato do campo pontosRef

NOME DO PONTO|LAT|LONG|RAIO|GRUPO;NOME DO PONTO 2|LAT|LONG|RAIO|GRUPO



Grupo deve ser 1 ou 2.

- Grupo 1 = PONTO PARADA CLIENTE.
- Grupo 2 = PONTO DE CHECAGEM.

Retorno

Retorna um objeto do tipo 'LogComando' com as propriedades abaixo:

Nome	Tipo	Descrição
mensagem	String	Mensagem de retorno do comando.
codigo	String	Identificador (Ticket de consulta) do comando enviado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/" >
  <soapenv:Header/>
  <soapenv:Body>
    <web:comandoEmbarquePontoDiario>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <idVeiculo>IDVeículo</idVeiculo>
      <pontosRef> nomeDoPonto | Latitude | Longitude | Raio |1 ou 2</pontosRef>
    </web:comandoEmbarquePontoDiario>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:comandoEmbarquePontoDiarioResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/" >
      <return>
        <codigo>2222222</codigo>
        <mensagem>Comando enviado para execucao!</mensagem>
      </return>
    </ns0:comandoEmbarquePontoDiarioResponse>
  </S:Body>
</S:Envelope>
```

4. Métodos – Veículos e Clientes

4.1. ObterClientes

Descrição

Método para consulta dos dados cadastrais de clientes pertencentes ao INTEGRADOR.

Declaração do Método

```
List obterClientes(String usuario, String senha, Integer quantidade, Integer idCliente)
```

Foi implementado uma adequação com o objetivo manter o método obterClientes funcionando para que os clientes que o utilizam não tenham problemas, e caso precisem de CNPJs alfanuméricos deverão utilizar o obterClientesV2.

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Quantidade máxima de registros da consulta, sendo o limite de 1000 registros. Para obter as informações pertinentes apenas à um registro de cliente, preencher o campo quantidade com o valor 1 (um) e indique o id do cliente desejado no campo idCliente.
idCliente	Integer	Não	O Limite máximo de resultados do método é 1000 registros. Neste caso para obter o restante dos registros, foi implementado um sistema de paginação, onde o INTEGRADOR deve seguir as seguintes regras: Para obter a primeira página com os 1000 registros iniciais enviar 0 (zero) como parâmetro; Para obter a próxima página, enviar como parâmetro o último id do cliente da última página consultada. Para obter as informações pertinentes apenas à um registro de cliente, enviar apenas o id do cliente desejado e o no campo quantidade o valor 1 (um).

Retorno

Retorna uma lista de objetos do tipo 'Cliente' com as propriedades abaixo:

Nome	Tipo	Descrição
idCliente	Integer	Código do cliente.

Nome	Tipo	Descrição
nome	String	Nome do cliente.
CPF	Long	CPF para clientes tipo pessoa física.
CNPJ	Long	CNPJ para clientes tipo pessoa jurídica, retorna CNPJ somente numérico. Caso mostre valor 0, use o método obterClientesV2, para obter o CNPJ em formato alfanumérico.



As informações obtidas podem ser utilizadas para alimentar a tabela 'cliente' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterClientes>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>ValorNumérico</quantidade>
      <!--Optional:-->
      <idCliente>idCliente</idCliente>
    </web:obterClientes>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterClientesResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <cnpj>CNPJVALOR</cnpj>
        <cpf> CPFVALOR </cpf>
        <idCliente> IDCLIENTE </idCliente>
        <nome>NOMECLIENTE</nome>
      </return>
    </ns0:obterClientesResponse>
  </S:Body>
</S:Envelope>
```

4.2. ObterClientesV2

Descrição

Novo método para consulta dos dados cadastrais de clientes, com suporte aprimorado para CNPJ alfanumérico. Verifica dinamicamente a existência da coluna na base de dados. Se presente e preenchida, prioriza esse campo para CNPJ; caso contrário, usa o campo convertido para string. Mantém compatibilidade com bases legadas e retorna CNPJ como string.

Declaração do Método

```
List obterClientesV2(String usuario, String senha, Integer quantidade, Integer idCliente)
```

Parâmetros

Mesmos parâmetros do método obterClientes (ver seção 4.5).

Retorno

Mesma estrutura de retorno do método obterClientes, com CNPJ sempre como string (suportando numérico e alfanumérico).



Recomendado para integrações que necessitam de CNPJ alfanumérico. O método obterClientes permanece disponível para compatibilidade.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterClientesV2>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>ValorNumérico</quantidade>
      <!--Optional:-->
      <idCliente>idCliente</idCliente>
    </web:obterClientesV2>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
```

```
</S:Header>
<S:Body>
  <ns0:obterClientesV2Response xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
</S:Body>
</S:Envelope>
```

4.3. ObterVeiculos

Descrição

Método para consultar as informações relativas aos veículos de cada INTEGRADOR.



Caso cliente esteja com restrição, não será possível consultar veículos do mesmo.

Declaração do Método

```
List obterVeiculos(String usuario, String senha, Integer quantidade, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Sim	Quantidade máxima de registros da consulta, sendo o limite de 1000 registros incluindo os que possuem restrição, ou seja, pode ser retornado 940 veículos e o restante que seria 60 veículos teriam alguma restrição. Para obter as informações pertinentes apenas à um registro de veículo, preencha o campo quantidade com o valor 1 (um) e indique o id do veículo desejado no campo idVeiculo.
idVeiculo	Integer	Não	O Limite máximo de resultados do método é 1000 registros. Neste caso para obter o restante dos registros, foi implementado um sistema de paginação, onde o INTEGRADOR deve seguir as seguintes regras: Para obter a primeira página, enviar 0 (zero) como parâmetro; Para obter a próxima página enviar como parâmetro o último id do veículo da última página consultada. Repetir esse processo até que não seja retornado mais nenhum veículo, pois o retorno "vazio" indica o fim da paginação dos veículos. Para obter as informações pertinentes apenas à um registro de veículo, enviar apenas o id do veículo desejado e o no campo quantidade o valor 1 (um).

Retorno

Retorna uma lista de objetos do tipo 'Veiculo' com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Identificação do veículo.
placa	String	Placa do veículo.

Nome	Tipo	Descrição
idCliente	Integer	Identificação do cliente a qual pertence o veículo.
descricao	String	Descrição do veículo.
idEquipamento	String	Identificação do equipamento instalado no veículo.
idSensor1	Integer	Código do sensor instalado na entrada 1 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
idSensor2	Integer	Código do sensor instalado na entrada 2 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
idSensor3	Integer	Código do sensor instalado na entrada 3 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
idSensor4	Integer	Código do sensor instalado na entrada 4 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
idAtuador1	Integer	Código do atuador instalado na saída 1 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
idAtuador2	Integer	Código do atuador instalado na saída 2 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
idAtuador3	Integer	Código do atuador instalado na saída 3 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
idAtuador4	Integer	Código do atuador instalado na saída 4 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
portaPanico	Integer	Identificação da porta onde está instalado o botão de pânico
portaBloqueio	Integer	Identificação da porta onde está instalado o botão de bloqueio
serial0	Integer	Código do periférico instalado no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de periféricos disponíveis para instalação nos veículos.

Nome	Tipo	Descrição
<code>serial1</code>	Integer	Código do periférico instalado no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de periféricos disponíveis para instalação nos veículos.
<code>satelital</code>	boolean	Propriedade que indica se o veículo possui antena satelital: <code>false</code> — Não possui antena satelital <code>true</code> — Possui antena satelital
<code>idSensor5</code>	Integer	Código do sensor instalado na entrada 5 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>idSensor6</code>	Integer	Código do sensor instalado na entrada 6 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>idSensor7</code>	Integer	Código do sensor instalado na entrada 7 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>idSensor8</code>	Integer	Código do sensor instalado na entrada 8 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>idAtuador5</code>	Integer	Código do atuador instalado na saída 5 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>idAtuador6</code>	Integer	Código do atuador instalado na saída 6 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>idAtuador7</code>	Integer	Código do atuador instalado na saída 7 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>idAtuador8</code>	Integer	Código do atuador instalado na saída 8 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>idEquipamentoDesc</code>	String	Descrição do ID equipamento (MTC700, MTC600, MTC550, MXT140A,.)



As informações obtidas podem ser utilizadas para alimentar a tabela 'veiculo' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterVeiculos>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>Quantidade</quantidade>
      <!--Optional:-->
      <idVeiculo>idVeiculo</idVeiculo>
    </web:obterVeiculos>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterVeiculosResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <descricao>GSM/GPS</descricao>
        <idAtuador1>0</idAtuador1>
        <idAtuador2>0</idAtuador2>
        <idAtuador3>0</idAtuador3>
        <idAtuador4>0</idAtuador4>
        <idAtuador5>0</idAtuador5>
        <idAtuador6>0</idAtuador6>
        <idAtuador7>0</idAtuador7>
        <idAtuador8>0</idAtuador8>
        <idCliente>114208</idCliente>
        <idEquipamento>75807</idEquipamento>
        <idEquipamentoDesc>MTC400</idEquipamentoDesc>
        <idSensor1>0</idSensor1>
        <idSensor2>0</idSensor2>
        <idSensor3>0</idSensor3>
        <idSensor4>0</idSensor4>
        <idSensor5>0</idSensor5>
        <idSensor6>0</idSensor6>
        <idSensor7>0</idSensor7>
        <idSensor8>0</idSensor8>
        <idSerial0>237</idSerial0>
        <idSerial1>0</idSerial1>
        <idVeiculo>53106</idVeiculo>
        <placa>DMH8074</placa>
        <portaBloqueio>1</portaBloqueio>
        <portaPanico>0</portaPanico>
        <satelital>false</satelital>
        <telemetria>false</telemetria>
      </return>
    </ns0:obterVeiculosResponse>
  </S:Body>
</S:Envelope>
```

4.4. ObterVeiculosJSON

Descrição

Método para consultar as informações relativas aos veículos de cada INTEGRADOR. Seu retorno é em forma de Json.



Caso cliente esteja com restrição, não será possível consultar veículos do mesmo.

Declaração do Método

```
List obterVeiculos(String usuario, String senha, Integer quantidade, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Sim	Quantidade máxima de registros da consulta, sendo o limite de 1000 registros incluindo os que possuem restrição, ou seja, pode ser retornado 940 veículos e o restante que seria 60 veículos teriam alguma restrição.; Para obter as informações pertinentes apenas à um registro de veículo, preencha o campo quantidade com o valor 1 (um) e indique o id do veículo desejado no campo idVeiculo.
idVeiculo	Integer	Não	O Limite máximo de resultados do método é 1000 registros. Neste caso para obter o restante dos registros, foi implementado um sistema de paginação, onde o INTEGRADOR deve seguir as seguintes regras: Para obter a primeira página, enviar 0 (zero) como parâmetro; Para obter a próxima página enviar como parâmetro o último id do veículo da última página consultada. Repetir esse processo até que não seja retornado mais nenhum veículo, pois o retorno "vazio" indica o fim da paginação dos veículos.; Para obter as informações pertinentes apenas à um registro de veículo, enviar apenas o id do veículo desejado e no campo quantidade o valor 1 (um).

Retorno

Retorna uma lista de objetos do tipo 'Veiculo' com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Identificação do veículo.

Nome	Tipo	Descrição
placa	String	Placa do veículo.
idCliente	Integer	Identificação do cliente a qual pertence o veículo.
descricao	String	Descrição da classe do veículo. Ex: SASTM FULL , SASTM COM TELEMETRIA , SASCARGA FULL SAT 200 , SASCARGA FULL SAT I .
idEquipamento	String	Identificação do equipamento instalado no veículo.
idSensor1	Integer	Código do sensor instalado na entrada 1 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
idSensor2	Integer	Código do sensor instalado na entrada 2 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
idSensor3	Integer	Código do sensor instalado na entrada 3 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
idSensor4	Integer	Código do sensor instalado na entrada 4 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
idAtuador1	Integer	Código do atuador instalado na saída 1 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
idAtuador2	Integer	Código do atuador instalado na saída 2 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
idAtuador3	Integer	Código do atuador instalado na saída 3 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
idAtuador4	Integer	Código do atuador instalado na saída 4 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
portaPanico	Integer	Identificação da porta onde está instalado o botão de pânico
portaBloqueio	Integer	Identificação da porta onde está instalado o botão de bloqueio

Nome	Tipo	Descrição
<code>serial0</code>	Integer	Código do periférico instalado no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de periféricos disponíveis para instalação nos veículos.
<code>serial1</code>	Integer	Código do periférico instalado no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de periféricos disponíveis para instalação nos veículos.
<code>satelital</code>	boolean	Propriedade que indica se o veículo possui antena satelital: <code>false</code> — Não possui antena satelital <code>true</code> — Possui antena satelital
<code>idSensor5</code>	Integer	Código do sensor instalado na entrada 5 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>idSensor6</code>	Integer	Código do sensor instalado na entrada 6 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>idSensor7</code>	Integer	Código do sensor instalado na entrada 7 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>idSensor8</code>	Integer	Código do sensor instalado na entrada 8 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>idAtuador5</code>	Integer	Código do atuador instalado na saída 5 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>idAtuador6</code>	Integer	Código do atuador instalado na saída 6 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>idAtuador7</code>	Integer	Código do atuador instalado na saída 7 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>idAtuador8</code>	Integer	Código do atuador instalado na saída 8 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.

Nome	Tipo	Descrição
idEquipamentoDesc	String	Descrição do ID equipamento (MTC700, MTC600, MTC550, MXT140A,.)



As informações obtidas podem ser utilizadas para alimentar a tabela 'veiculo' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterVeiculosJson>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>Quantidade</quantidade>
      <!--Optional:-->
      <idVeiculo>idVeiculo</idVeiculo>
    </web:obterVeiculosJson>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterVeiculosJsonResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        {
          "idVeiculo": 53106,
          "placa": "DMH8074",
          "idCliente": 114208,
          "descricao": "GSM/GPS",
          "idEquipamento": 75807,
          "idEquipamentoDesc": "MTC400",
          "idSensor1": 0,
          "idSensor2": 0,
          "idSensor3": 0,
          "idSensor4": 0,
          "idAtuador1": 0,
          "idAtuador2": 0,
          "idAtuador3": 0,
          "idAtuador4": 0,
          "portaPanico": 0,
          "portaBloqueio": 1,
          "idSerial0": 237,
          "idSerial1": null,
          "satelital": false,
          "idSensor5": 0,
          "idSensor6": 0,
        }
      </return>
    </ns0:obterVeiculosJsonResponse>
  </S:Body>
</S:Envelope>
```



```
        "idSensor7": 0,  
        "idSensor8": 0,  
        "idAtuador5": 0,  
        "idAtuador6": 0,  
        "idAtuador7": 0,  
        "idAtuador8": 0,  
        "esn": "null",  
        "idProjeto": null,  
        "isTelemetria": false  
    }  
    </return>  
</ns0:obterVeiculosJsonResponse>  
</S:Body>  
</S:Envelope>
```

4.5. ObterVeiculosRFNacional

Descrição

Método para consultar as informações relativas aos veículos de cada INTEGRADOR.



Caso cliente esteja com restrição, não será possível consultar veículos do mesmo.

Declaração do Método

```
List obterVeiculosRFNacional(String usuario, String senha, Integer quantidade, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Sim	Quantidade máxima de registros da consulta, sendo o limite de 1000 registros incluindo os que possuem restrição, ou seja, pode ser retornado 940 veículos e o restante que seria 60 veículos teriam alguma restrição.; Para obter as informações pertinentes apenas à um registro de veículo, preencha o campo quantidade com o valor 1 (um) e indique o id do veículo desejado no campo idVeiculo.
idVeiculo	Integer	Não	O Limite máximo de resultados do método é 1000 registros. Neste caso para obter o restante dos registros, foi implementado um sistema de paginação, onde o INTEGRADOR deve seguir as seguintes regras: Para obter a primeira página, enviar 0 (zero) como parâmetro; Para obter a próxima página enviar como parâmetro o último id do veículo da última página consultada. Repetir esse processo até que não seja retornado mais nenhum veículo, pois o retorno "vazio" indica o fim da paginação dos veículos.; Para obter as informações pertinentes apenas à um registro de veículo, enviar apenas o id do veículo desejado e no campo quantidade o valor 1 (um).

Retorno

Retorna uma lista de objetos do tipo 'Veiculo' com as propriedades abaixo:

Nome	Tipo	Descrição
ccid	String	Id do SIM Card.

Nome	Tipo	Descrição
descricao	String	Descrição da classe do veículo. Ex: SASTM FULL, SASTM COM TELEMETRIA, SASCARGA FULL SAT 200, SASCARGA FULL SAT I.
idCliente	Integer	Identificação do cliente a qual pertence o veículo.
idVeiculo	Integer	Identificação do veículo.
placa	String	Placa do veículo.
satelital	boolean	Se contem posição satelital.
telemetria	boolean	Se contem telemetria.



As informações obtidas podem ser utilizadas para alimentar a tabela 'veiculo_rfnacional' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterVeiculosRFNacional>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade></quantidade>
      <!--Optional:-->
      <idVeiculo></idVeiculo>
    </web:obterVeiculosRFNacional>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterVeiculosRFNacionalResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <ccid>89550532420010181845</ccid>
        <descricao>RF-Nacional</descricao>
        <idCliente>68553</idCliente>
        <idVeiculo>15213</idVeiculo>
        <placa>MRB1053-1</placa>
        <satelital>false</satelital>
        <telemetria>false</telemetria>
      </return>
    </ns0:obterVeiculosRFNacionalResponse>
  </S:Body>
</S:Envelope>
```

4.6. getVehiclesJSON

Descrição

Método para consultar as informações relativas aos veículos de cada INTEGRADOR. Seu retorno é em forma de Json. Retorno em inglês.



Caso cliente esteja com restrição, não será possível consultar veículos do mesmo.

Declaração do Método

```
List getVehiclesJSON(String user, String password, Integer quantity, Integer vehicleId)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
user	String	Sim	Login do integrador para acesso ao sistema.
password	String	Sim	Senha do integrador para acesso ao sistema.
quantity	Integer	Sim	Quantidade máxima de registros da consulta, sendo o limite de 1000 registros incluindo os que possuem restrição, ou seja, pode ser retornado 940 veículos e o restante que seria 60 veículos teriam alguma restrição.; Para obter as informações pertinentes apenas à um registro de veículo, preencha o campo quantidade com o valor 1 (um) e indique o id do veículo desejado no campo idVeiculo.
vehicleId	Integer	Não	O Limite máximo de resultados do método é 1000 registros. Neste caso para obter o restante dos registros, foi implementado um sistema de paginação, onde o INTEGRADOR deve seguir as seguintes regras: Para obter a primeira página, enviar 0 (zero) como parâmetro; Para obter a próxima página enviar como parâmetro o último id do veículo da última página consultada. Repetir esse processo até que não seja retornado mais nenhum veículo, pois o retorno "vazio" indica o fim da paginação dos veículos.; Para obter as informações pertinentes apenas à um registro de veículo, enviar apenas o id do veículo desejado e no campo quantidade o valor 1 (um).

Retorno

Retorna uma lista de objetos do tipo 'Veiculo' com as propriedades abaixo:

Nome	Tipo	Descrição
vehicleId	Integer	Identificação do veículo.

Nome	Tipo	Descrição
licensePlate	String	Placa do veículo.
clientId	Integer	Identificação do cliente a qual pertence o veículo.
description	String	Descrição da classe do veículo. Ex: SASTM FULL , SASTM COM TELEMETRIA , SASCARGA FULL SAT 200 , SASCARGA FULL SAT I .
deviceId	String	Identificação do equipamento instalado no veículo.
deviceDescId	String	Descrição do ID equipamento (MTC700, MTC600, MTC550, MXT140A,.)
sensor1Id	Integer	Código do sensor instalado na entrada 1 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
sensor2Id	Integer	Código do sensor instalado na entrada 2 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
sensor3Id	Integer	Código do sensor instalado na entrada 3 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
sensor4Id	Integer	Código do sensor instalado na entrada 4 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
actuator1Id	Integer	Código do atuador instalado na saída 1 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
actuator2Id	Integer	Código do atuador instalado na saída 2 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
actuator3Id	Integer	Código do atuador instalado na saída 3 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
actuator4Id	Integer	Código do atuador instalado na saída 4 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
panicPort	Integer	Identificação da porta onde está instalado o botão de pânico

Nome	Tipo	Descrição
<code>blockingPort</code>	Integer	Identificação da porta onde está instalado o botão de bloqueio
<code>serial0Id</code>	Integer	Código do periférico instalado no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de periféricos disponíveis para instalação nos veículos.
<code>serial1Id</code>	Integer	Código do periférico instalado no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de periféricos disponíveis para instalação nos veículos.
<code>layoutId</code>	Integer	Retorna null.
<code>satellite</code>	boolean	Propriedade que indica se o veículo possui antena satelital: • <code>false</code> — Não possui antena satelital • <code>true</code> — Possui antena satelital
<code>sensor5Id</code>	Integer	Código do sensor instalado na entrada 5 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>sensor6Id</code>	Integer	Código do sensor instalado na entrada 6 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>sensor7Id</code>	Integer	Código do sensor instalado na entrada 7 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>sensor8Id</code>	Integer	Código do sensor instalado na entrada 8 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de sensores disponíveis para instalação nos veículos.
<code>actuator5Id</code>	Integer	Código do atuador instalado na saída 5 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>actuator6Id</code>	Integer	Código do atuador instalado na saída 6 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
<code>actuator7Id</code>	Integer	Código do atuador instalado na saída 7 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método <code>ObterAtuadorGrupo()</code> , cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.

Nome	Tipo	Descrição
actuator8Id	Integer	Código do atuador instalado na saída 8 no veículo. Esse id está relacionado com a tabela 'atuador_grupo' ou com o resultado do método ObterAtuadorGrupo(), cujo retorno é a lista de atuadores disponíveis para instalação nos veículos.
esn	String	Retorna null.
projectId	Integer	Retorna null.
isTelemetry	String	Possui telemetria ("true", "false").



As informações obtidas podem ser utilizadas para alimentar a tabela 'veiculo' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/" >
  <soapenv:Header/>
  <soapenv:Body>
    <web:getVehiclesJSON>
      <user>Usuário</user>
      <password>Senha</password>
      <!--Optional:-->
      <quantity>Quantidade</quantity>
      <!--Optional:-->
      <vehicleId>IDVEICULO</vehicleId>
    </web:getVehiclesJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:getVehiclesJSONResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        {
          "vehicleId": 53106,
          "licensePlate": "DMH8074",
          "clientId": 114208,
          "description": "GSM/GPS",
          "deviceId": 75807,
          "deviceDescId": "MTC400",
          "sensor1Id": 0,
          "sensor2Id": 0,
          "sensor3Id": 0,
          "sensor4Id": 0,
          "actuator1Id": 0,
          "actuator2Id": 0,
```



```
        "actuator3Id": 0,  
        "actuator4Id": 0,  
        "panicPort": 0,  
        "blockingPort": 1,  
        "serial0Id": 237,  
        "serial1Id": 0,  
        "layoutId": null,  
        "satellite": false,  
        "sensor5Id": 0,  
        "sensor6Id": 0,  
        "sensor7Id": 0,  
        "sensor8Id": 0,  
        "actuator5Id": 0,  
        "actuator6Id": 0,  
        "actuator7Id": 0,  
        "actuator8Id": 0,  
        "esn": "null",  
        "projectId": null,  
        "isTelemetry": false  
    }  
    </return>  
</ns0:getVehiclesJSONResponse>  
</S:Body>  
</S:Envelope>
```

4.7. ObterDadosAdicionais

Descrição

Método para consultar as informações sobre notas cadastradas nos veículos.

Declaração do Método

```
List obterDadosAdicionais(String usuario, String senha, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idVeiculo	Integer	Não	Para obter página com todos os veículos que possuem nota cadastrada enviar 0 (zero) como parâmetro; Para obter as informações pertinentes apenas à um registro de veículo, enviar apenas o id do veículo desejado.

Retorno

Retorna uma lista de objetos do tipo 'Veiculo' com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Identificação do veículo.
placa	String	Placa do veículo.
idCliente	Integer	Identificação do cliente a qual pertence o veículo.
dataAlteracao	String	Data da ultima alteração realizada.
notaUm	String	Informações cadastradas no sasgc para campo nota 1.
notaDois	String	Informações cadastradas no sasgc para campo nota 2.



Informações serão obtidas somente se houver pelo menos uma nota cadastrada. As informações devem ser cadastrada pela mesma gerenciadora de consulta para o veículo desejado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterDadosAdicionais>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <idVeiculo>IDVEICULO</idVeiculo>
    </web:obterDadosAdicionais>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterDadosAdicionaisResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <dataAlteracao>2023-08-18 05:50:50</dataAlteracao>
        <idCliente>265107</idCliente>
        <idVeiculo>1462395</idVeiculo>
        <notaDois>nota 2 teste OK</notaDois>
        <notaUm>teste bezerra 3</notaUm>
        <placa>0002MTN</placa>
      </return>
    </ns0:obterDadosAdicionaisResponse>
  </S:Body>
</S:Envelope>
```

4.8. ObterDadosAdicionaisCliente

Descrição

Método para consultar os campos editáveis pelo cliente, para visualização da gerenciadora.

Declaração do Método

```
List obterDadosAdicionaisCliente(String usuario, String senha, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idVeiculo	Integer	Não	Para obter página com todos os veículos que possuem nota cadastrada enviar 0 (zero) como parâmetro; Para obter as informações pertinentes apenas à um registro de veículo, enviar apenas o id do veículo desejado.

Retorno

Retorna uma lista de objetos do tipo 'DadosAdicionaisCliente' com as propriedades abaixo:

Nome	Tipo	Descrição
dataAlteracao	String	Data da ultima alteração realizada.
descricaoDois	String	Informações cadastradas no sasgc para campo descrição 2.
descricaoUm	String	Informações cadastradas no sasgc para campo descrição 1.
frota	String	Informações cadastradas no sasgc para campo frota.
grupo	String	Informações cadastradas no sasgc para campo grupo.
idCliente	Integer	Identificação do cliente a qual pertence o veículo.
idVeiculo	Integer	Identificação do veículo.
notaDois	String	Informações cadastradas no sasgc para campo nota 2.
notaUm	String	Informações cadastradas no sasgc para campo nota 1.
placa	String	Placa do veículo



Informações serão obtidas somente se houver pelo menos uma nota cadastrada. As informações devem ser cadastrada pela mesma gerenciadora de consulta para o veículo

desejado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterDadosAdicionaisCliente>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <idVeiculo>881456</idVeiculo>
    </web:obterDadosAdicionaisCliente>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterDadosAdicionaisClienteResponse
      xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <dataAlteracao>2023-07-11 15:11:49</dataAlteracao>
        <descricaoDois/>
        <descricaoUm/>
        <frota/>
        <grupo/>
        <idCliente>50774</idCliente>
        <idVeiculo>881456</idVeiculo>
        <notaDois>03194998562</notaDois>
        <notaUm/>
        <placa>3002TCP</placa>
      </return>
    </ns0:obterDadosAdicionaisClienteResponse>
  </S:Body>
</S:Envelope>
```

4.9. ObterPontosReferencia

Descrição

Método para consultar os pontos de referência criados no SASGC e marcados embarcável.

Declaração do Método

```
List obterPontosReferencia(String usuario, String senha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do usuário para acesso ao sistema.
senha	String	Sim	Senha do usuário para acesso ao sistema

Retorno

Retorna uma lista de objetos do tipo 'PontoReferencia' com as propriedades abaixo:

Nome	Tipo	Descrição
idPontoReferencia	Integer	Identificação do ponto de referência.
codigo	String	Código da Sascar para visualização do ponto.
descricao	String	Descrição do ponto.
latitudes	Double	Latitude superior do ponto.
longitudes	Double	Longitude superior do ponto.
latitudei	Double	Latitude inferior do ponto.
longitudei	Double	Longitude inferior do ponto.
endereco	String	Endereço do ponto criado.
data	Calendar	Data de criação do ponto no servidor de integração.
nome	String	Nome do ponto de referência.



As informações obtidas podem ser utilizadas para alimentar a tabela 'ponto_referencia' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPontosReferencia>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
    </web:obterPontosReferencia>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPontosReferenciaResponse xmlns:ns0=
"http://webservice.web.integracao.sascar.com.br/">
      <return>
        <codigo>1568</codigo>
        <data>2008-01-27T00:00:00.0</data>
        <descricao>Teste de ponto de referencia para o MTC600.</descricao>
        <endereco/>
        <idPontoReferencia>4008</idPontoReferencia>
        <latitudeI>-25.539593</latitudeI>
        <latitudeS>-25.538414</latitudeS>
        <longitudeI>-49.199693</longitudeI>
        <longitudeS>-49.198265</longitudeS>
        <nome>SASCAR - LITORAL</nome>
      </return>
    </ns0:obterPontosReferenciaResponse>
  </S:Body>
</S:Envelope>
```

4.10. ObterRotas

Descrição

Método para consulta de cadastro de rotas associadas aos veículos.

Declaração do Método

```
String obterRotas(String usuario, String senha, String data)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha atual do integrador para acesso ao sistema.
data	String	Não	Data para consulta (YYYY-MM-DD HH:MM:DD)

Retorno

Retorna uma lista de objetos do tipo “ObterRotas” com as propriedades abaixo:

Nome	Tipo	Descrição
Login	String	Identificação do login criador da rota
Id	String	Identificação do id da rota
Nome rota	String	Descrição da rota

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterRotas>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <dataInicio/>
    </web:obterRotas>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:


```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterRotasResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <idrota>992</idrota>
        <loginRota>KIELSE</loginRota>
        <nome>novoll</nome>
      </return>
    </ns0:obterRotasResponse>
  </S:Body>
</S:Envelope>
```

4.11. ObterEnderecoPosicao

Descrição

Método para consultar os dados de endereço através de coordenadas de latitude e longitude retornados em um pacote de posição.

Declaração do Método

```
List obterEnderecoPosicao(String usuario, String senha, String latitude, String longitude)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do usuário para acesso ao sistema.
senha	String	Sim	Senha do usuário para acesso ao sistema
latitude	String	Sim	Coordenada de latitude.
longitude	String	Sim	Coordenada de longitude.

Retorno

Retorna uma lista de objetos do tipo 'EnderecoPosicao' com as propriedades abaixo:

Nome	Tipo	Descrição
cidade	String	Descrição da cidade do endereço da posição.
Rua	String	Descrição da rua do endereço da posição.
uf	String	UF.



As informações obtidas podem ser utilizadas para alimentar a tabela 'enderecoposicao' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterEnderecoPosicao>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <latitude>-6.91337333333333</latitude>
```

```
<longitude>-37.632421666666666</longitude>
</web:obterEnderecoPosicao>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterEnderecoPosicaoResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <address>
          <city>
            <state>PB</state>
            <name>Condado</name>
          </city>
          <street>Rod BR-230</street>
        </address>
      </return>
    </ns0:obterEnderecoPosicaoResponse>
  </S:Body>
</S:Envelope>
```

5. Métodos – Posição

5.1. ObterPacotePosicoes

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicoes(String usuario, String senha, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão. Quando o cliente informar um valor não numérico ou menor ou igual a 0.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	String	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.

Nome	Tipo	Descrição
longitude	Double	Longitude em formato decimal.
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MS (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.

MTC (BIT)	LMU/MSC (Graus)	Direção
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.

MTC (BIT)	LMU/MSC (Graus)	Direção
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: <u>João da Silva</u>.</i>

MTC (BIT)	LMU/MSC (Graus)	Direção
textoMensagem	String	Conteúdo da mensagem tipo texto livre.
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ws="http://webservice.web.integracao.sascar.com.br/"
  <soapenv:Header/>
  <soapenv:Body>
    <ws:obterPacotePosicoes>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <quantidade>valorNumérico</quantidade>
    </ws:obterPacotePosicoes>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicoesResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <anguloReferencia>0</anguloReferencia>
        <bloqueio>0</bloqueio>
        <cidade>Curitiba</cidade>
        <codigoMacro>0</codigoMacro>
        <conteudoMensagem/>
        <dataPacote>2023-10-10T09:05:13.0</dataPacote>
        <dataPosicao>2023-10-10T09:05:14.0</dataPosicao>
        <direcao>189</direcao>
        <distanciaReferencia>0</distanciaReferencia>
        <entrada1>0</entrada1>
        <entrada2>-250</entrada2>
        <entrada3>-251</entrada3>
        <entrada4>-241</entrada4>
        <entrada5>-247</entrada5>
        <entrada6>-248</entrada6>
        <entrada7>-231</entrada7>
        <entrada8>0</entrada8>
        <eventoFormatado/>
        <eventoSeqFormatado/>
        <gps>1</gps>
        <horimetro>1496825</horimetro>
        <idPacote>81204594</idPacote>
        <idReferencia>0</idReferencia>
        <idVeiculo>608581</idVeiculo>
        <ignicao>0</ignicao>
        <integradoraId>80</integradoraId>
        <jamming>0</jamming>
        <latitude>-25.4516521</latitude>
        <longitude>-49.2460098</longitude>
        <memoria>0</memoria>
        <nomeMensagem/>
        <odometro>2662</odometro>
        <pontoEntrada>0</pontoEntrada>
        <pontoReferencia>novo</pontoReferencia>
        <pontoSaida>0</pontoSaida>
        <rpm>0</rpm>
        <rua>R Joao Marchesini</rua>
        <saida1>0</saida1>
        <saida2>-240</saida2>
        <saida3>-254</saida3>
        <saida4>-253</saida4>
        <saida5>0</saida5>
        <saida6>-232</saida6>
      </return>
    </ns0:obterPacotePosicoesResponse>
  </S:Body>
</S:Envelope>
```

```
<saida7>-232</saida7>
<saida8>0</saida8>
<satelite>0</satelite>
<temperatura1>15</temperatura1>
<temperatura2>-125</temperatura2>
<temperatura3>-125</temperatura3>
<temperaturaSerial>-1250</temperaturaSerial>
<tensao>11</tensao>
<textoMensagem/>
<tipoTeclado>0</tipoTeclado>
<uf>PR</uf>
<umidadeSerial>-1250</umidadeSerial>
<velocidade>0</velocidade>
</return>
</ns0:obterPacotePosicoesResponse>
</S:Body>
</S:Envelope>
```

5.2. ObterPacotePosicoesMotorista

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicoesMotorista(String usuario, String senha, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	String	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro atual do veículo.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).

MTC (BIT)	LMU/MSC (Graus)	Direção
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pais	String	País.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma:</i> _João da Silva_.
textoMensagem	String	Conteúdo da mensagem tipo texto livre.

MTC (BIT)	LMU/MSC (Graus)	Direção
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: Código (Integer) – Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicoesMotorista>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>ValorNumérico</quantidade>
    </web:obterPacotePosicoesMotorista>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicoesMotoristaResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <anguloReferencia>134</anguloReferencia>
        <bloqueio>0</bloqueio>
        <cidade>Campo Grande</cidade>
        <codigoMacro>0</codigoMacro>
        <conteudoMensagem/>
        <dataPacote>2023-10-09T11:02:21.0</dataPacote>
        <dataPosicao>2023-10-09T11:02:27.0</dataPosicao>
        <direcao>0</direcao>
        <distanciaReferencia>8563</distanciaReferencia>
        <entrada1>0</entrada1>
        <entrada2>0</entrada2>
        <entrada3>0</entrada3>
        <entrada4>0</entrada4>
        <entrada5>0</entrada5>
        <entrada6>0</entrada6>
        <entrada7>0</entrada7>
        <entrada8>0</entrada8>
        <estadoLimpadorParabrisa>0</estadoLimpadorParabrisa>
        <eventoFormatado/>
        <eventoSeqFormatado/>
        <eventosTelemetria/>
        <gps>1</gps>
        <horimetro>43807</horimetro>
        <idMotorista>0</idMotorista>
        <idPacote>81182489</idPacote>
        <idReferencia>0</idReferencia>
        <idVeiculo>1894048</idVeiculo>
        <ignicao>0</ignicao>
        <integradoraId>80</integradoraId>
        <jamming>0</jamming>
        <latitude>-20.5580288</latitude>
        <longitude>-54.6784256</longitude>
        <memoria>0</memoria>
        <nomeMensagem/>
        <nomeMotorista/>
        <odometro>35541</odometro>
        <pontoEntrada>0</pontoEntrada>
        <pontoReferencia>PE - Couro Azul - Campo Grande, MS</pontoReferencia>
        <pontoSaida>0</pontoSaida>
        <rpm>0</rpm>
        <rua>Sem nome</rua>
        <saida1>0</saida1>
        <saida2>0</saida2>
```

```
<saida3>0</saida3>
<saida4>0</saida4>
<saida5>0</saida5>
<saida6>0</saida6>
<saida7>0</saida7>
<saida8>0</saida8>
<satelite>0</satelite>
<temperatura1>-125</temperatura1>
<temperatura2>-125</temperatura2>
<temperatura3>-125</temperatura3>
<tensao>25</tensao>
<textoMensagem/>
</return>
</ns0:obterPacotePosicoesMotoristaResponse>
</S:Body>
</S:Envelope>
```

5.3. ObterPacotePosicoesMotoristaComPlaca

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos. Além de trazer as informações que o método ObterPacotePosicoes trazia, esse método traz também a informação de Motorista(ID e Nome) e o Status do Limpador de Parabrisa, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicoesMotoristaComPlaca(String usuario, String senha, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	String	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.

Nome	Tipo	Descrição
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro atual do veículo.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória

MTC (BIT)	LMU/MSC (Graus)	Direção
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pais	String	País.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma:</i> _João da Silva_.
textoMensagem	String	Conteúdo da mensagem tipo texto livre.

MTC (BIT)	LMU/MS (Graus)	Direção
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: Código (Integer) – Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
idMotorista	Integer	Identificador do Motorista. (0 caso sem motorista)
nomeMotorista	String	Nome do Motorista se estiver logado.
limpadorParabrisa	Integer	Status do Limpador de para-brisa: • 0 — Limpador Não acionado • 1 — Limpador Acionado
eventosTelemetria	Array<int>	Lista de evento(s) de telemetria composta pelos campos: • idEvento — Identificador do Evento • tempoDuracao — Duração do Evento em segundos • velocidadeMaximaEvento — Velocidade Máxima atingida no Evento • velocidadeReferencia — Velocidade de Referência
placa	String	Placa do veículo.

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicoesMotoristaComPlaca>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>ValorNumérico</quantidade>
    </web:obterPacotePosicoesMotoristaComPlaca>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <work:WorkContext xmlns:work="http://oracle.com/weblogic/soap/workarea/"
XYZ</work:WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicoesMotoristaComPlacaResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <anguloReferencia>0</anguloReferencia>
        <bloqueio>0</bloqueio>
        <cidade>Curitiba</cidade>
        <codigoMacro>0</codigoMacro>
        <conteudoMensagem/>
        <dataPacote>2018-05-04T06:20:16.0</dataPacote>
        <dataPosicao>2018-05-04T06:20:37.0</dataPosicao>
        <direcao>102</direcao>
        <distanciaReferencia>0</distanciaReferencia>
        <entrada1>0</entrada1>
        <entrada2>0</entrada2>
        <entrada3>0</entrada3>
        <entrada4>-241</entrada4>
        <entrada5>-247</entrada5>
        <entrada6>-248</entrada6>
        <entrada7>-231</entrada7>
        <entrada8>0</entrada8>
        <estadoLimpadorParabrisa>0</estadoLimpadorParabrisa>
```

```

<eventoFormatado/>
<eventoSeqFormatado/>
<eventosTelemetria>
  <eventoTelemetria>
    <idEvento>245</idEvento>
    <tempoDuracao>100</tempoDuracao>
    <velocidadeMaximaEvento>100</velocidadeMaximaEvento>
    <velocidadeReferencia>23</velocidadeReferencia>
  </eventoTelemetria>
  <eventoTelemetria>
    <idEvento>24</idEvento>
    <tempoDuracao>100</tempoDuracao>
    <velocidadeMaximaEvento>100</velocidadeMaximaEvento>
    <velocidadeReferencia>40</velocidadeReferencia>
  </eventoTelemetria>
</eventosTelemetria>
<gps>0</gps>
<horimetro>123003</horimetro>
<idMotorista>0</idMotorista>
<idPacote>1315843</idPacote>
<idReferencia>0</idReferencia>
<idVeiculo>406575</idVeiculo>
<ignicao>1</ignicao>
<integradoraId>80</integradoraId>
<jamming>0</jamming>
<latitude>-25.451593</latitude>
<longitude>-49.2465991</longitude>
<memoria>0</memoria>
<nomeMensagem/>
<nomeMotorista/>
<odometro>50000</odometro>
<placa>ABCD1239</placa>
<pontoEntrada>0</pontoEntrada>
<pontoReferencia/>
<pontoSaida>0</pontoSaida>
<rpm>0</rpm>
<rua>Rua Joao Marchesini</rua>
<saida1>0</saida1>
<saida2>-240</saida2>
<saida3>0</saida3>
<saida4>0</saida4>
</return>
</ns0:obterPacotePosicoesMotoristaComPlacaResponse>
</S:Body>
</S:Envelope>

```

```
0 0 0 0 23 -125 2 24 0 PR 0 </ns0: obterPacotePosicoesMotoristaComPlacaResponse>
```

5.4. ObterPacotePosicoesMotoristaRestricao

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos no momento que cliente estava com restrição. Além de trazer as informações que o método ObterPacotePosicoesRestricao trazia, esse método traz também a informação de Motorista(ID e Nome) e o Status do Limpador de Parabrisa, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento. Método exclusivo para pacotes com restrição.

Declaração do Método

```
List obterPacotePosicoesMotoristaRestricao(String usuario, String senha, Integer quantidade, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 300 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.
idVeiculo	Integer	Sim	Código do veículo que pode ser obtido através do método obterVeiculos.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	Calendar	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.

Nome	Tipo	Descrição
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo direcao:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória

MTC (BIT)	LMU/MSC (Graus)	Direção
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma:</i> _João da Silva_.
textoMensagem	String	Conteúdo da mensagem tipo texto livre.

MTC (BIT)	LMU/MS (Graus)	Direção
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
idMotorista	Integer	Identificador do Motorista.
nomeMotorista	String	Nome do Motorista logado.
limpadorParabrisa	Integer	Status do Limpador de para-brisa: • 0 — Limpador Não Acionado • 1 — Limpador Acionado
eventosTelemetria	Array<int>	Lista de evento(s) de telemetria composta pelos campos: • idEvento — Identificador do Evento • tempoDuracao — Duração do Evento em segundos • velocidadeMaximaEvento — Velocidade Máxima atingida no Evento • velocidadeReferencia — Velocidade de Referência



As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:

- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicoesMotoristaRestricao>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>ValorNumérico</quantidade>
      <!--Optional:-->
      <idVeiculo>IdVeiculo</idVeiculo>
    </web:obterPacotePosicoesMotoristaRestricao>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <work:WorkContext xmlns:work="http://oracle.com/weblogic/soap/workarea/">
      XYZ</work:WorkContext>
    </S:Header>
  <S:Body>
    <ns0:obterPacotePosicoesMotoristaRestricaoResponse
      xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <anguloReferencia>0</anguloReferencia>
        <bloqueio>0</bloqueio>
        <cidade>Curitiba</cidade>
        <codigoMacro>0</codigoMacro>
        <conteudoMensagem/>
        <dataPacote>2018-05-02T13:42:22.0</dataPacote>
        <dataPosicao>2018-05-02T13:42:38.0</dataPosicao>
        <direcao>102</direcao>
        <entrada1>0</entrada1>
        <entrada2>0</entrada2>
        <entrada3>0</entrada3>
        <entrada4>-241</entrada4>
        <entrada5>-247</entrada5>
        <entrada6>-248</entrada6>
        <entrada7>-231</entrada7>
        <entrada8>0</entrada8>
        <estadoLimpadorParabrisa>0</estadoLimpadorParabrisa>
        <eventoFormatado/>
        <eventoSeqFormatado/>
        <eventosTelemetria>
          <eventoTelemetria>
            <idEvento>245</idEvento>
            <tempoDuracao>100</tempoDuracao>
          </eventoTelemetria>
          <eventoTelemetria>
            <idEvento>24</idEvento>
            <tempoDuracao>100</tempoDuracao>
            <velocidadeMaximaEvento>100</velocidadeMaximaEvento>
            <velocidadeReferencia>40</velocidadeReferencia>
          </eventoTelemetria>
        </eventosTelemetria>
        <gps>0</gps>
        <horimetro>120567</horimetro>
        <idMotorista>0</idMotorista>
        <idPacote>1310503</idPacote>
        <idReferencia>0</idReferencia>
        <idVeiculo>406575</idVeiculo>
        <ignicao>1</ignicao>
        <integradoraId>80</integradoraId>
        <jamming>0</jamming>
        <latitude>-25.451593</latitude>
        <longitude>-49.2465991</longitude>
      </return>
    </ns0:obterPacotePosicoesMotoristaRestricaoResponse>
  </S:Body>
</S:Envelope>
```

0 50000 0 0 400 Rua Joao Marchesini 0 -240 0 0 0 0 0 23 -125 2 24 0 PR 0

5.5. ObterPacotePosicaoMotoristaPorRange

Descrição

Método que permite obter um ou mais pacotes de posições específicos, através do identificador único de cada pacote. Este método retorna as mesmas informações do método ObterPacotePosicoesMotorista.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicaoMotoristaPorRange(String usuario, String senha, Integer idInicio, Integer idFinal, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idInicio	Integer	Não	Identificador do pacote de posições a partir do qual se deseja que o web service reenvie.
idFinal	Integer	Sim	Identificador do pacote de posições a partir até o qual se deseja que o web service reenvie.
quantidade	Integer	Sim	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.



Caso se deseje receber apenas um pacote de posições, deve-se informar o mesmo idInicio e idFinal.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição

Nome	Tipo	Descrição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	Calendar	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.

MTC (BIT)	LMU/MSC (Graus)	Direção
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.

MTC (BIT)	LMU/MSC (Graus)	Direção
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: <u>João da Silva</u>.</i>

MTC (BIT)	LMU/MSC (Graus)	Direção
textoMensagem	String	Conteúdo da mensagem tipo texto livre.
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
idMotorista	Integer	Identificador do Motorista.
nomeMotorista	String	Nome do Motorista Logado.
limpadorParabrisa	Integer	Status do Limpador de para-brisa: • 0 — Limpador Não acionado • 1 — Limpador Acionado
eventosTelemetria	Array<Int>	Lista de evento(s) de telemetria composta pelos campos: • idEvento — Identificador do Evento • tempoDuracao — Duração do Evento em segundos • velocidadeMaximaEvento — Velocidade Máxima atingida no Evento • velocidadeReferencia — Velocidade de Referência

MTC (BIT)	LMU/MSC (Graus)	Direção
acessorios	Json	Esse campo retorna a lista de acessorios vinculados ao componente "Rede de Acessorios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físicas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALOGICO 6 INDEFINIDO 65.535

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicaoMotoristaPorRange>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <idInicio>PacoteInicial</idInicio>
      <idFinal>PacoteFinal</idFinal>
      <quantidade>ValorNumérico</quantidade>
    </web:obterPacotePosicaoMotoristaPorRange>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicaoMotoristaPorRangeResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <idVeiculo>1186796</idVeiculo>
        <idPacote>81181262</idPacote>
        <dataPosicao>2023-10-09T10:08:43.0</dataPosicao>
        <dataPacote>2023-10-09T10:08:42.0</dataPacote>
        <latitude>-20.997711</latitude>
        <longitude>-51.800974</longitude>
        <direcao>176</direcao>
        <velocidade>59</velocidade>
        <ignicao>1</ignicao>
        <horimetro>1602858</horimetro>
        <odometro>464030</odometro>
        <tensao>25</tensao>
        <saida1>0</saida1>
        <saida2>0</saida2>
        <saida3>0</saida3>
        <saida4>0</saida4>
      </return>
    </ns0:obterPacotePosicaoMotoristaPorRangeResponse>
  </S:Body>
</S:Envelope>
```

```

    <entrada1>0</entrada1>
    <entrada2>0</entrada2>
    <entrada3>0</entrada3>
    <entrada4>0</entrada4>
    <satelite>0</satelite>
    <memoria>0</memoria>
    <idReferencia>0</idReferencia>
    <bloqueio>0</bloqueio>
    <gps>1</gps>
    <uf>MS</uf>
    <cidade>Tres Lagoas</cidade>
    <rua>Rod P/ Brasilandia</rua>
    <pais/>
    <pontoReferencia>PE - PRF. Tres lagoas/MS</pontoReferencia>
    <anguloReferencia>176</anguloReferencia>
    <distanciaReferencia>20052</distanciaReferencia>
    <rpm>0</rpm>
    <temperatura1>-125</temperatura1>
    <temperatura2>-125</temperatura2>
    <temperatura3>-125</temperatura3>
    <saida5>0</saida5>
    <saida6>0</saida6>
    <saida7>0</saida7>
    <saida8>0</saida8>
    <entrada5>0</entrada5>
    <entrada6>0</entrada6>
    <entrada7>0</entrada7>
    <entrada8>0</entrada8>
    <pontoEntrada>0</pontoEntrada>
    <pontoSaida>0</pontoSaida>
    <codigoMacro>0</codigoMacro>
    <nomeMensagem/>
    <conteudoMensagem/>
    <textoMensagem/>
    <tipoTeclado>0</tipoTeclado>
    <jamming>0</jamming>
    <idMotorista>0</idMotorista>
    <nomeMotorista/>
    <integradoraId>80</integradoraId>
    <eventoFormatado/>
    <eventoSeqFormatado/>
    <eventosTelemetria/>
  </return>
</ns0:obterPacotePosicaoMotoristaPorRangeResponse>
</S:Body>
</S:Envelope>

```

5.6. ObterPacotePosicaoMotoristaHistorico

Descrição

Método para consultar histórico de pacotes de posições enviados pelos veículos. Além de trazer as informações que o método ObterPacotePosicaoHistorico trazia, esse método traz também a informação de Motorista(ID e Nome) e o Status do Limpador de Parabrisa, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicaoMotoristaHistorico(String usuario, String senha, String dataInicio, String dataFinal, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
dataInicio	String	Sim	Início do range de data para pesquisa
dataFinal	String	Sim	Fim do range de data para pesquisa
idVeiculo	Integer	Não	Código do veículo.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	String	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.

Nome	Tipo	Descrição
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória

MTC (BIT)	LMU/MSC (Graus)	Direção
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma:</i> _João da Silva_.
textoMensagem	String	Conteúdo da mensagem tipo texto livre.

MTC (BIT)	LMU/MS (Graus)	Direção
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
idMotorista	Integer	Identificador do Motorista.
nomeMotorista	String	Nome do Motorista logado.
limpadorParabrisa	Integer	Status do Limpador de para-brisa: • 0 — Limpador Não Acionado • 1 — Limpador Acionado
eventosTelemetria	Array<int>	Lista de evento(s) de telemetria composta pelos campos: • idEvento — Identificador do Evento • tempoDuracao — Duração do Evento em segundos • velocidadeMaximaEvento — Velocidade Máxima atingida no Evento • velocidadeReferencia — Velocidade de Referência



As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:

- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicaoMotoristaHistorico>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <dataInicio>YYYY-MM-DD HH:MM:SS</dataInicio>
      <dataFinal>2023-10-09 23:59:00</dataFinal>
      <!--Optional:-->
      <idVeiculo>IDVEICULO</idVeiculo>
    </web:obterPacotePosicaoMotoristaHistorico>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <work:WorkContext xmlns:work="http://oracle.com/weblogic/soap/workarea/">
      XYZ</work:WorkContext>
    </S:Header>
  <S:Body>
    <ns0:obterPacotePosicaoMotoristaHistoricoResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <anguloReferencia>0</anguloReferencia>
        <bloqueio>0</bloqueio>
        <cidade>Curitiba</cidade>
        <codigoMacro>0</codigoMacro>
        <conteudoMensagem/>
        <dataPacote>2018-05-02T13:41:21.0</dataPacote>
        <dataPosicao>2018-05-02T13:41:38.0</dataPosicao>
        <direcao>102</direcao>
        <entrada1>0</entrada1>
        <entrada2>0</entrada2>
        <entrada3>0</entrada3>
        <entrada4>-241</entrada4>
        <entrada5>-247</entrada5>
        <entrada6>-248</entrada6>
        <entrada7>-231</entrada7>
        <entrada8>0</entrada8>
        <estadoLimpadorParabrisa>0</estadoLimpadorParabrisa>
        <eventoFormatado/>
        <eventoSeqFormatado/>
        <eventosTelemetria>
          <eventoTelemetria>
```

```

    <idEvento>245</idEvento>
    <tempoDuracao>100</tempoDuracao>
  </eventoTelemetria>
  <eventoTelemetria>
    <idEvento>24</idEvento>
    <tempoDuracao>100</tempoDuracao>
    <velocidadeMaximaEvento>100</velocidadeMaximaEvento>
    <velocidadeReferencia>40</velocidadeReferencia>
  </eventoTelemetria>
</eventosTelemetria>
<gps>0</gps>
<horimetro>120566</horimetro>
<idMotorista>0</idMotorista>
<idPacote>1310500</idPacote>
<idReferencia>0</idReferencia>
<idVeiculo>406575</idVeiculo>
<ignicao>1</ignicao>
<integradoraId>80</integradoraId>
<jamming>0</jamming>
<latitude>-25.451593</latitude>
<longitude>-49.2465991</longitude>
<memoria>0</memoria>
<nomeMensagem/>
<nomeMotorista/>
<odometro>50000</odometro>
<pontoEntrada>0</pontoEntrada>
<pontoReferencia/>
<pontoSaida>0</pontoSaida>
<rpm>400</rpm>
<rua>Rua Joao Marchesini</rua>
<saida1>0</saida1>
<saida2>-240</saida2>
<saida3>0</saida3>
<saida4>0</saida4>
<saida5>0</saida5>
<saida6>0</saida6>
<saida7>0</saida7>
<saida8>0</saida8>
<satelite>0</satelite>
<temperatura1>23</temperatura1>
<temperatura2>-125</temperatura2>
<temperatura3>2</temperatura3>
<tensao>24</tensao>
<textoMensagem/>
<tipoTeclado>0</tipoTeclado>
<uf>PR</uf>
<velocidade>0</velocidade>
  </return>
</ns0:obterPacotePosicaoMotoristaHistoricoResponse>
</S:Body>
</S:Envelope>

```

5.7. ObterPacotePosicaoMotoristaPorRangeJSON

Descrição

Método que permite obter um ou mais pacotes de posições específicos, através do identificador único de cada pacote. Este método retorna as mesmas informações do método ObterPacotePosicoesMotorista.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicaoMotoristaPorRangeJSON(String usuario, String senha, Integer idInicio, Integer idFinal, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idInicio	Integer	Sim	Identificador do pacote de posições a partir do qual se deseja que o web service reenvie.
idFinal	Integer	Sim	Identificador do pacote de posições a partir até o qual se deseja que o web service reenvie.
quantidade	Integer	Sim	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	String	Data do GPS (GMT) no momento do envio da posição.

Nome	Tipo	Descrição
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MS (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.

MTC (BIT)	LMU/MSC (Graus)	Direção
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo "2" (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.

MTC (BIT)	LMU/MSC (Graus)	Direção
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL.
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pais	String	País.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: <u>João da Silva</u>.</i>

MTC (BIT)	LMU/MSC (Graus)	Direção
textoMensagem	String	Conteúdo da mensagem tipo texto livre.
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
idMotorista	Integer	Identificador do Motorista.
nomeMotorista	String	Nome do Motorista.
nivelCombustivel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
litmetro	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litmetro estiver indisponível no veículo.
limpadorParabrisa	Integer	Status do Limpador de para-brisa: • 0 — Limpador Não Acionado • 1 — Limpador Não Acionado

MTC (BIT)	LMU/MSC (Graus)	Direção
eventosTelemetria	Array[int]	Lista de evento(s) de telemetria composta pelos campos: • idEvento — Identificador do Evento • tempoDuracao — Duração do Evento em segundos • velocidadeMaximaEvento — Velocidade Máxima atingida no Evento • velocidadeReferencia — Velocidade de Referência

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicaoMotoristaPorRangeJSON>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <idInicio>idPacoteInicio</idInicio>
      <idFinal>idPacoteFinal</idFinal>
      <quantidade>ValorNumérico</quantidade>
    </web:obterPacotePosicaoMotoristaPorRangeJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicaoMotoristaPorRangeJSONResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>{
        "idVeiculo": 1126486,
        "dataPosicao": "2026-08-10 00:00:00.0",
        "dataPacote": "2026-08-10 00:00:00.0",
        "latitude": -25.449528,
        "longitude": -49.288528,
        "direcao": 0,
        "velocidade": 0,
        "ignicao": 1,
        "odometro": 36,
        "horimetro": 6570,
        "tensao": 10,
        "saida1": 0,
        "saida2": -240,
        "saida3": 0,
        "saida4": 0,
        "entrada1": 0,
        "entrada2": 0,
        "entrada3": 0,
      }
```

```

"entrada4": 0,
"satelite": 0,
"memoria": 0,
"idReferencia": 0,
"bloqueio": 0,
"gps": 0,
"uf": "PR",
"cidade": "Curitiba",
"rua": "Av Iguacu",
"pais": "BR",
"pontoReferencia": "",
"anguloReferencia": 0,
"distanciaReferencia": 0,
"rpm": 0,
"temperatura1": 32,
"temperatura2": -125,
"temperatura3": -125,
"saida5": 0,
"saida6": 0,
"saida7": 0,
"saida8": 0,
"entrada5": -247,
"entrada6": -248,
"entrada7": -231,
"entrada8": 0,
"pontoEntrada": 0,
"pontoSaida": 0,
"codigoMacro": 0,
"nomeMensagem": "",
"conteudoMensagem": "",
"textoMensagem": "",
"tipoTeclado": 0,
"eventoSequenciamento": [
],
"eventos": [
  {
    "codigo": -247
  },
  {
    "codigo": -240
  }
],
"jamming": 0,
"statusAncora": 0,
"idPacote": 104648958,
"integradoraId": 80,
"idMotorista": 0,
"nomeMotorista": "",
"nivelCombustivel": "0",
"litrometro": "0.0",
"estadoLimpadorParabrisa": 0,
"eventosTelemetria": [
],
"odometroExato": 36.0,
"acessorios": [
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 2,
    "valor": 59.2,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 4,
    "indice": 2,
    "valor": 60.0,
    "local": 2
  }
]

```

```
    },  
    {  
      "numeroSerial": "",  
      "tipo": 3,  
      "indice": 1,  
      "valor": -34.7,  
      "local": 2  
    },  
    {  
      "numeroSerial": "",  
      "tipo": 4,  
      "indice": 1,  
      "valor": 75.0,  
      "local": 2  
    }  
  ]  
}</return>  
</ns0:obterPacotePosicaoMotoristaPorRangeJSONResponse>  
</S:Body>  
</S:Envelope>
```

5.8. ObterPacotePosicoesMotoristaJSON

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos. Além de trazer as informações que o método ObterPacotePosicoesJSON trazia, esse método traz também a informação de Motorista(ID e Nome) e o Status do Limpador de Parabrisa, sendo o seu conteúdo composto pelos seguintes blocos de informações:

- Pacote de posição;
- Pacote de mensagens (todos os tipos);
- Pacote de eventos;
- Pacote Alarme Sequenciamento.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicoesMotoristaJSON(String usuario, String senha, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Sim	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.

Nome	Tipo	Descrição
<code>dataPacote</code>	String	Data do GPS (GMT) no momento do envio da posição.
<code>latitude</code>	Double	Latitude em formato decimal.
<code>longitude</code>	Double	Longitude em formato decimal.
<code>direcao</code>	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
<code>velocidade</code>	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
<code>ignicao</code>	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
<code>horimetro</code>	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
<code>odometro</code>	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
<code>tensao</code>	Integer	Valor de tensão da bateria.
<code>saida1</code>	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.

MTC (BIT)	LMU/MSC (Graus)	Direção
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.

MTC (BIT)	LMU/MSC (Graus)	Direção
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pais	String	País.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: <u>João da Silva</u>.</i>

MTC (BIT)	LMU/MSC (Graus)	Direção
textoMensagem	String	Conteúdo da mensagem tipo texto livre.
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
idMotorista	Integer	Identificador do Motorista.
nomeMotorista	String	Nome do Motorista logado.
nivelCombustivel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
litmetro	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litmetro estiver indisponível no veículo.
limpadorParabrisa	Integer	Status do Limpador de para-brisa: • 0 — Limpador Não Acionado • 1 — Limpador Acionado

MTC (BIT)	LMU/MS (Graus)	Direção
eventosTelemetria	Array[int]	Lista de evento(s) de telemetria composta pelos campos: • idEvento — Identificador do Evento • tempoDuracao — Duração do Evento em segundos • velocidadeMaximaEvento — Velocidade Máxima atingida no Evento • velocidadeReferencia — Velocidade de Referência
acessorios	Json	Esse campo retorna a lista de acessorios vinculados ao componente "Rede de Acessorios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físcas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Indice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSAO 5 ANALOGICO 6 INDEFINIDO 65.535

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicoesMotoristaJSON>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <quantidade>ValorNumérico</quantidade>
    </web:obterPacotePosicoesMotoristaJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicoesMotoristaJSONResponse
      xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>{
        "idVeiculo": 1126486,
        "dataPosicao": "2026-08-10 00:11:00.0",
        "dataPacote": "2026-08-10 00:11:00.0",
        "latitude": -25.449528,
        "longitude": -49.288528,
        "direcao": 0,
        "velocidade": 0,
        "ignicao": 1,
        "odometro": 36,
        "horimetro": 6570,
        "tensao": 10,
        "saida1": 0,
        "saida2": -240,
        "saida3": 0,
        "saida4": 0,
        "entrada1": 0,
        "entrada2": 0,
        "entrada3": 0,
        "entrada4": 0,
        "satelite": 0,
        "memoria": 0,
        "idReferencia": 0,
        "bloqueio": 0,
        "gps": 0,
        "uf": "PR",
        "cidade": "Curitiba",
        "rua": "Av Iguacu",
        "pais": "BR",
        "pontoReferencia": "",
        "anguloReferencia": 0,
        "distanciaReferencia": 0,
        "rpm": 0,
        "temperatura1": 32,
        "temperatura2": -125,
        "temperatura3": -125,
        "saida5": 0,
        "saida6": 0,
        "saida7": 0,
        "saida8": 0,
        "entrada5": -247,
        "entrada6": -248,
        "entrada7": -231,
        "entrada8": 0,
        "pontoEntrada": 0,
        "pontoSaida": 0,
        "codigoMacro": 0,
        "nomeMensagem": "",
        "conteudoMensagem": "",
        "textoMensagem": "",
        "tipoTeclado": 0,
        "eventoSequenciamento": [
        ],
        "eventos": [
        ],
        "jamming": 0,
        "statusAncora": 0,
        "idPacote": 104649044,
        "integradoraId": 80,
        "idMotorista": 0,
      }
```

```
"nomeMotorista": "",
"nivelCombustivel": "0",
"litrometro": "0.0",
"estadoLimpadorParabrisa": 0,
"eventosTelemetria": [
],
"odometroExato": 36.0,
"acessorios": [
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 2,
    "valor": 59.2,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 4,
    "indice": 2,
    "valor": 60.0,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 1,
    "valor": -34.7,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 4,
    "indice": 1,
    "valor": 75.0,
    "local": 2
  }
]
}</return>
</ns0:obterPacotePosicoesMotoristaJSONResponse>
</S:Body>
</S:Envelope>
```

5.9. ObterPacotePosicoesRestricao

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos no momento que cliente estava restricao, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento. Método exclusivo para pacotes com restricao.

Declaração do Método

```
List obterPacotePosicoesRestricao(String usuario, String senha, Integer quantidade, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 300 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.
idVeiculo	Integer	Sim	Código do veículo que pode ser obtido através do método obterVeiculos.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	Calendar	Data da gravação da posição no servidor.
dataPacote	Calendar	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo direcao:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).

MTC (BIT)	LMU/MSC (Graus)	Direção
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: _João da Silva_.</i>
textoMensagem	String	Conteúdo da mensagem tipo texto livre.
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT

MTC (BIT)	LMU/MSC (Graus)	Direção
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicoesRestricao>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <quantidade>ValorNumérico</quantidade>
      <idVeiculo>idVeiculo</idVeiculo>
    </web:obterPacotePosicoesRestricao>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:obterPacotePosicoesRestricaoResponse xmlns:ns2="http://ws.integra.sascar.com.br/">
      <return>
        <bloqueio>0</bloqueio>
        <cidade>São José dos Pinhais</cidade>
        <conteudoMensagem/>
        <dataPacote>2012-07-27T07:38:04-03:00</dataPacote>
        <dataPosicao>2012-07-27T10:53:17-03:00</dataPosicao>
        <direcao>4</direcao>
        <entrada1>0</entrada1>
        <entrada2>0</entrada2>
        <entrada3>0</entrada3>
        <entrada4>-241</entrada4>
        <entrada5>-247</entrada5>
        <entrada6>-248</entrada6>
        <entrada7>-231</entrada7>
        <entrada8>0</entrada8>
        <eventos>
          <codigo>501</codigo>
        </eventos>
        <eventos>
          <codigo>599</codigo>
        </eventos>
        <eventos>
          <codigo>653</codigo>
        </eventos>
        <eventos>
          <codigo>1002</codigo>
        </eventos>
        <eventos>
          <codigo>652</codigo>
        </eventos>
        <gps>1</gps>
        <idReferencia>0</idReferencia>
        <horimetro>94297</horimetro>
        <idPacote>6199608</idPacote>
        <idVeiculo>325900</idVeiculo>
        <ignicao>1</ignicao>
        <latitude>-25.540326666666665</latitude>
        <longitude>-49.200445</longitude>
        <memoria>1</memoria>
        <nomeMensagem/>
        <odometro>2028335</odometro>
        <pontoEntrada>0</pontoEntrada>
        <pontoReferencia/>
        <pontoSaida>0</pontoSaida>
        <rpm>1119</rpm>
        <rua>R Voluntários da Pátria</rua>
        <saida1>0</saida1>
        <saida2>-240</saida2>
        <saida3>0</saida3>
        <saida4>0</saida4>
      </return>
    </ns2:obterPacotePosicoesRestricaoResponse>
  </S:Body>
</S:Envelope>
```

```
0 0 0 0 -125 -125 -125 0 PR 24 <jamming>0</jamming> </ns2:obterPacotePosicoesRestricaoResponse>
```

5.10. ObterPacotePosicaoPorRange

Descrição

Método que permite obter um ou mais pacotes de posições específicos, através do identificador único de cada pacote. Este método retorna as mesmas informações do método ObterPacotePosicoes.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicaoPorRange(String usuario, String senha, Integer idInicio, Integer idFinal, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idInicio	Integer	Sim	Identificador do pacote de posições a partir do qual se deseja que o web service reenvie.
idFinal	Integer	Sim	Identificador do pacote de posições a partir até o qual se deseja que o web service reenvie.
quantidade	Integer	Sim	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.



Caso se deseje receber apenas um pacote de posições, deve-se informar o mesmo idInicio e idFinal.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição

Nome	Tipo	Descrição
<code>dataPosicao</code>	String	Data da gravação da posição no servidor.
<code>dataPacote</code>	String	Data do GPS (GMT) no momento do envio da posição.
<code>latitude</code>	Double	Latitude em formato decimal.
<code>longitude</code>	Double	Longitude em formato decimal.
<code>direcao</code>	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
<code>velocidade</code>	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
<code>ignicao</code>	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
<code>horimetro</code>	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
<code>odometro</code>	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
<code>tensao</code>	Integer	Valor de tensão da bateria.
<code>saida1</code>	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.

MTC (BIT)	LMU/MSC (Graus)	Direção
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.

MTC (BIT)	LMU/MSC (Graus)	Direção
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: <u>João da Silva</u>.</i>

MTC (BIT)	LMU/MSC (Graus)	Direção
textoMensagem	String	Conteúdo da mensagem tipo texto livre.
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicaoPorRange>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <idInicio>ValorIDInicial</idInicio>
      <!--Optional:-->
      <idFinal>ValorIDFinal</idFinal>
      <quantidade>ValorNumérico</quantidade>
    </web:obterPacotePosicaoPorRange>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">
      r00ABXdRABt3ZWJsb2dpYy5hcHAubW9kdWxvLXdLYi1lYXIAAADWAAAAI3dLYmxvZ2ljLndvcmtbcmVhLlN0cmLuZ1dvcmtDb
      250ZXh0AAV2XzQwMQAA</WorkContext>
    </S:Header>
  <S:Body>
    <ns0:obterPacotePosicaoPorRangeResponse
      xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <anguloReferencia>168</anguloReferencia>
        <bloqueio>0</bloqueio>
        <cidade>Cajuru</cidade>
        <codigoMacro>0</codigoMacro>
        <conteudoMensagem/>
        <dataPacote>2023-10-10T09:05:24.0</dataPacote>
        <dataPosicao>2023-10-10T09:05:39.0</dataPosicao>
        <direcao>114</direcao>
        <distanciaReferencia>13563</distanciaReferencia>
        <entrada1>0</entrada1>
        <entrada2>0</entrada2>
        <entrada3>0</entrada3>
        <entrada4>-241</entrada4>
        <entrada5>-247</entrada5>
        <entrada6>-248</entrada6>
        <entrada7>-231</entrada7>
        <entrada8>0</entrada8>
        <eventoFormatado/>
        <eventoSeqFormatado/>
        <gps>1</gps>
        <horimetro>92352</horimetro>
        <idPacote>81204600</idPacote>
        <idReferencia>0</idReferencia>
        <idVeiculo>1731832</idVeiculo>
        <ignicao>1</ignicao>
        <integradoraId>80</integradoraId>
        <jamming>0</jamming>
        <latitude>-21.154035</latitude>
        <longitude>-47.3294983</longitude>
        <memoria>1</memoria>
        <nomeMensagem/>
        <odometro>281582</odometro>
        <pontoEntrada>0</pontoEntrada>
        <pontoReferencia>PE - Cajuru-SP</pontoReferencia>
        <pontoSaida>0</pontoSaida>
        <rpm>601</rpm>
        <rua>Unnamed Road</rua>
        <saida1>0</saida1>
        <saida2>-240</saida2>
        <saida3>0</saida3>
        <saida4>0</saida4>
        <saida5>0</saida5>
        <saida6>-232</saida6>
        <saida7>0</saida7>
        <saida8>0</saida8>
        <satelite>0</satelite>
        <temperatura1>-125</temperatura1>
        <temperatura2>-125</temperatura2>
        <temperatura3>-125</temperatura3>
        <temperaturaSerial>-1250</temperaturaSerial>
        <tensao>28</tensao>
        <textoMensagem/>
        <tipoTeclado>0</tipoTeclado>
        <uf>SP</uf>
        <umidadeSerial>-1250</umidadeSerial>
      </return>
    </ns0:obterPacotePosicaoPorRangeResponse>
  </S:Body>
</S:Envelope>
```

```
<velocidade>0</velocidade>
</return>
</ns0:obterPacotePosicaoPorRangeResponse>
</S:Body>
</S:Envelope>
```

5.11. ObterPacoteLocalizacao

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos.

Declaração do Método

```
List obterPacoteLocalizacao(String usuario, String senha, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Sim	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 2000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
dataPacote	Calendar	Data do GPS (GMT) no momento do envio da posição.
direcao	Integer	Direção do veículo. Ver tabela abaixo.
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
idVeiculo	Integer	Id do veículo.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.

Nome	Tipo	Descrição
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/" >
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacoteLocalizacao>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>Quantidade</quantidade>
    </web:obterPacoteLocalizacao>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:obterPacoteLocalizacaoResponse xmlns:ns0="http://webservice.web.integracao.sascar/" >
      <return>
        <dataPacote>2015-04-06T12:00:42.0</dataPacote>
        <direcao>4</direcao>
        <gps>0</gps>
        <idVeiculo>021456</idVeiculo>
        <ignicao>1</ignicao>
        <latitude>-23.62143</latitude>
        <longitude>-46.829361666666664</longitude>
        <velocidade>60</velocidade>
      </return>
    </ns0:obterPacoteLocalizacaoResponse>
  </S:Body>
</S:Envelope>
```

```
</return>  
</ns0:obterPacoteLocalizacaoResponse>  
</S:Body>  
</S:Envelope>
```

5.12. ObterPacotePosicaoHistorico

Descrição

Método para consultar histórico de pacotes de posições enviados pelos veículos, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicaoHistorico(String usuario, String senha, String dataInicio, String dataFinal, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
dataInicio	String	Sim	Início do range de data para pesquisa. Formato: (YYYY-MM-DD HH:MM:SS).
dataFinal	String	Sim	Fim do range de data para pesquisa. Formato: (YYYY-MM-DD HH:MM:SS).
idVeiculo	Integer	Não	Código do veículo.

Retorno

O sistema possui um limite padrão de 3000 registros por requisição. Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	String	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.

Nome	Tipo	Descrição
longitude	Double	Longitude em formato decimal.
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MS (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.

MTC (BIT)	LMU/MSC (Graus)	Direção
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo "2" (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.

MTC (BIT)	LMU/MSC (Graus)	Direção
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: <u>João da Silva</u>.</i>

MTC (BIT)	LMU/MSC (Graus)	Direção
textoMensagem	String	Conteúdo da mensagem tipo texto livre.
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicaoHistorico>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <dataInicio>2023-10-16 00:00:00</dataInicio>
      <dataFinal>2023-10-16 23:59:00</dataFinal>
      <!--Optional:-->
      <idVeiculo>900757</idVeiculo>
    </web:obterPacotePosicaoHistorico>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicaoHistoricoResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <anguloReferencia>0</anguloReferencia>
        <bloqueio>0</bloqueio>
        <cidade>Curitiba</cidade>
        <codigoMacro>0</codigoMacro>
        <conteudoMensagem/>
        <dataPacote>2023-10-16T00:00:03.0</dataPacote>
        <dataPosicao>2023-10-16T00:00:09.0</dataPosicao>
        <direcao>0</direcao>
        <distanciaReferencia>0</distanciaReferencia>
        <entrada1>0</entrada1>
        <entrada2>0</entrada2>
        <entrada3>-231</entrada3>
        <entrada4>-247</entrada4>
        <entrada5>-248</entrada5>
        <entrada6>-241</entrada6>
        <entrada7>-250</entrada7>
        <entrada8>-251</entrada8>
        <eventoFormatado/>
        <eventoSeqFormatado/>
        <gps>1</gps>
        <horimetro>261533</horimetro>
        <idPacote>81327633</idPacote>
        <idReferencia>2387650</idReferencia>
        <idVeiculo>900757</idVeiculo>
        <ignicao>1</ignicao>
        <integradoraId>80</integradoraId>
        <jamming>0</jamming>
        <latitude>-25.451672</latitude>
        <longitude>-49.2460896</longitude>
        <memoria>0</memoria>
        <nomeMensagem/>
        <odometro>559</odometro>
        <pontoEntrada>0</pontoEntrada>
        <pontoReferencia>JUNIOR</pontoReferencia>
        <pontoSaida>0</pontoSaida>
        <rpm>0</rpm>
        <rua>R Joao Marchesini</rua>
        <saida1>0</saida1>
        <saida2>-240</saida2>
        <saida3>254</saida3>
        <saida4>253</saida4>
        <saida5>245</saida5>
        <saida6>0</saida6>
        <saida7>-232</saida7>
        <saida8>0</saida8>
        <satelite>0</satelite>
        <temperatura1>-125</temperatura1>
        <temperatura2>-125</temperatura2>
        <temperatura3>-125</temperatura3>
        <tensao>11</tensao>
        <textoMensagem/>
        <tipoTeclado>0</tipoTeclado>
        <uf>PR</uf>
        <velocidade>0</velocidade>
      </return>
    </ns0:obterPacotePosicaoHistoricoResponse>
  </S:Body>
</S:Envelope>
```

5.13. ObterPacotePosicoesJSON

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicoesJSON(String usuario, String senha, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	String	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).

MTC (BIT)	LMU/MSC (Graus)	Direção
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pais	String	País.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma:</i> _João da Silva_.
textoMensagem	String	Conteúdo da mensagem tipo texto livre.

MTC (BIT)	LMU/MS (Graus)	Direção
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
nivelCombustivel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
litmetro	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litmetro estiver indisponível no veículo.
acessorios	Json	Esse campo retorna a lista de acessórios vinculados ao componente "Rede de Acessórios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físicas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALÓGICO 6 INDEFINIDO 65.535



As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:

- pacote
- evento

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicoesJSON>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>?</quantidade>
    </web:obterPacotePosicoesJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicoesJSONResponse xmlns:ns0=
"http://webservice.web.integracao.sascar.com.br/">
      <return>{
        "idVeiculo": 1126486,
        "dataPosicao": "2026-08-10 00:06:00.0",
        "dataPacote": "2026-08-10 00:06:00.0",
        "latitude": -25.449528,
        "longitude": -49.288528,
        "direcao": 0,
        "velocidade": 0,
        "ignicao": 1,
        "odometro": 36,
        "horimetro": 6570,
        "tensao": 10,
        "saida1": 0,
        "saida2": -240,
        "saida3": 0,
        "saida4": 0,
        "entrada1": 0,
        "entrada2": 0,
        "entrada3": 0,
        "entrada4": 0,
        "satelite": 0,
        "memoria": 0,
        "idReferencia": 0,
        "bloqueio": 0,
        "gps": 0,
        "uf": "PR",
        "cidade": "Curitiba",
        "rua": "Av Iguacu",
        "pais": "BR",
        "pontoReferencia": "",
        "anguloReferencia": 0,
```

```
"distanciaReferencia": 0,
"rpm": 0,
"temperatura1": 32,
"temperatura2": -125,
"temperatura3": -125,
"saida5": 0,
"saida6": 0,
"saida7": 0,
"saida8": 0,
"entrada5": -247,
"entrada6": -248,
"entrada7": -231,
"entrada8": 0,
"pontoEntrada": 0,
"pontoSaida": 0,
"codigoMacro": 0,
"nomeMensagem": "",
"conteudoMensagem": "",
"textoMensagem": "",
"tipoTeclado": 0,
"eventoSequenciamento": [
],
"eventos": [
  {
    "codigo": -240
  }
],
"jamming": 0,
"statusAncora": 0,
"idPacote": 104649002,
"integradoraId": 80,
"idMotorista": 0,
"nomeMotorista": "",
"nivelCombustivel": "0",
"litrometro": "0.0",
"estadoLimpadorParabrisa": null,
"umidadeSerial": 0,
"temperaturaSerial": 0,
"odometroExato": 36.0,
"acessorios": [
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 2,
    "valor": 59.2,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 4,
    "indice": 2,
    "valor": 60.0,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 1,
    "valor": -34.7,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 4,
    "indice": 1,
    "valor": 75.0,
    "local": 2
  }
]
```

```
    ]  
  }</return>  
</ns0:obterPacotePosicoesJSONResponse>  
</S:Body>  
</S:Envelope>
```


5.14. ObterPacotePosicoesJSONComPlaca

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicoesJSONComPlaca(String usuario, String senha, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição
dataPosicao	String	Data da gravação da posição no servidor.
dataPacote	String	Data do GPS (GMT) no momento do envio da posição.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.
direcao	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
velocidade	Integer	Velocidade obtida do GPS.
ignicao	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
horimetro	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
odometro	Integer	Odômetro atual do veículo.
tensao	Integer	Valor de tensão da bateria.
saida1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).

MTC (BIT)	LMU/MSC (Graus)	Direção
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pais	String	País.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: _João da Silva_.</i>
textoMensagem	String	Conteúdo da mensagem tipo texto livre.

MTC (BIT)	LMU/MS (Graus)	Direção
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
nivelCombustivel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
litometro	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litometro estiver indisponível no veículo.
Placa	String	Placa do veículo.
acessorios	Json	Esse campo retorna a lista de acessórios vinculados ao componente "Rede de Acessórios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físicas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALÓGICO 6 INDEFINIDO 65.535



As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de

dados de integração:

- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicoesJSONComPlaca>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>?</quantidade>
    </web:obterPacotePosicoesJSONComPlaca>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicoesJSONComPlacaResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>{
        "idVeiculo": 1126486,
        "dataPosicao": "2026-08-10 00:08:00.0",
        "dataPacote": "2026-08-10 00:08:00.0",
        "latitude": -25.449528,
        "longitude": -49.288528,
        "direcao": 0,
        "velocidade": 0,
        "ignicao": 1,
        "odometro": 36,
        "horimetro": 6570,
        "tensao": 10,
        "saida1": 0,
        "saida2": -240,
        "saida3": 0,
        "saida4": 0,
        "entrada1": 0,
        "entrada2": 0,
        "entrada3": 0,
        "entrada4": 0,
        "satelite": 0,
        "memoria": 0,
      }
```

```
"idReferencia": 0,
"bloqueio": 0,
"gps": 0,
"uf": "PR",
"cidade": "Curitiba",
"rua": "Av Iguacu",
"pais": "BR",
"pontoReferencia": "",
"anguloReferencia": 0,
"distanciaReferencia": 0,
"rpm": 0,
"temperatura1": 32,
"temperatura2": -125,
"temperatura3": -125,
"saida5": 0,
"saida6": 0,
"saida7": 0,
"saida8": 0,
"entrada5": -247,
"entrada6": -248,
"entrada7": -231,
"entrada8": 0,
"pontoEntrada": 0,
"pontoSaida": 0,
"codigoMacro": 0,
"nomeMensagem": "",
"conteudoMensagem": "",
"textoMensagem": "",
"tipoTeclado": 0,
"eventoSequenciamento": [
],
"eventos": [
  {
    "codigo": -240
  }
],
"jamming": 0,
"statusAncora": 0,
"idPacote": 104649024,
"integradoraId": 80,
"idMotorista": 0,
"nomeMotorista": "",
"nivelCombustivel": "0",
"litrometro": "0.0",
"estadoLimpadorParabrisa": null,
"umidadeSerial": 0,
"temperaturaSerial": 0,
"placa": "0000APO-1",
"odometroExato": 36.0,
"acessorios": [
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 2,
    "valor": 59.2,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 4,
    "indice": 2,
    "valor": 60.0,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 1,
```



```
        "valor": -34.7,  
        "local": 2  
    },  
    {  
        "numeroSerial": "",  
        "tipo": 4,  
        "indice": 1,  
        "valor": 75.0,  
        "local": 2  
    }  
]  
    }</return>  
</ns0:obterPacotePosicoesJSONComPlacaResponse>  
</S:Body>  
</S:Envelope>
```

5.15. ObterPacotePosicaoPorRangeJSON

Descrição

Método que permite obter um ou mais pacotes de posições específicos, através do identificador único de cada pacote. Este método retorna as mesmas informações do método ObterPacotePosicoes.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List obterPacotePosicaoPorRangeJSON(String usuario, String senha, Integer idInicio, Integer idFinal, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idInicio	Integer	Sim	Identificador do pacote de posições a partir do qual se deseja que o web service reenvie.
idFinal	Integer	Sim	Identificador do pacote de posições a partir até o qual se deseja que o web service reenvie.
quantidade	Integer	Sim	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.



Caso se deseje receber apenas um pacote de posições, deve-se informar o mesmo idInicio e idFinal.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Código do veículo.
idPacote	Integer	Identificador único do pacote de posição

Nome	Tipo	Descrição
<code>dataPosicao</code>	String	Data da gravação da posição no servidor.
<code>dataPacote</code>	String	Data do GPS (GMT) no momento do envio da posição.
<code>latitude</code>	Double	Latitude em formato decimal.
<code>longitude</code>	Double	Longitude em formato decimal.
<code>direcao</code>	Integer	Direção do veículo. Ver tabela abaixo.

Valores para o campo **direcao**:

MTC (BIT)	LMU/MSC (Graus)	Direção
0	0 a 44	(N - Norte)
1	45 a 89	(NE - Nordeste)
2	90 a 134	(L - Leste)
3	135 a 179	(SE - Sudeste)
4	180 a 224	(S - Sul)
5	225 a 269	(SO - Sudoeste)
6	270 a 314	(O - Oeste)
7	315 a 359	(NO - Noroeste)
<code>velocidade</code>	Integer	Velocidade obtida do GPS, caso o veículo possua Telemetria, a velocidade será a da Telemetria.
<code>ignicao</code>	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
<code>horimetro</code>	Integer	Horímetro atual do veículo. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
<code>odometro</code>	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
<code>tensao</code>	Integer	Valor de tensão da bateria.
<code>saida1</code>	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.

MTC (BIT)	LMU/MSC (Graus)	Direção
saida2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: • 240 (Código de sirene informando que está ativada) • -240 (Código de sirene informando que está desativada) • 0 (Nenhum dispositivo).
saida3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satelite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo "2" (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.

MTC (BIT)	LMU/MS (Graus)	Direção
memoria	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
idReferencia	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
bloqueio	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
uf	String	Unidade da Federação.
cidade	String	Cidade.
rua	String	Endereço.
pais	String	País.
pontoReferencia	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
anguloReferencia	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
distanciaReferencia	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperatura1	Integer	Dados do sensor de temperatura 1.
temperatura2	Integer	Dados do sensor de temperatura 2.
temperatura3	Integer	Dados do sensor de temperatura 3.
saida5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

MTC (BIT)	LMU/MSC (Graus)	Direção
saida6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
saida8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entrada8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
pontoEntrada	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
pontoSaida	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
codigoMacro	Integer	Código da macro.
conteudoMensagem	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: <u>João da Silva</u>.</i>

MTC (BIT)	LMU/MSC (Graus)	Direção
textoMensagem	String	Conteúdo da mensagem tipo texto livre.
tipoTeclado	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
eventoSequenciamento	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: Id (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); tempo (Integer) (Duração do alarme)
evento	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: codigo (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
nivelCombustivel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
litrometro	Double	Quantidade de combustível consumida durante a vida útil do veículo.
acessorios	Json	Esse campo retorna a lista de acessórios vinculados ao componente "Rede de Acessórios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físcas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALÓGICO 6 INDEFINIDO 65.535

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicaoPorRangeJSON>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <idInicio>IDPACOTEINICIAL</idInicio>
      <idFinal>IDPACOTEFINAL</idFinal>
      <quantidade>QUANTIDADE</quantidade>
    </web:obterPacotePosicaoPorRangeJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterPacotePosicaoPorRangeJSONResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>{
        "idVeiculo": 1126486,
        "dataPosicao": "2026-08-10 00:00:00.0",
        "dataPacote": "2026-08-10 00:00:00.0",
        "latitude": -25.449528,
        "longitude": -49.288528,
        "direcao": 0,
        "velocidade": 0,
        "ignicao": 1,
        "odometro": 36,
        "horimetro": 6570,
        "tensao": 10,
        "saida1": 0,
        "saida2": -240,
        "saida3": 0,
        "saida4": 0,
        "entrada1": 0,
        "entrada2": 0,
        "entrada3": 0,
        "entrada4": 0,
        "satelite": 0,
        "memoria": 0,
        "idReferencia": 0,
        "bloqueio": 0,
        "gps": 0,
        "uf": "PR",
        "cidade": "Curitiba",
        "rua": "Av Iguacu",
        "pais": "BR",
        "pontoReferencia": "",
        "anguloReferencia": 0,
        "distanciaReferencia": 0,
        "rpm": 0,
        "temperatura1": 32,
        "temperatura2": -125,
        "temperatura3": -125,
        "saida5": 0,
```



```
"saida6": 0,
"saida7": 0,
"saida8": 0,
"entrada5": -247,
"entrada6": -248,
"entrada7": -231,
"entrada8": 0,
"pontoEntrada": 0,
"pontoSaida": 0,
"codigoMacro": 0,
"nomeMensagem": "",
"conteudoMensagem": "",
"textoMensagem": "",
"tipoTeclado": 0,
"eventoSequenciamento": [
],
"eventos": [
  {
    "codigo": -247
  },
  {
    "codigo": -240
  }
],
"jamming": 0,
"statusAncora": 0,
"idPacote": 104648958,
"integradoraId": 80,
"idMotorista": null,
"nomeMotorista": "null",
"nivelCombustivel": "0",
"litometro": "0.0",
"estadoLimpadorParabrisa": null,
"umidadeSerial": 0,
"temperaturaSerial": 0,
"odometroExato": 36.0,
"acessorios": [
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 2,
    "valor": 59.2,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 4,
    "indice": 2,
    "valor": 60.0,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 3,
    "indice": 1,
    "valor": -34.7,
    "local": 2
  },
  {
    "numeroSerial": "",
    "tipo": 4,
    "indice": 1,
    "valor": 75.0,
    "local": 2
  }
]
}</return>
</ns0:obterPacotePosicaoPorRangeJSONResponse>
```

```
</S:Body>  
</S:Envelope>
```

5.16. ObterPacotePosicoesRFNacional

Descrição

Método para consultar a lista de pacotes de posições de rastreadores de cargas.

Declaração do Método

```
List obterPacotePosicoesRFNacional(String usuario, String senha, Integer quantidade)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
ccid	String	Id do SIM Card.
cidade	String	Cidade.
dataPacote	Calendar	Data do GPS (GMT) no momento do envio da posição.
dataPosicao	Calendar	Data da gravação da posição no servidor.
idPacote	Integer	Identificador único do pacote de posição
idVeiculo	Integer	Código do veículo.
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: 0 — Não houve jamming 1 — Ocorreu jamming
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.

Nome	Tipo	Descrição
rf	Integer	Informa se o RF do equipamento está ligado: • 0 — Não está ligado • 1 — Está ligado
rua	String	Endereço.
uf	String	Unidade da Federação.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterPacotePosicoesRFNacional>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade/>
    </web:obterPacotePosicoesRFNacional>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Header><work:WorkContext
  xmlns:work="http://oracle.com/weblogic/soap/workarear"
  >
  XYZ</work:WorkContext></S:Header><S:Body><ns0:obterPacotePosicoesRFNacionalResponse
  xmlns:nse="http://webservice.web.integracao.sascar.com.br"
  ><return><ccid>89551020226000238751</ccid><cidade>Sao Bernardo do Campo</cidade><detaPacote>2017-03-
  17 T 16:60:25</dataPacote><dataPosicao>2017-03-17T16:60:25</dataPosicao><idPacote>
  9547614</idPacote><idVeiculo>622828</idVeiculo><jamming>0</jamming><latitude>-
  23.6524171586162</latitude><longitude>-46.5734561983093</longitude><rf>1</rf><rua>Avenida Bispo Cesar
  Dacorso Filho</rua><uf>
  SP</uf></return></ns0:obterPacotePosicoesRFNacionalResponse></S:Body></S:Envelope></detaPacote>
```

5.17. getPositionsPacketJSON

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento. As informações retornadas são as mesmas do método obterPacotePosicoesJSON, porém traduzido para o inglês e mostrando as datas em UTC-0.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List getPositionsPacketJSON(String user, String password, Integer quantity)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
user	String	Sim	Login do integrador para acesso ao sistema.
password	String	Sim	Senha do integrador para acesso ao sistema.
quantity	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
vehicleId	Integer	Código do veículo.
positionDateUtc	Calendar	Data da gravação da posição no servidor, em UTC-0.
packetDateUtc	Calendar	Data do GPS no momento do envio da posição, em UTC-0.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.

Nome	Tipo	Descrição
direction	Integer	Direção do veículo: • 0 — N - Norte • 1 — NE - Nordeste • 2 — L - Leste • 3 — SE - Sudeste • 4 — S - Sul • 5 — SO - Sudoeste • 6 — O - Oeste • 7 — NO - Noroeste
speed	Integer	Velocidade em km/h obtida do GPS. Caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignition	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
odometer	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
horimeter	Integer	Horímetro atual do veículo, em minutos. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
power	Integer	Valor de tensão da bateria, em Volts.
output1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
output2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: 240 (Código de sirene informando que está ativada); -240 (Código de sirene informando que está desativada); 0 (Nenhum dispositivo).
output3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

Nome	Tipo	Descrição
input1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satellite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memory	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
referenceId	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
blocking	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado

Nome	Tipo	Descrição
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
state	String	Unidade da Federação.
city	String	Cidade.
street	String	Endereço.
country	String	País.
referencePoint	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
referenceAngle	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
referenceDistance	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperature1	Integer	Dados do sensor de temperatura 1, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature2	Integer	Dados do sensor de temperatura 2, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature3	Integer	Dados do sensor de temperatura 3, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
output5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

Nome	Tipo	Descrição
output8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entryPoint	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
exitPoint	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
macroCode	Integer	Código da macro.
messageName	String	Nome da mensagem tipo macro
messageContent	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: _João da Silva_.</i>
messageText	String	Conteúdo da mensagem tipo texto livre.

Nome	Tipo	Descrição
mdtType	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
sequencingEvent	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: eventId (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); time (Integer) (Duração do alarme)
events	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: code (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
packetId	Integer	Identificador único do pacote de posição.
integratorId	Integer	Identificador único da integradora.
driverId	Integer	Retorna sempre null.
driverName	String	Retorna sempre null.
fuelLevel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
lithometer	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litmetro estiver indisponível no veículo.
windshieldWiperState	Integer	Retorna sempre null.

Nome	Tipo	Descrição
telemetryEvents	Array<int>	Lista de evento(s) de telemetria composta pelos campos: • idEvent — Identificador do Evento • durationTime — Duração do Evento em segundos • maximumSpeedEvent — Velocidade Máxima atingida no Evento • speedReference — Velocidade de Referência
humiditySerial	Integer	Informação do sensor de umidade em porcentagem multiplicado por 10. Exemplo: 534 = 53,4%. Não aparece caso não esteja instalado.
temperatureSerial	Integer	Informação do sensor de temperatura em graus Celsius multiplicado por 10. Exemplo: 248 = 24,8°C Não aparece caso não esteja instalado.
acessorios	Json	Esse campo retorna a lista de acessórios vinculados ao componente "Rede de Acessórios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físcas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALÓGICO 6 INDEFINIDO 65.535

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getPositionsPacketJSON>
      <user>Usuário</user>
      <password>Senha</password>
    </web:getPositionsPacketJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

```

    <!--Optional:-->
    <quantity?>/quantity>
  </web:getPositionsPacketJSON>
</soapenv:Body>
</soapenv:Envelope>

```

Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:getPositionsPacketJSONResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>{
        "vehicleId": 1126486,
        "positionDateUtc": "2026-08-10 03:00:00.0",
        "packetDateUtc": "2026-08-10 03:00:00.0",
        "latitude": -25.449528,
        "longitude": -49.288528,
        "direction": 0,
        "speed": 0,
        "ignition": 1,
        "odometer": 36,
        "horimeter": 6570,
        "power": 10,
        "output1": 0,
        "output2": -240,
        "output3": 0,
        "output4": 0,
        "input1": 0,
        "input2": 0,
        "input3": 0,
        "input4": 0,
        "satellite": 0,
        "memory": 0,
        "referenceId": 0,
        "blocking": 0,
        "gps": 0,
        "state": "PR",
        "city": "Curitiba",
        "country": "BR",
        "street": "Av Iguacu",
        "referencePoint": "",
        "referenceAngle": 0,
        "referenceDistance": 0,
        "rpm": 0,
        "temperature1": 32,
        "temperature2": -125,
        "temperature3": -125,
        "output5": 0,
        "output6": 0,
        "output7": 0,
        "output8": 0,
        "input5": -247,
        "input6": -248,
        "input7": -231,
        "input8": 0,
        "entryPoint": 0,
        "exitPoint": 0,
        "macroCode": 0,
        "messageName": "",
        "messageContent": "",
        "messageText": "",
        "mdtType": 0,
      }
    </ns0:getPositionsPacketJSONResponse>
  </S:Body>
</S:Envelope>

```

```
"sequencingEvent": [
],
"events": [
  {
    "code": -247
  },
  {
    "code": -240
  }
],
"jamming": 0,
"anchorStatus": 0,
"packetId": 104648958,
"integratorId": 80,
"driverId": null,
"driverName": "null",
"fuelLevel": 0,
"lithometer": 0.0,
"windshieldWiperState": null,
"accessories": [
  {
    "serialNumber": "",
    "type": 3,
    "index": 2,
    "value": 59.2,
    "location": 2
  },
  {
    "serialNumber": "",
    "type": 4,
    "index": 2,
    "value": 60.0,
    "location": 2
  },
  {
    "serialNumber": "",
    "type": 3,
    "index": 1,
    "value": -34.7,
    "location": 2
  },
  {
    "serialNumber": "",
    "type": 4,
    "index": 1,
    "value": 75.0,
    "location": 2
  }
],
"humiditySerial": 0,
"temperatureSerial": 0,
"accurateOdometer": 36.0
}</return>
</ns0:getPositionsPacketJSONResponse>
</S:Body>
</S:Envelope>
```

5.18. getDriverPositionPacketJSON

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos. Além de trazer as informações que o método ObterPacotePosicoesJSON trazia, esse método traz também a informação de Motorista(ID e Nome) e o Status do Limpador de Parabrisa, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento. As informações retornadas são as mesmas do método obterPacotePosicoesMotoristaJSON, porém traduzido para o inglês e mostrando as datas em UTC-0.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List getDriverPositionPacketJSON(String user, String password, Integer quantity)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
user	String	Sim	Login do integrador para acesso ao sistema.
password	String	Sim	Senha do integrador para acesso ao sistema.
quantity	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
vehicleId	Integer	Código do veículo.
positionDateUtc	Calendar	Data da gravação da posição no servidor, em UTC-0.
packetDateUtc	Calendar	Data do GPS no momento do envio da posição, em UTC-0.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.

Nome	Tipo	Descrição
direction	Integer	Direção do veículo: • 0 — N - Norte • 1 — NE - Nordeste • 2 — L - Leste • 3 — SE - Sudeste • 4 — S - Sul • 5 — SO - Sudoeste • 6 — O - Oeste • 7 — NO - Noroeste
speed	Integer	Velocidade em km/h obtida do GPS. Caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignition	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
odometer	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
horimeter	Integer	Horímetro atual do veículo, em minutos. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
power	Integer	Valor de tensão da bateria, em Volts.
output1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
output2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: 240 (Código de sirene informando que está ativada); -240 (Código de sirene informando que está desativada); 0 (Nenhum dispositivo).
output3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

Nome	Tipo	Descrição
input1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satellite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memory	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
referenceId	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
blocking	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado

Nome	Tipo	Descrição
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
state	String	Unidade da Federação.
city	String	Cidade.
street	String	Endereço.
country	String	País.
referencePoint	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
referenceAngle	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
referenceDistance	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperature1	Integer	Dados do sensor de temperatura 1, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature2	Integer	Dados do sensor de temperatura 2, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature3	Integer	Dados do sensor de temperatura 3, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
output5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

Nome	Tipo	Descrição
output8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entryPoint	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
exitPoint	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
macroCode	Integer	Código da macro.
messageName	String	Nome da mensagem tipo macro
messageContent	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: _João da Silva_.</i>
messageText	String	Conteúdo da mensagem tipo texto livre.

Nome	Tipo	Descrição
mdtType	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
sequencingEvent	EventoSeq uenciament to[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: eventId (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); time (Integer) (Duração do alarme)
events	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: code (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
packetId	Integer	Identificador único do pacote de posição
integratorId	Integer	Identificador único da integradora.
driverId	Integer	Identificador do Motorista.
driverName	String	Nome do Motorista logado.
fuelLevel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
lithometer	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litrometro estiver indisponível no veículo.
windshieldWiperState	Integer	Status do Limpador de para-brisa: • 0 — Limpador Não acionado • 1 — Limpador Acionado

Nome	Tipo	Descrição
telemetryEvents	Array<int>	Lista de evento(s) de telemetria composta pelos campos: • idEvent — Identificador do Evento • durationTime — Duração do Evento em segundos • maximumSpeedEvent — Velocidade Máxima atingida no Evento • speedReference — Velocidade de Referência
humiditySerial	Integer	Informação do sensor de umidade em porcentagem multiplicado por 10. Exemplo: 534 = 53,4%. Não aparece caso não esteja instalado.
temperatureSerial	Integer	Informação do sensor de temperatura em graus Celsius multiplicado por 10. Exemplo: 248 = 24,8°C. Não aparece caso não esteja instalado.
acessorios	Json	Esse campo retorna a lista de acessórios vinculados ao componente "Rede de Acessórios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físicas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALÓGICO 6 INDEFINIDO 65.535

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getDriverPositionPacketJSON>
      <user>Usuário</user>
      <password>Senha</password>
    </web:getDriverPositionPacketJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

```

    <!--Optional:-->
    <quantity>?</quantity>
  </web:getDriverPositionPacketJSON>
</soapenv:Body>
</soapenv:Envelope>

```

Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:getDriverPositionPacketJSONResponse
      xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>{
        "vehicleId": 1126486,
        "positionDateUtc": "2026-08-10 03:03:00.0",
        "packetDateUtc": "2026-08-10 03:03:00.0",
        "latitude": -25.449528,
        "longitude": -49.288528,
        "direction": 0,
        "speed": 0,
        "ignition": 1,
        "odometer": 36,
        "horimeter": 6570,
        "power": 10,
        "output1": 0,
        "output2": -240,
        "output3": 0,
        "output4": 0,
        "input1": 0,
        "input2": 0,
        "input3": 0,
        "input4": 0,
        "satellite": 0,
        "memory": 0,
        "referenceId": 0,
        "blocking": 0,
        "gps": 0,
        "state": "PR",
        "city": "Curitiba",
        "country": "BR",
        "street": "Av Iguacu",
        "referencePoint": "",
        "referenceAngle": 0,
        "referenceDistance": 0,
        "rpm": 0,
        "temperature1": 32,
        "temperature2": -125,
        "temperature3": -125,
        "output5": 0,
        "output6": 0,
        "output7": 0,
        "output8": 0,
        "input5": -247,
        "input6": -248,
        "input7": -231,
        "input8": 0,
        "entryPoint": 0,
        "exitPoint": 0,
        "macroCode": 0,
        "messageName": "",
        "messageContent": "",
        "messageText": "",
      }
    </ns0:getDriverPositionPacketJSONResponse>
  </S:Body>
</S:Envelope>

```

```
"mdtType": 0,
"sequencingEvent": [
],
"events": [
],
"jamming": 0,
"anchorStatus": 0,
"packetId": 104648978,
"integratorId": 80,
"driverId": 0,
"driverName": "",
"fuelLevel": 0,
"lithometer": 0.0,
"windshieldWiperState": 0,
"accessories": [
  {
    "serialNumber": "",
    "type": 3,
    "index": 2,
    "value": 59.2,
    "location": 2
  },
  {
    "serialNumber": "",
    "type": 4,
    "index": 2,
    "value": 60.0,
    "location": 2
  },
  {
    "serialNumber": "",
    "type": 3,
    "index": 1,
    "value": -34.7,
    "location": 2
  },
  {
    "serialNumber": "",
    "type": 4,
    "index": 1,
    "value": 75.0,
    "location": 2
  }
],
"telemetryEvents": [
],
"humiditySerial": 0,
"temperatureSerial": 0,
"accurateOdometer": 36.0
}</return>
</ns0:getDriverPositionPacketJSONResponse>
</S:Body>
</S:Envelope>
```

5.19. getPositionPacketByRangeJSON

Descrição

Método que permite obter um ou mais pacotes de posições específicos, através do identificador único de cada pacote. Este método retorna as mesmas informações do método ObterPacotePosicoes, porém em inglês e com as datas em UTC-0.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List getPositionPacketByRangeJSON(String user, String password, Integer startId, Integer endId, Integer quantity)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
user	String	Sim	Login do integrador para acesso ao sistema.
password	String	Sim	Senha do integrador para acesso ao sistema.
startId	Integer	Sim	Identificador do pacote de posições a partir do qual se deseja que o web service reenvie.
endId	Integer	Sim	Identificador do pacote de posições a partir até o qual se deseja que o web service reenvie.
quantity	Integer	Sim	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.



Caso se deseje receber apenas um pacote de posições, deve-se informar o mesmo startId e endId.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
vehicleId	Integer	Código do veículo.

Nome	Tipo	Descrição
positionDateUtc	Calendar	Data da gravação da posição no servidor, em UTC-0.
packetDateUtc	Calendar	Data do GPS no momento do envio da posição, em UTC-0.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.
direction	Integer	Direção do veículo: • 0 — N - Norte • 1 — NE - Nordeste • 2 — L - Leste • 3 — SE - Sudeste • 4 — S - Sul • 5 — SO - Sudoeste • 6 — O - Oeste • 7 — NO - Noroeste
speed	Integer	Velocidade em km/h obtida do GPS. Caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignition	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
odometer	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
horimeter	Integer	Horímetro atual do veículo, em minutos. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
power	Integer	Valor de tensão da bateria, em Volts.
output1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
output2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: 240 (Código de sirene informando que está ativada); -240 (Código de sirene informando que está desativada); 0 (Nenhum dispositivo).

Nome	Tipo	Descrição
output3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satellite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memory	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória

Nome	Tipo	Descrição
referenceId	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
blocking	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
state	String	Unidade da Federação.
city	String	Cidade.
street	String	Endereço.
country	String	País.
referencePoint	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
referenceAngle	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
referenceDistance	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperature1	Integer	Dados do sensor de temperatura 1, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature2	Integer	Dados do sensor de temperatura 2, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature3	Integer	Dados do sensor de temperatura 3, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
output5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

Nome	Tipo	Descrição
output6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entryPoint	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
exitPoint	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
macroCode	Integer	Código da macro.
messageName	String	Nome da mensagem tipo macro
messageContent	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: _João da Silva_.</i>

Nome	Tipo	Descrição
messageText	String	Conteúdo da mensagem tipo texto livre.
mdtType	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
sequencingEvent	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: eventId (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); time (Integer) (Duração do alarme)
events	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: code (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
packetId	Integer	Identificador único do pacote de posição
integratorId	Integer	Identificador único da integradora.
driverId	Integer	Retorna sempre null.
driverName	String	Retorna sempre null.
fuelLevel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
lithometer	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litrometro estiver indisponível no veículo.
windshieldWiperState	Integer	Retorna sempre null.

Nome	Tipo	Descrição
telemetryEvents	Array<int>	Lista de evento(s) de telemetria composta pelos campos: • idEvent — Identificador do Evento • durationTime — Duração do Evento em segundos • maximumSpeedEvent — Velocidade Máxima atingida no Evento • speedReference — Velocidade de Referência
humiditySerial	Integer	Informação do sensor de umidade em porcentagem multiplicado por 10. Exemplo: 534 = 53,4%. Não aparece caso não esteja instalado.
temperatureSerial	Integer	Informação do sensor de temperatura em graus Celsius multiplicado por 10. Exemplo: 248 = 24,8°C. Não aparece caso não esteja instalado.
acessorios	Json	Esse campo retorna a lista de acessórios vinculados ao componente "Rede de Acessórios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físicas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALÓGICO 6 INDEFINIDO 65.535

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getPositionPacketByRangeJSON>
      <user>Usuário</user>
      <password>Senha</password>
    </web:getPositionPacketByRangeJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

```

<!--Optional:-->
<startId>81336079</startId>
<!--Optional:-->
<endId/>
<!--Optional:-->
<quantity>Quantidade</quantity>
</web:getPositionPacketByRangeJSON>
</soapenv:Body>
</soapenv:Envelope>

```

Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns:getPositionPacketByRangeJSONResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        {
          "vehicleId": 1563022,
          "positionDateUtc": "2023-10-16 14:06:57.0",
          "packetDateUtc": "2023-10-16 14:06:56.0",
          "latitude": -19.948707,
          "longitude": -44.010242,
          "direction": 67,
          "speed": 11,
          "ignition": 1,
          "odometer": 323519,
          "horimeter": 191239,
          "power": 27,
          "output1": 0,
          "output2": 0,
          "output3": 0,
          "output4": 0,
          "input1": 0,
          "input2": 0,
          "input3": 0,
          "input4": 0,
          "satellite": 0,
          "memory": 0,
          "referenceId": 0,
          "blocking": 1,
          "gps": 1,
          "state": "MG",
          "city": "Belo Horizonte",
          "country": "BR",
          "street": "Av Amazonas",
          "referencePoint": "PE - Posto 040-MG (2)",
          "referenceAngle": 12,
          "referenceDistance": 1054,
          "rpm": 956,
          "temperature1": -125,
          "temperature2": -125,
          "temperature3": -125,
          "output5": 0,
          "output6": -232,
          "output7": 0,
          "output8": 0,
          "input5": 0,
          "input6": 0,
          "input7": 0,
          "input8": 0,
          "entryPoint": 0,

```

```
"exitPoint": 0,  
"macroCode": 0,  
"messageName": "",  
"messageContent": "",  
"messageText": "",  
"mdtType": 0,  
"sequencingEvent": [],  
"events": [],  
"jamming": 0,  
"anchorStatus": 0,  
"packetId": 81336079,  
"integratorId": 80,  
"driverId": 0,  
"driverName": "",  
"fuelLevel": 99,  
"lithometer": 182447.616,  
"windshieldWiperState": 0,  
"telemetryEvents": [  
  {  
    "eventId": 227,  
    "durationTime": 11  
  }  
],  
"humiditySerial": -1250,  
"temperatureSerial": -1250  
}  
</return>  
</ns0:getPositionPacketByRangeJSONResponse>  
</S:Body>  
</S:Envelope>
```

5.20. getDriverPositionPacketByRangeJSON

Descrição

Método que permite obter um ou mais pacotes de posições específicos, através do identificador único de cada pacote. Este método retorna as mesmas informações do método ObterPacotePosicoesMotorista, em inglês e apresentando as datas em UTC-0.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List getDriverPositionPacketByRangeJSON(String user, String password, Integer startId, Integer endId, Integer quantity)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
user	String	Sim	Login do integrador para acesso ao sistema.
password	String	Sim	Senha do integrador para acesso ao sistema.
startId	Integer	Sim	Identificador do pacote de posições a partir do qual se deseja que o web service reenvie.
endId	Integer	Sim	Identificador do pacote de posições a partir até o qual se deseja que o web service reenvie.
quantity	Integer	Sim	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.



Caso se deseje receber apenas um pacote de posições, deve-se informar o mesmo startId e endId.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
vehicleId	Integer	Código do veículo.

Nome	Tipo	Descrição
positionDateUtc	Calendar	Data da gravação da posição no servidor, em UTC-0.
packetDateUtc	Calendar	Data do GPS no momento do envio da posição, em UTC-0.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.
direction	Integer	Direção do veículo: • 0 — N - Norte • 1 — NE - Nordeste • 2 — L - Leste • 3 — SE - Sudeste • 4 — S - Sul • 5 — SO - Sudoeste • 6 — O - Oeste • 7 — NO - Noroeste
speed	Integer	Velocidade em km/h obtida do GPS. Caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignition	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
odometer	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
horimeter	Integer	Horímetro atual do veículo, em minutos. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
power	Integer	Valor de tensão da bateria, em Volts.
output1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
output2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: 240 (Código de sirene informando que está ativada); -240 (Código de sirene informando que está desativada); 0 (Nenhum dispositivo).

Nome	Tipo	Descrição
output3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satellite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2 Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memory	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória

Nome	Tipo	Descrição
referenceId	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
blocking	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
state	String	Unidade da Federação.
city	String	Cidade.
street	String	Endereço.
country	String	País.
referencePoint	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
referenceAngle	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
referenceDistance	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperature1	Integer	Dados do sensor de temperatura 1, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature2	Integer	Dados do sensor de temperatura 2, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature3	Integer	Dados do sensor de temperatura 3, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
output5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

Nome	Tipo	Descrição
output6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entryPoint	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
exitPoint	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
macroCode	Integer	Código da macro.
messageName	String	Nome da mensagem tipo macro
messageContent	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: _João da Silva_.</i>

Nome	Tipo	Descrição
messageText	String	Conteúdo da mensagem tipo texto livre.
mdtType	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
sequencingEvent	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: eventId (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); time (Integer) (Duração do alarme)
events	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: code (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
packetId	Integer	Identificador único do pacote de posição
integratorId	Integer	Identificador único da integradora.
driverId	Integer	Identificador do Motorista.
driverName	String	Nome do Motorista logado.
fuelLevel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
lithometer	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litrometro estiver indisponível no veículo.
windshieldWiperState	Integer	Status do Limpador de para-brisa: • 0 — Limpador Não acionado • 1 — Limpador Acionado

Nome	Tipo	Descrição
telemetryEvents	Array<int>	Lista de evento(s) de telemetria composta pelos campos: • idEvent — Identificador do Evento • durationTime — Duração do Evento em segundos • maximumSpeedEvent — Velocidade Máxima atingida no Evento • speedReference — Velocidade de Referência
humiditySerial	Integer	Informação do sensor de umidade em porcentagem multiplicado por 10. Exemplo: 534 = 53,4%. Não aparece caso não esteja instalado.
temperatureSerial	Integer	Informação do sensor de temperatura em graus Celsius multiplicado por 10. Exemplo: 248 = 24,8°C. Não aparece caso não esteja instalado.
acessorios	Json	Esse campo retorna a lista de acessórios vinculados ao componente "Rede de Acessórios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físcas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALÓGICO 6 INDEFINIDO 65.535

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getDriverPositionPacketByRangeJSON>
      <user>Usuário</user>
      <password>Senha</password>
```

```
<!--Optional:-->
<startId>81336079</startId>
<!--Optional:-->
<endId/>
<!--Optional:-->
<quantity?</quantity>
</web:getDriverPositionPacketByRangeJSON>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:getDriverPositionPacketByRangeJSONResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        {
          "vehicleId": 900757,
          "positionDateUtc": "2023-10-16 14:21:34.0",
          "packetDateUtc": "2023-10-16 14:21:41.0",
          "latitude": -25.4514864,
          "longitude": -49.2461824,
          "direction": 193,
          "speed": 2,
          "ignition": 1,
          "odometer": 561,
          "horimeter": 262214,
          "power": 11,
          "output1": 0,
          "output2": -240,
          "output3": 254,
          "output4": 253,
          "input1": 0,
          "input2": 0,
          "input3": -231,
          "input4": -247,
          "satellite": 0,
          "memory": 0,
          "referenceId": 2387650,
          "blocking": 0,
          "gps": 0,
          "state": "PR",
          "city": "Curitiba",
          "country": "BR",
          "street": "R Joao Marchesini",
          "referencePoint": "novo",
          "referenceAngle": 0,
          "referenceDistance": 0,
          "rpm": 0,
          "temperature1": -125,
          "temperature2": -125,
          "temperature3": -125,
          "output5": 245,
          "output6": 0,
          "output7": -232,
          "output8": 0,
          "input5": -248,
          "input6": -241,
          "input7": -250,
          "input8": -251,
          "entryPoint": 0,
```

```
        "exitPoint": 0,  
        "macroCode": 0,  
        "messageName": "",  
        "messageContent": "",  
        "messageText": "",  
        "mdtType": 0,  
        "sequencingEvent": [],  
        "events": [],  
        "jamming": 0,  
        "anchorStatus": 0,  
        "packetId": 81336320,  
        "integratorId": 80,  
        "driverId": 3539390,  
        "driverName": "Jean Santos",  
        "fuelLevel": 0,  
        "lithometer": 0,  
        "windshieldWiperState": 0,  
        "telemetryEvents": [],  
        "humiditySerial": 0,  
        "temperatureSerial": 0  
    }  
    </return>  
</ns0:getDriverPositionPacketByRangeJSONResponse>  
</S:Body>  
</S:Envelope>
```


5.21. getPositionPacketWithLicensePlateJSON

Descrição

Método para consultar a lista de pacotes de posições enviados pelos veículos, sendo o seu conteúdo composto pelos seguintes blocos de informações: Pacote de posição; Pacote de mensagens (todos os tipos); Pacote de eventos; Pacote Alarme Sequenciamento.

Baseado no método obterPacotePosicoesJSONComPlaca, apresenta as informações em inglês e com as datas em UTC-0.



Caso cliente esteja com restrição, não será possível consumir pacotes desse período, o mesmo poderá ser obtido no método obterPacotePosicoesRestricao.

Declaração do Método

```
List getPositionPacketWithLicensePlateJSON(String user, String password, Integer quantity)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
user	String	Sim	Login do integrador para acesso ao sistema.
password	String	Sim	Senha do integrador para acesso ao sistema.
quantity	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão.

Retorno

Retorna uma lista de objetos do tipo 'PacotePosicao', sendo o retorno dos dados mais antigos para os mais novos (D-1 e dia atual), com as propriedades abaixo:

Nome	Tipo	Descrição
vehicleId	Integer	Código do veículo.
positionDateUtc	Calendar	Data da gravação da posição no servidor, em UTC-0.
packetDateUtc	Calendar	Data do GPS no momento do envio da posição, em UTC-0.
latitude	Double	Latitude em formato decimal.
longitude	Double	Longitude em formato decimal.

Nome	Tipo	Descrição
direction	Integer	Direção do veículo: • 0 — N - Norte • 1 — NE - Nordeste • 2 — L - Leste • 3 — SE - Sudeste • 4 — S - Sul • 5 — SO - Sudoeste • 6 — O - Oeste • 7 — NO - Noroeste
speed	Integer	Velocidade em km/h obtida do GPS. Caso o veículo possua Telemetria, a velocidade será a da Telemetria.
ignition	Integer	Status da ignição: • 1 — Ligada • 0 — Desligada
odometer	Integer	Odômetro obtido do GPS, caso o veículo possua Telemetria, o GPS será o da Telemetria.
horimeter	Integer	Horímetro atual do veículo, em minutos. Quando o campo "SATELITE", for igual a 1 ou 2 esse campo virá = 0 (zero) para tecnologias MSC. O valor corrente do horímetro será enviado apenas quando "Satelite=0", informando que a mensagem foi enviada via GPRS.
power	Integer	Valor de tensão da bateria, em Volts.
output1	Integer	Status do dispositivo instalado na saída 1, que por padrão da Sascar é o bloqueio do equipamento.
output2	Integer	Status do dispositivo instalado na saída 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado, cujo valor positivo indica que o mesmo está ativo e valor negativo indica que está inativo. Por exemplo: 240 (Código de sirene informando que está ativada); -240 (Código de sirene informando que está desativada); 0 (Nenhum dispositivo).
output3	Integer	Status do dispositivo instalado na saída 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output4	Integer	Status do dispositivo instalado na saída 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

Nome	Tipo	Descrição
input1	Integer	Status do dispositivo instalado na entrada 1. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input2	Integer	Status do dispositivo instalado na entrada 2. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input3	Integer	Status do dispositivo instalado na entrada 3. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input4	Integer	Status do dispositivo instalado na entrada 4. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
satellite	Integer	Informa a origem do pacote: • 0 — GPRS • 1 — SATELITAL • 2 — SATELITAL TEXTO LIVRE IMPORTANTE: Se o pacote for do tipo “2” (Satelital texto livre) considere apenas os campos abaixo: idVeiculo idPacote dataPosicao dataPacote latitude longitude textoMensagem satélite = 2. Os demais devem ser desconsiderados pois não são usados na transmissão da mensagem satelital com texto livre.
memory	Integer	Informa o tipo do pacote: • 0 — Online • 1 — Memória
referenceId	Integer	Identificação do ponto de referência. Esse id está relacionado com a tabela 'ponto_referencia' ou com o resultado do método ObterPontosReferencia(), cujo retorno é a lista de pontos de referência criados no SASGC e marcados como EMBARCÁVEL. Este é o id do ponto de referência EMBARCADO (dentro do equipamento, pode não ser embarcado pela gerenciadora que está consumindo o pacote).
blocking	Integer	Informa se o veículo está bloqueado ou não: • 0 — Desbloqueado • 1 — Bloqueado

Nome	Tipo	Descrição
gps	Integer	Status do sinal: • 0 — GPS inválido • 1 — GPS válido
state	String	Unidade da Federação.
city	String	Cidade.
street	String	Endereço.
country	String	País.
referencePoint	String	Este é nome do ponto de referência mais próximo cadastrado pela gerenciadora que está consumindo o pacote.
referenceAngle	Integer	Ângulo em relação ao ponto de referência (Pacote.pontoReferencia), cadastrado pela gerenciadora que está consumindo o pacote.
referenceDistance	Integer	Distância em metros do ponto de referência mais próximo, cadastrado pela gerenciadora que está consumindo o pacote.
rpm	Integer	Dados de RPM (Disponível para veículos com multisensor).
temperature1	Integer	Dados do sensor de temperatura 1, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature2	Integer	Dados do sensor de temperatura 2, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
temperature3	Integer	Dados do sensor de temperatura 3, em graus Celsius. Se apresentar o valor -125, indica que não há valor disponível ou válido para esse sensor.
output5	Integer	Status do dispositivo instalado na saída 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output6	Integer	Status do dispositivo instalado na saída 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
output7	Integer	Status do dispositivo instalado na saída 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).

Nome	Tipo	Descrição
output8	Integer	Status do dispositivo instalado na saída 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input5	Integer	Status do dispositivo instalado na entrada 5. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input6	Integer	Status do dispositivo instalado na entrada 6. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input7	Integer	Status do dispositivo instalado na entrada 7. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
input8	Integer	Status do dispositivo instalado na entrada 8. Retorna 0 (zero) se não houver dispositivo instalado ou o código do dispositivo instalado com valor positivo (dispositivo ativo) ou negativo (dispositivo inativo).
entryPoint	Integer	Informa se o veículo entrou em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Entrou no ponto
exitPoint	Integer	Informa se o veículo saiu em um ponto embarcado no veículo: • 0 — Não houve alteração • 1 — Saída do ponto
macroCode	Integer	Código da macro.
messageName	String	Nome da mensagem tipo macro
messageContent	String	Conteúdo da mensagem tipo macro, separado por underline (). <i>Por exemplo, se a macro contém os campos MOTORISTA e SENHA, o conteúdo da mensagem será apresentado da seguinte forma: _João da Silva_.</i>
messageText	String	Conteúdo da mensagem tipo texto livre.

Nome	Tipo	Descrição
mdtType	Integer	Identificação do tipo de teclado: • 0 — Não contém mensagem no pacote • 1 — TD50 • 3 — TD40 • 5 — TMCD • 7 — TMS3 • 8 — SASMDT
sequencingEvent	EventoSequenciamento[]	Lista com os alarmes de sequenciamento de macros. Cada objeto da lista será formado com as seguintes propriedades: eventId (Integer) (id do sequenciamento que está relacionado com a tabela 'sequenciamento_evento' ou com o resultado do método ObterSequenciamentoEvento()); time (Integer) (Duração do alarme)
events	Evento[]	Lista dos eventos gerados pelo equipamento. Cada objeto da lista será formado com a seguinte propriedade: code (Integer) - Código do atuador que gerou o evento, cujo valor poderá ser positivo indicando que foi ativado ou negativo indicando que foi desativado. Este código está relacionado com a tabela atuador_grupo ou com o resultado do método ObterGrupoAtuadores().
jamming	Integer	Informa ocorrência de evento de jamming pelo equipamento: • 0 — Não houve jamming • 1 — Ocorreu jamming
packetId	Integer	Identificador único do pacote de posição.
integratorId	Integer	Identificador único da integradora.
driverId	Integer	Retorna sempre null.
driverName	String	Retorna sempre null.
fuelLevel	Integer	Percentual de preenchimento do tanque de combustível no momento da captura da posição.
lithometer	Double	Quantidade de combustível em litros consumida durante a vida útil do veículo (combustível total). Retorna "-1.0" se a informação de litrometro estiver indisponível no veículo.
windshieldWiperState	Integer	Retorna sempre null.

Nome	Tipo	Descrição
telemetryEvents	Array<int>	Lista de evento(s) de telemetria composta pelos campos: • idEvent — Identificador do Evento • durationTime — Duração do Evento em segundos • maximumSpeedEvent — Velocidade Máxima atingida no Evento • speedReference — Velocidade de Referência
humiditySerial	Integer	Informação do sensor de umidade em porcentagem multiplicado por 10. Exemplo: 534 = 53,4%. Não aparece caso não esteja instalado.
temperatureSerial	Integer	Informação do sensor de temperatura em graus Celsius multiplicado por 10. Exemplo: 248 = 24,8°C Não aparece caso não esteja instalado.
licensePlate	String	Placa do veículo.
acessorios	Json	Esse campo retorna a lista de acessórios vinculados ao componente "Rede de Acessórios". A rede de acessórios é preparada para receber uma quantidade indefinida de acessórios que são adicionados ao rastreador sem a necessidade de ocupação das portas físcas. Cada um dos sensores envia os seguintes campos abaixo: Número Serial * Tipo Índice Valor Local *O campo serial virá vazio e sua implementação irá ocorrer em desenvolvimento futuro. O campo Tipo pode conter os seguintes valores: ENTRADA_DIGITAL 1 SAIDA_DIGITAL 2 TEMPERATURA 3 UMIDADE 4 TENSÃO 5 ANALÓGICO 6 INDEFINIDO 65.535

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- pacote
- evento
- mensagem_td50
- mensagem_tmcd
- teclado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getPositionPacketWithLicensePlateJSON>
```

```

    <user>Usuário</user>
    <password>Senha</password>
    <!--Optional:-->
    <quantity>aaa</quantity>
  </web:getPositionPacketWithLicensePlateJSON>
</soapenv:Body>
</soapenv:Envelope>

```

Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:getPositionPacketWithLicensePlateJSONResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return> { "vehicleId": 608581, "positionDateUtc": "2023-10-16 16:49:06.0",
"packetDateUtc": "2023-10-16 16:49:05.0", "latitude": -25.4516521, "longitude": -49.2460098,
"direction": 189, "speed": 0, "ignition": 0, "odometer": 2662, "horimeter": 1496825, "power": 11,
"output1": 0, "output2": -240, "output3": -254, "output4": -253, "input1": 0, "input2": -250,
"input3": -251, "input4": -241, "satellite": 0, "memory": 0, "referenceId": 0, "blocking": 0, "gps":
1, "state": "PR", "city": "Curitiba", "country": "BR", "street": "R Joao Marchesini",
"referencePoint": "novo", "referenceAngle": 0, "referenceDistance": 0, "rpm": 0, "temperature1": 23,
"temperature2": -125, "temperature3": -125, "output5": 0, "output6": -232, "output7": -232, "output8":
0, "input5": -247, "input6": -248, "input7": -231, "input8": 0, "entryPoint": 0, "exitPoint": 0,
"macroCode": 0, "messageName": "", "messageContent": "", "messageText": "", "mdtType": 0,
"sequencingEvent": [], "events": [], "jamming": 0, "anchorStatus": 0, "packetId": 81339207,
"integratorId": 80, "driverId": null, "driverName": "null", "fuelLevel": 0, "lithometer": 0,
"windshieldWiperState": null, "humiditySerial": -1250, "temperatureSerial": -1250, "licensePlate":
AND3007 } </return>
    </ns0:getPositionPacketWithLicensePlateJSONResponse>
  </S:Body>
</S:Envelope>

```


6. Métodos – Layout e Macros

6.1. ObterMacroTd50Tmcd

Descrição

Método para consultar o layout/formato das macros dos teclados TD50 e TMCD que estão carregadas nos veículos de uma gerenciadora.

Declaração do Método

```
List obterMacroTd50Tmcd(String usuario, String senha, TipoTeclado teclado)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
teclado	TipoTeclado	Sim	Tipo do teclado a ser consultado. Esse parâmetro é um enumerador com o tipos de teclados atendidos por este método: - TECLADO_TD50 - TECLADO_TMCD

Retorno

Retorna uma lista de objetos do tipo 'MacroTd50Tmcd' com as propriedades abaixo:

Nome	Tipo	Descrição
idMacroTd50Tmcd	Integer	Código da macro.
idVeiculo	Integer	Código do veículo que possui a macro.
nome	String	Nome da macro.
layout	String	Layout/formato da macro.
layoutDetalhado	String	Retorna detalhes de cada layout personalizado



As informações obtidas podem ser utilizadas para alimentar as tabelas 'macro_td50' ou 'macro_tmcd' da base de dados de integração de acordo com o tipo do teclado consultado. Se a macro estiver contida em mais de um veículo, será retornada diversas vezes onde somente o id do veículo é alterado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterMacroTd50Tmcd>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <tipoTeclado>TIPO_TECLADO</tipoTeclado>
    </web:obterMacroTd50Tmcd>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterMacroTd50TmcdResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <idMacroTd50Tmcd>2</idMacroTd50Tmcd>
        <idVeiculo>1363161</idVeiculo>
        <layout/>
        <layoutDetalhado/>
        <nome>BLOQUEIO IGNICAO</nome>
      </return>
    </ns0:obterMacroTd50TmcdResponse>
  </S:Body>
</S:Envelope>
```

6.2. ObterMacroTd50TmcdDetalhado

Descrição

Método para consultar o layout/formato das macros dos teclados TD50 e TMCD que estão carregadas nos veículos de uma gerenciadora.

Declaração do Método

```
List obterMacroTd50Tmcd(String usuario, String senha, TipoTeclado teclado)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
teclado	TipoTeclado	Sim	Tipo do teclado a ser consultado. Esse parâmetro é um enumerador com o tipos de teclados atendidos por este método: • TECLADO_TD50 • TECLADO_TMCD

Retorno

Retorna uma lista de objetos do tipo 'MacroTd50Tmcd', que possui a lista de layouts da macro e a lista de veículos que possui a macro, com as propriedades abaixo:

Nome	Tipo	Descrição
idMacroTd50Tmcd	Integer	Código da macro.
nome	String	Nome da macro.
listaLayout	Array	Lista de layouts da macro embarcados nos veículos da gerenciadora, composto por: • Descricao — descrição do layout • IdLayout — Identificador do layout
listaVeiculos	Array	Lista de todos os veículos que possuem a macro em questão.



As informações obtidas podem ser utilizadas para alimentar as tabelas 'macro_td50' ou 'macro_tmcd' da base de dados de integração de acordo com o tipo do teclado consultado.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterMacroTd50TmcdDetalhado>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <tipoTeclado>TIPO_TECLADO</tipoTeclado>
    </web:obterMacroTd50TmcdDetalhado>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterMacroTd50TmcdDetalhadoResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <idMacroTd50Tmcd>5004</idMacroTd50Tmcd>
        <listaLayout>
          <descricao>PADRAO</descricao>
          <idLayout>1521</idLayout>
        </listaLayout>
        <listaVeiculos>1436376</listaVeiculos>
        <nome>REI</nome>
      </return>
    </ns0:obterMacroTd50TmcdDetalhadoResponse>
  </S:Body>
</S:Envelope>
```

6.3. ObterMascaraDispositivo

Descrição

Método para consultar a relação de sensores ou atuadores inibidos de um veículo.

Declaração do Método

```
MascaraDispositivo obterMascaraDispositivo(String usuario, String senha, Integer idVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idVeiculo	Integer	Sim	Código do veículo a ser consultado.

Retorno

Retorna um objeto do tipo 'MascaraDispositivo' com a propriedade abaixo:

Nome	Tipo	Descrição
atuadores	Integer[]	Lista com os atuadores ou sensores inibidos no veículo informado como parâmetro.



As informações obtidas podem ser utilizadas para alimentar a tabela 'mascara_dispositivo' da base de dados de integração.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterMascaraDispositivos>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <idVeiculo>IDVeiculo</idVeiculo>
    </web:obterMascaraDispositivos>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope">
  <S:Body>
    <ns2:obterMascaraDispositivosResponse xmlns:ns2="http://ws.integracao.sascar.com/">
      <return>
        <atuadores>1</atuadores>
        <atuadores>2</atuadores>
        <atuadores>3</atuadores>
      </return>
    </ns2:obterMascaraDispositivosResponse>
  </S:Body>
</S:Envelope>
```

6.4. ObterMacroTD40

Descrição

Método para consultar as macros do teclado TD40 que estão carregadas nos veículos de uma gerenciadora.

Declaração do Método

```
List obterMacroTD40(String usuario, String senha, Integer idVeiculo, boolean satelital)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idVeiculo	Integer	Sim	Código do veículo que pode ser obtido através do método obterVeiculos.
satelital	Boolean	Sim	Parâmetro para consulta de equipamentos com antena satelital, sendo os seus valores: • <code>true</code> — Possuem antena satelital • <code>false</code> — Não possuem antena satelital

Retorno

Retorna uma lista de objetos do tipo 'MacroTd40' com as propriedades abaixo:

Nome	Tipo	Descrição
idMacroTd40	Integer	Código da macro enviada pelo veículo.
idVeiculo	Integer	Id do veículo que possui a macro.
Mensagem	String	Conteúdo da macro.
tipoMensagem	Integer	Tipo da mensagem: • <code>0</code> — Mensagem • <code>1</code> — Defeito



As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:

- 'mensagem_teclado' para veículos que não possuem antena satelital;

- 'mensagem_teclado_avl' para veículos que possuem antena satelital.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterMacroTd40>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <satelital>BOOLEAN</satelital>
    </web:obterMacroTd40>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope">
  <S:Body>
    <ns2:obterMacroTd40Response xmlns:ns2="http://ws.integracao.sascar.com/">
      <return>
        <idMacroTd40>1</idMacroTd40>
        <mensagem>Mensagem</mensagem>
        <tipoMensagem>1</tipoMensagem>
      </return>
    </ns2:obterMacroTd40Response>
  </S:Body>
</S:Envelope>
```


6.5. ObterLayout

Descrição

Método para consultar a relação de layouts disponíveis para serem enviados aos teclados via comando no Web Service XML-RPC.

Declaração do Método

```
List obterLayout(String usuario, String senha, TipoLayout layout)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
layout	TipoLayout	Sim	Tipo do layout a ser consultado. Esse parâmetro é um enumerador com os tipos de layouts existentes: • LAYOUT_TD40 • LAYOUT_TD50 • LAYOUT_TMCD • LAYOUT_SEQUENCIAMENTO_TD50 • LAYOUT_TMS3

Retorno

Retorna uma lista de objetos do tipo 'Layout' com as propriedades abaixo:

Nome	Tipo	Descrição
idLayout	Integer	Código interno Sascar do layout e visa permitir o envio através do comando de integração.
descricao	String	Nome do layout.
tipoTeclado	Integer	Tipo do teclado informado: • 0 — TD40 • 1 — TD50 • 2 — TMCD • 3 — Sequenciamento TD50 • 4 — TMS3

As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:



- tipo_layout_teclado
- tipo_layout_teclado_td50
- tipo_layout_teclado_tmcd
- tipo_sequenciamento_macro_td50

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterLayout>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <layout>TIPO_LAYOUT</layout>
    </web:obterLayout>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterLayoutResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <descricao>Fitel</descricao>
        <idLayout>1826</idLayout>
        <tipoTeclado>0</tipoTeclado>
      </return>
    </ns0:obterLayoutResponse>
  </S:Body>
</S:Envelope>
```

6.6. ObterLayoutDetalhado

Descrição

Método para consultar a relação de layouts disponíveis e suas respectivas macros para serem enviados aos teclados via comando no Web Service XML-RPC.

Declaração do Método

```
List obterLayoutDetalhado(String usuario, String senha, TipoLayout layout, Integer idLayout, String dataReferencia)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
layout	TipoLayout	Sim	Tipo do layout a ser consultado. Esse parâmetro é um enumerador com os tipos de layouts existentes: • LAYOUT_TD40 • LAYOUT_TD50 • LAYOUT_TMCD • LAYOUT_SEQUENCIAMENTO_TD50 • LAYOUT_TMS3
idLayout	Integer	Não	Para obter as informações pertinentes apenas à um registro de layout, enviar apenas o id do layout desejado.
dataReferencia	String	Não	Filtro utilizado para recuperar layouts que foram criados ou alterados após a data informada. Formato data: YYYY-DD-MM HH:MM:SS



Para os teclados do tipo SASMDT, utilizar os os [LAYOUT_SEQUENCIAMENTO_TD50](#).

Retorno

Retorna uma lista de objetos do tipo 'Layout' com as propriedades abaixo:

Nome	Tipo	Descrição
codigo	Integer	Código interno Sascar do layout e visa permitir o envio através do comando de integração.
descricao	String	Nome do layout.

Nome	Tipo	Descrição
macros	ArrayList	Lista de descrições de macros (lista de macros). Possui os campos: idMacro – Identificador da macro descricao – Nome da macro layout – Formato da Macro



As informações obtidas podem ser utilizadas para alimentar as seguintes tabelas da base de dados de integração:

- tipo_layout_teclado
- tipo_layout_teclado_td50

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterLayoutDetalhado>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <layout>TIPO_LAYOUT</layout>
      <!--Optional:-->
      <idLayout/>
      <!--Optional:-->
      <dataReferencia/>
    </web:obterLayoutDetalhado>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XUZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterLayoutDetalhadoResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <descricao>Fitel</descricao>
        <idLayout>1826</idLayout>
        <macros>
          <descricao>INICIO DE VIAGEM FITEL</descricao>
        </macros>
      </return>
    </ns0:obterLayoutDetalhadoResponse>
  </S:Body>
</S:Envelope>
```

6.7. ObterLayoutAcaoEmbarcadaAVD

Descrição

Método para consulta os dados cadastrais de Layouts de Ação Embarcada AVD do INTEGRADOR.

Declaração do Método

```
List obterLayoutAcaoEmbarcadaAVD(String usuario, String senha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.

Retorno

Retorna uma lista de objetos do tipo ' LayoutAcaoEmbarcadaAVD ' com as propriedades abaixo:

Nome	Tipo	Descrição
idLayoutAcaoEmbarcadaAVD	Integer	Id do layout da Ação Embarcada AVD
nome	Integer	Nome do layout da Ação Embarcada AVD

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterLayoutAcaoEmbarcadaAVD>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
    </web:obterLayoutAcaoEmbarcadaAVD>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterLayoutAcaoEmbarcadaAVDResponse
```

```
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
  <return>
    <idLayoutAcaoEmbarcadaAVD>1229</idLayoutAcaoEmbarcadaAVD>
    <nome>TEMPO_PANICO</nome>
  </return>
</ns0:obterLayoutAcaoEmbarcadaAVDResponse>
</S:Body>
</S:Envelope>
```

6.8. ObterLayoutAreaAvd

Descrição

Método para consultar os dados de Id e nome dos layouts de áreas cadastrados pelo Integrador.

Declaração do Método

```
List obterLayoutAreaAvd(String usuario, String senha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema

Retorno

Retorna uma lista de objetos do tipo 'ObterLayoutAreaAvd' com as propriedades abaixo:

Nome	Tipo	Descrição
id	Integer	Id do layout área avd
nome	String	Nome do layout da área avd

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterLayoutAreaAvd>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
    </web:obterLayoutAreaAvd>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterLayoutAreaAvdResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
```

```
<return>
  <id>1792</id>
  <nome>TREINAMENTO AREA DE RISCO</nome>
</return>
<return>
  <id>1026</id>
  <nome>Teste Area</nome>
</return>
<return>
  <id>1061</id>
  <nome>Layout areas 1</nome>
</return>
<return>
  <id>862</id>
  <nome>Layout areas 2</nome>
</return>
</ns0:obterLayoutAreaAvdResponse>
</S:Body>
</S:Envelope>
```


6.9. ObterLayoutData

Descrição

Método para consulta de id, nome, data de alteração e data de criação de Layout.

Declaração do Método

```
List obterLayoutData(String usuario, String senha, String layout)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema
layout	String	Sim	Indica o tipo do layout de teclado, podendo ser: • LAYOUT_TD40 • LAYOUT_TD50 • LAYOUT_TMCD • LAYOUT_SEQUENCIAMENTO_TD50 • LAYOUT_TMS3

Retorno

Retorna uma lista de objetos do tipo 'ObterLayoutData' com as propriedades abaixo:

Nome	Tipo	Descrição
dataAlteracao	Calendar	Data de alteração
dataCriacao	Calendar	Data de criação
descricao	String	Nome do layout
idLayout	Integer	Id do layout
TipoTeclado	Integer	Tipo de teclado utilizado 0 - TD40; 1 - TD50; 2 - TMCD; 3 - Sequenciamento TD50. 4 - TMS3

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:web="http://webservice.web.integracao.sascar.com.br/">>
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterLayoutData>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <layout/>
    </web:obterLayoutData>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterLayoutDataResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <dataAlteracao>2008-09-01 21:16:26</dataAlteracao>
        <dataCriacao>2008-09-01 21:16:26</dataCriacao>
        <descricao>Fitel</descricao>
        <idLayout>1826</idLayout>
        <tipoTeclado>0</tipoTeclado>
      </return>
    </ns0:obterLayoutDataResponse>
  </S:Body>
</S:Envelope>
```

6.10. ObterLayoutTecladoVeiculos

Descrição

Método para consulta os vínculos entre veículos e layouts de teclado pertencentes ao INTEGRADOR.

Declaração do Método

```
List obterLayoutTecladoVeiculos(String usuario, String senha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.

Retorno

Retorna uma lista de objetos do tipo 'LayoutTecladoVeiculos' com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Id do veículo.
idLayout	Integer	Id do layout de teclado.
tipoLayout	String	Indica o tipo do layout de teclado: - LAYOUT_TD40 - LAYOUT_TD50 - LAYOUT_TMCD - LAYOUT_SEQUENCIAMENTO_TD50 - LAYOUT_TMS3

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/" >
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterLayoutTecladoVeiculos>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
    </web:obterLayoutTecladoVeiculos>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterLayoutTecladoVeiculosResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/" >
      <return>
        <idLayout>1777</idLayout>
        <idVeiculo>89533</idVeiculo>
        <tipoLayout>LAYOUT_SEQUENCIAMENTO_TD50</tipoLayout>
      </return>
    </ns0:obterLayoutTecladoVeiculosResponse>
  </S:Body>
</S:Envelope>
```

6.11. ObterLayoutGrupoPontos

Descrição

Método para consulta os dados cadastrais de layouts de grupo de pontos pertencentes ao INTEGRADOR.

Declaração do Método

```
List obterLayoutGrupoPontos(String usuario, String senha)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.

Retorno

Retorna uma lista de objetos do tipo 'LayoutTecladoVeiculos' com as propriedades abaixo:

Nome	Tipo	Descrição
idLayoutGrupoPonto	Integer	Id do layout do grupo de pontos
nome	Integer	Nome do layout de grupo de pontos

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/" >
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterLayoutGrupoPontos>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
    </web:obterLayoutGrupoPontos>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterLayoutGrupoPontosResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/" >
```

```
<return>
  <idLayoutGrupoPonto>982</idLayoutGrupoPonto>
  <nome>ALMIR</nome>
</return>
</ns0:obterLayoutGrupoPontosResponse>
</S:Body>
</S:Envelope>
```

7. Métodos – Telemetria e Câmeras

7.1. ObterEventosTempoDirecao

Descrição

Método para consultar as informações de Tempo de Direção dos motoristas. Todas os eventos de Tempo de Direção enviados pelo motorista utilizando o tablet (SasMDT) podem ser acessadas através desse método. Além do evento atual, o último evento, latitude, longitude, motorista reserva, odômetro e endereço (rua, cidade e UF) também são exibidos no retorno do método.

Na integração, a implementação para se entender o **INÍCIO de JORNADA** deve ser em conjunto com o evento anterior **ENCERRAR**. Esse par de eventos é que irá determinar o fim de uma jornada e o início da próxima.



Durante a jornada, vai ocorrer essa alteração entre evento de **JORNADA** e **PARADO**, pois toda vez que encerra um evento, como DIRIGINDO, ESPERA ou REFEIÇÃO, ele volta para a JORNADA/PARADO, que são os eventos de ID 1 e 4 respectivamente.

Então a implementação pode ser considerada algo semelhante a:

- **Evento de JORNADA com evento anterior de ENCERRAR** = Início efetivo da jornada.
- **Evento de JORNADA com qualquer outro evento anterior** = Faz parte da jornada ativa e pode ser considerado como o evento PARADO, pois tem a mesma função.

Declaração do Método

```
List obterEventosTempoDirecao(String usuario, String senha, Integer quantidade, Integer idMotorista, String dataInicio, String dataFim)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do usuário para acesso ao sistema.
senha	String	Sim	Senha do usuário para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão. Para obter o restante dos registros, o INTEGRADOR deve refazer a requisição utilizando como data de início do parâmetro de entrada a última data retornada no campo dataInicio do parâmetro de saída.

Nome	Tipo	Obrigatório	Descrição
idMotorista	Integer	Não	ID do Motorista desejado.
dataInicio	String	Sim	Data inicial do período desejado.
dataFim	String	Sim	Data final do período desejado.

Retorno

Retorna uma lista de objetos do tipo 'EventoTempoDirecao' com as propriedades abaixo:

Nome	Tipo	Descrição
cidade	String	Descrição da cidade do endereço da posição do evento.
dataInicio	String	Data do evento (macro).
descricaoEventoTempoDirecao	String	Descrição do evento atual.
descricaoEventoTempoDirecaoAnterior	String	Descrição do evento anterior.
eventoTempoDirecao	Integer	Identificação do evento atual.
eventoTempoDirecaoAnterior	Integer	Identificação do evento anterior.
idCliente	Integer	Identificação do cliente.
idMotorista	Integer	Identificação do motorista principal.
idMotoristaReserva	Integer	Identificação do motorista reserva.
idVeiculo	Integer	Identificação do veículo.
latitude	Double	Latitude da posição do evento.
longitude	Double	Longitude da posição do evento.
nomeCliente	String	Nome do cliente.
nomeMotorista	String	Nome do motorista principal.
nomeMotoristaReserva	String	Nome do motorista reserva.
odometro	Integer	Odômetro da posição do evento.
placa	String	Placa do veículo.
rua	String	Descrição da rua do endereço da posição do evento.
uf	String	UF do endereço da posição do evento.

Relação Evento Descrição

Id do Evento	Descrição
1	Jornada
2	Dirigindo
3	Pausa
4	Parada
5	Refeição
6	Esperar
7	Encerrar
8	Trocar

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterEventosTempoDirecao>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>?</quantidade>
      <!--Optional:-->
      <idMotorista>?</idMotorista>
      <dataInicio>2023-10-10 00:00:00</dataInicio>
      <dataFim>2023-10-10 23:59:00</dataFim>
    </web:obterEventosTempoDirecao>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterEventosTempoDirecaoResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <cidade>Mogi Guacu</cidade>
        <dataInicio>2023-10-10 01:09:53</dataInicio>
        <descricaoEventoTempoDirecao>Jornada</descricaoEventoTempoDirecao>
        <descricaoEventoTempoDirecaoAnterior>Encerrar</descricaoEventoTempoDirecaoAnterior>
        <eventoTempoDirecao>1</eventoTempoDirecao>
        <eventoTempoDirecaoAnterior>7</eventoTempoDirecaoAnterior>
        <idCliente>IDCLIENTE</idCliente>
        <idMotorista>IDMOTORISTA</idMotorista>
        <idMotoristaReserva>0</idMotoristaReserva>
        <idVeiculo>IDVEICULO</idVeiculo>
        <latitude>-22.3563471</latitude>
        <longitude>-46.9690173</longitude>
        <nomeCliente>NOME CLIENTE</nomeCliente>
        <nomeMotorista>NOME MOTORISTA</nomeMotorista>
        <odometro>281420</odometro>
        <placa>PLACA-VEICULO</placa>
        <rua>Sem nome</rua>
        <uf>SP</uf>
      </return>
    </ns0:obterEventosTempoDirecaoResponse>
  </S:Body>
</S:Envelope>
```

7.2. ObterEventosTempoDirecaoDataChegada

Descrição

Método para consultar as informações de Tempo de Direção dos motoristas contendo a data de chegada inicial e data chegada final do pacote de posição. Todas os eventos de Tempo de Direção enviado pelo motorista utilizando o tablet (SasMDT) podem ser acessadas através desse método. Além do evento atual, o último evento, latitude, longitude, motorista reserva, odômetro e endereço (rua, cidade e UF) também são exibidos no retorno do método.

Na integração, a implementação para se entender o **INÍCIO de JORNADA** deve ser em conjunto com o evento anterior **ENCERRAR**. Esse par de eventos é que irá determinar o fim de uma jornada e o início da próxima.



Durante a jornada, vai ocorrer essa alteração entre evento de **JORNADA** e **PARADO**, pois toda vez que encerra um evento, como DIRIGINDO, ESPERA ou REFEIÇÃO, ele volta para a JORNADA/PARADO, que são os eventos de ID 1 e 4 respectivamente.

Então a implementação pode ser considerada algo semelhante a:

- **Evento de JORNADA com evento anterior de ENCERRAR** = Início efetivo da jornada.
- **Evento de JORNADA com qualquer outro evento anterior** = Faz parte da jornada ativa e pode ser considerado como o evento PARADO, pois tem a mesma função.

Declaração do Método

```
List obterEventosTempoDirecaoDataChegada(String usuario, String senha, Integer quantidade, Integer idMotorista, String dataInicio, String dataFim, String dataChegadaInicial, String dataChegadaFinal)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do usuário para acesso ao sistema.
senha	String	Sim	Senha do usuário para acesso ao sistema.
quantidade	Integer	Não	Limite de registros como resultado da consulta. O sistema possui um limite padrão de 3000 registros, que será aplicado nos seguintes casos: Quando o cliente não informar esse parâmetro; Quando o cliente informar um número maior que o limite padrão. Para obter o restante dos registros, o INTEGRADOR deve refazer a requisição utilizando como data de início do parâmetro de entrada a última data retornada no campo dataInicio do parâmetro de saída.
idMotorista	Integer	Não	ID do Motorista desejado.

Nome	Tipo	Obrigatório	Descrição
dataInicio	String	Não	Data inicial do período desejado.
dataFim	String	Não	Data final do período desejado.
dataChegadaInicial	String	Sim	Data de chegada inicial do pacote de posição.
dataChegadaFinal	String	Sim	Data de chegada final do pacote de posição.

Retorno

Retorna uma lista de objetos do tipo 'EventoTempoDirecaoDataChegada' com as propriedades abaixo:

Nome	Tipo	Descrição
cidade	String	Descrição da cidade do endereço da posição do evento.
dataInicio	String	Data do evento (macro).
descricaoEventoTempoDirecao	String	Descrição do evento atual.
descricaoEventoTempoDirecaoAnterior	String	Descrição do evento anterior.
eventoTempoDirecao	Integer	Identificação do evento atual.
eventoTempoDirecaoAnterior	Integer	Identificação do evento anterior.
idCliente	Integer	Identificação do cliente.
idMotorista	Integer	Identificação do motorista principal.
idMotoristaReserva	Integer	Identificação do motorista reserva.
idVeiculo	Integer	Identificação do veículo.
latitude	Double	Latitude da posição do evento.
longitude	Double	Longitude da posição do evento.
nomeCliente	String	Nome do cliente.
nomeMotorista	String	Nome do motorista principal.
nomeMotoristaReserva	String	Nome do motorista reserva.
odometro	Integer	Odômetro da posição do evento.
placa	String	Placa do veículo.
rua	String	Descrição da rua do endereço da posição do evento.
uf	String	UF do endereço da posição do evento.
dataChegada	String	Data da chegada do pacote de posição.

Relação Evento Descrição

Id do Evento	Descrição
1	Jornada
2	Dirigindo
3	Pausa
4	Parada
5	Refeição
6	Esperar
7	Encerrar
8	Trocar

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterEventosTempoDirecaoDataChegada>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>?</quantidade>
      <!--Optional:-->
      <idMotorista>?</idMotorista>
      <!--Optional:-->
      <dataInicio>2023-10-17 00:00:00</dataInicio>
      <!--Optional:-->
      <dataFim>2023-10-17 23:59:00</dataFim>
      <!--Optional:-->
      <dataChegadaInicial>2023-10-17 00:00:00</dataChegadaInicial>
      <!--Optional:-->
      <dataChegadaFinal>2023-10-17 00:00:00</dataChegadaFinal>
    </web:obterEventosTempoDirecaoDataChegada>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterEventosTempoDirecaoDataChegadaResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <cidade>Mogi Guacu</cidade>
        <dataInicio>2023-10-17 01:24:54</dataInicio>
        <descricaoEventoTempoDirecao>Jornada</descricaoEventoTempoDirecao>
        <descricaoEventoTempoDirecaoAnterior>Dirigindo</descricaoEventoTempoDirecaoAnterior>
        <eventoTempoDirecao>1</eventoTempoDirecao>
        <eventoTempoDirecaoAnterior>2</eventoTempoDirecaoAnterior>
        <idCliente>412543</idCliente>
        <idMotorista>2741963</idMotorista>
        <idMotoristaReserva>0</idMotoristaReserva>
        <idVeiculo>1731832</idVeiculo>
        <latitude>-22.2996305</latitude>
        <longitude>-46.9839418</longitude>
        <nomeCliente>ECO FORTE BIOENERGIA LTDA</nomeCliente>
        <nomeMotorista>GABRIEL APARECIDO ROMAO FILHO</nomeMotorista>
        <odometro>285609</odometro>
        <placa>FDR9I01-2</placa>
        <rua>Rodovia Vicinal Vice-Governador Almino Monteiro Alvares Afonso</rua>
        <uf>SP</uf>
        <dataChegada>2023-10-17 01:24:56</dataChegada>
      </return>
    </ns0:obterEventosTempoDirecaoDataChegadaResponse>
  </S:Body>
</S:Envelope>
```

7.3. ObterDeltaTelemetriaIntegracaoInercia



Necessário possuir Obrigação Financeira de Software para consumo dos Deltas.

Descrição

Serviço responsável por disponibilizar aos clientes os Deltas de Telemetria, respeitando as regras de negócio previamente estabelecidas ao INTEGRADOR.

Declaração do Método

```
List obterDeltaTelemetriaIntegracaoInercia(String usuario, String senha, Integer idVeiculo, String dataInicio, String dataFinal)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idVeiculo	Integer	Sim	Id do veículo
dataInicio	String	Sim	Data inicial para a consulta Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataFinal	String	Sim	Data final para a consulta Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.

Retorno

Retorna uma lista de objetos do tipo Delta com as propriedades abaixo:

Nome	Tipo	Descrição
idVeiculo	Integer	Id do veículo
idMotorista	Integer	Id do motorista
loginMotorista	Integer	Login do motorista
nomeMotorista	String	Nome Motorista
latitude	Integer	Latitude
longitude	Integer	Longitude
dataPosicao	String	Data do GPS (GMT-0) no momento do envio do pacote

Nome	Tipo	Descrição
tempoDuracaoGiroMotor	Integer	Tempo em segundos(s) que houve giro de motor
tempoDuracaoTotal	Integer	Tempo em segundos(s) decorrido do início ao fechamento do delta
tempoDuracaoMovimento	Integer	Tempo em segundos(s) que o veículo se movimentou
tempoDuracaoParado	Integer	Tempo em segundos(s) que o veículo ficou parado
velocidadeMaximaFaixaAmarela	Integer	Velocidade máxima em que o veículo permaneceu na faixa amarela
tipoDelta	Integer	Tipo do Delta: – Ignição ON/OFF – Login/ Logoff
tempoDuracaoFaixaMarchaLenta	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de marcha lenta
tempoDuracaoFaixaMarchaLentaComVelocidade	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de marcha lenta estando em velocidade.
tempoDuracaoFaixaDeTransicaoComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de transição com inércia.
tempoDuracaoFaixaDeTransicaoSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de transição sem inércia.
tempoDuracaoFaixaVerdeEconomicaComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa verde econômica com inércia.
tempoDuracaoFaixaVerdeEconomicaSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa verde econômica sem inércia.
tempoDuracaoFaixaVerdeComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa verde com inércia.
tempoDuracaoFaixaVerdeSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa verde sem inércia.
tempoDuracaoFaixaAmarelaSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa amarela com inércia.
tempoDuracaoFaixaAmarelaComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa amarela sem inércia.
tempoDuracaoFaixaDePerigoComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de perigo com inércia.
tempoDuracaoFaixaDePerigoSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de perigo sem inércia.
horimetro	Integer	Horímetro final (no momento do fechamento do delta)
odometro	Integer	Hodômetro final (no momento do fechamento do delta)
distanciaPercorrida	Integer	Distância percorrida (da abertura ao fechamento do delta)
velocidadeMedia	Integer	Velocidade média
rpmMaximo	Integer	RPM máximo atingido entre abertura e fechamento de delta

Nome	Tipo	Descrição
rpmMedia	Integer	Média de RPM (entre abertura e fechamento de delta)
tempoDuracaoFreioMotor	Integer	Tempo de acionamento (em segundos) do freio motor
distanciaPercorridaEmbreagemA cionada	Integer	Distância (em km) que o veículo percorreu com embreagem acionada
distanciaPercorridaFreioAccionado	Integer	Distância (em km) que o veículo percorreu com freio acionado
consumoCombustivel	Integer	Consumo (em mililitros) de combustível desde o início até o fim do delta. Nas situações onde o Delta de Telemetria não possui a informação de Consumo de Combustível, esse campo não será apresentado no XML de Retorno.
distPercorridaAscendenteFxAmarela	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa Amarela
distPercorridaAscendenteFxMarchaLenta	Integer	Distância (em km) percorrida pelo veículo em ascendente na Marcha Lenta
distPercorridaAscendenteFxPerigo	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa de Perigo
distPercorridaAscendenteFxTransic	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa de Transição
distPercorridaAscendenteFxVerde	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa Verde
distPercorridaAscendenteFxVerde_ext	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa Verde Econômica
distPercorridaDescendenteFxAmarela	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa Amarela
distPercorridaDescendenteFxMarchaLenta	Integer	Distância (em km) percorrida pelo veículo em descendente na Marcha Lenta
distPercorridaDescendenteFxPerigo	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa de Perigo
distPercorridaDescendenteFxTransic	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa de Transição
distPercorridaDescendenteFxVerde	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa Verde
distPercorridaDescendenteFxVerde_ext	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa Verde Econômica
distPercorridaEstavelFxAmarela	Integer	Distância (em km) percorrida pelo veículo estável na Faixa Amarela
distPercorridaEstavelFxMarchaLenta	Integer	Distância (em km) percorrida pelo veículo estável na Marcha Lenta

Nome	Tipo	Descrição
distPercorridaEstavelFxPerigo	Integer	Distância (em km) percorrida pelo veículo estável na Faixa de Perigo
distPercorridaEstavelFxTransic	Integer	Distância (em km) percorrida pelo veículo estável na Faixa de Transição
distPercorridaEstavelFxVerde	Integer	Distância (em km) percorrida pelo veículo estável na Faixa Verde
distPercorridaEstavelFxVerde_ext	Integer	Distância (em km) percorrida pelo veículo estável na Faixa Verde Econômica
distTotalFxAscendente	Integer	Distância Total (em km) percorrida pelo veículo em ascendente
distTotalFxDescendente	Integer	Distância Total (em km) percorrida pelo veículo em descendente
distTotalFxEstavel	Integer	Distância Total (em km) percorrida pelo veículo estável
tempoDuracaoAscendenteFxAmarela	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa Amarela
tempoDuracaoAscendenteFxMarchaLenta	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Marcha Lenta
tempoDuracaoAscendenteFxPerigo	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa de Perigo
tempoDuracaoAscendenteFxTransic	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa de Transição
tempoDuracaoAscendenteFxVerde	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa Verde
tempoDuracaoAscendenteFxVerde_ext	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa Verde Econômica
tempoDuracaoDescendenteFxAmarela	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa Amarela
tempoDuracaoDescendenteFxMarchaLenta	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Marcha Lenta
tempoDuracaoDescendenteFxPerigo	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa de Perigo
tempoDuracaoDescendenteFxTransic	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa de Transição
tempoDuracaoDescendenteFxVerde	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa Verde
tempoDuracaoDescendenteFxVerde_ext	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa Verde Econômica
tempoDuracaoEstavelFxAmarela	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa Amarela

Nome	Tipo	Descrição
tempoDuracaoEstavelFxMarchaLenta	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Marcha Lenta
tempoDuracaoEstavelFxPerigo	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa de Perigo
tempoDuracaoEstavelFxTransic	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa de Transição
tempoDuracaoEstavelFxVerde	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa Verde
tempoDuracaoEstavelFxVerde_ext	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa Verde Econômica
tempoTotalFxAscendente	Integer	Tempo Total em segundos(s) que o veículo permaneceu em ascendente
tempoTotalFxDescendente	Integer	Tempo Total em segundos(s) que o veículo permaneceu em descendente
tempoTotalFxEstavel	Integer	Tempo Total em segundos(s) que o veículo permaneceu estável
pressaoMediaDoOleoDoMotor	Integer	Pressão média do óleo
pressaoMediaDoSistemaDeFreioAr	Integer	Pressão Média do Sistema de Freios a Ar
temperaturaMediaDoArrefecimento	Integer	Temperatura Média de Arrefecimento
temperaturaMediaDoCombustivel	Integer	Temperatura Média do Combustível
temperaturaMediaDoOleoDoMotor	Integer	Temperatura Média do Óleo
tempoTotalComCinto	Integer	Tempo Total com Cinto
tensaoMediaDaBateria	Integer	Tensão Média da Bateria

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterDeltaTelemetriaIntegracaoInercia>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <dataInicio>2023-10-17 00:00:00</dataInicio>
      <dataFinal>2023-10-17 23:59:00</dataFinal>
      <idVeiculo>881456</idVeiculo>
    </web:obterDeltaTelemetriaIntegracaoInercia>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterDeltaTelemetriaIntegracaoInerciaResponse
      xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <consumoCombustivel>0</consumoCombustivel>
        <dataPosicao>17/10/2023 07:11:46</dataPosicao>
        <distPercorridaAscendenteFxAmarela>0</distPercorridaAscendenteFxAmarela>
        <distPercorridaAscendenteFxFxMarchaLenta>0</distPercorridaAscendenteFxFxMarchaLenta>
        <distPercorridaAscendenteFxFxPerigo>0</distPercorridaAscendenteFxFxPerigo>
        <distPercorridaAscendenteFxFxTransic>0</distPercorridaAscendenteFxFxTransic>
        <distPercorridaAscendenteFxFxVerde>0</distPercorridaAscendenteFxFxVerde>
        <distPercorridaAscendenteFxFxVerde_ext>0</distPercorridaAscendenteFxFxVerde_ext>
        <distPercorridaDescendenteFxAmarela>0</distPercorridaDescendenteFxAmarela>
        <distPercorridaDescendenteFxFxMarchaLenta>0</distPercorridaDescendenteFxFxMarchaLenta>
        <distPercorridaDescendenteFxFxPerigo>0</distPercorridaDescendenteFxFxPerigo>
        <distPercorridaDescendenteFxFxTransic>0</distPercorridaDescendenteFxFxTransic>
        <distPercorridaDescendenteFxFxVerde>0</distPercorridaDescendenteFxFxVerde>
        <distPercorridaDescendenteFxFxVerde_ext>0</distPercorridaDescendenteFxFxVerde_ext>
        <distPercorridaEstavelFxAmarela>0</distPercorridaEstavelFxAmarela>
        <distPercorridaEstavelFxFxMarchaLenta>0</distPercorridaEstavelFxFxMarchaLenta>
        <distPercorridaEstavelFxFxPerigo>0</distPercorridaEstavelFxFxPerigo>
        <distPercorridaEstavelFxFxTransic>0</distPercorridaEstavelFxFxTransic>
        <distPercorridaEstavelFxFxVerde>0</distPercorridaEstavelFxFxVerde>
        <distPercorridaEstavelFxFxVerde_ext>0</distPercorridaEstavelFxFxVerde_ext>
        <distTotalFxAscendente>0</distTotalFxAscendente>
        <distTotalFxDescendente>0</distTotalFxDescendente>
        <distTotalFxEstavel>0</distTotalFxEstavel>
        <distanciaPercorrida>0</distanciaPercorrida>
        <distanciaPercorridaEmbreagemAccionada>0</distanciaPercorridaEmbreagemAccionada>
        <distanciaPercorridaFreioAccionado>0</distanciaPercorridaFreioAccionado>
        <horimetro>535443</horimetro>
        <idMotorista>0</idMotorista>
        <idVeiculo>881456</idVeiculo>
        <latitude>-22.7543</latitude>
        <loginMotorista>0</loginMotorista>
        <longitude>-43.3877</longitude>
        <nomeMotorista>Sem motorista</nomeMotorista>
        <odometro>4294967</odometro>
        <pressaoMediaDoOleoDoMotor>0</pressaoMediaDoOleoDoMotor>
        <pressaoMediaDoSistemaDeFreioAAR>0</pressaoMediaDoSistemaDeFreioAAR>
        <rpmMaximo>0</rpmMaximo>
        <rpmMedia>0</rpmMedia>
        <temperaturaMediaDoArrefecimento>0</temperaturaMediaDoArrefecimento>
        <temperaturaMediaDoCombustivel>0</temperaturaMediaDoCombustivel>
        <temperaturaMediaDoOleoDoMotor>0</temperaturaMediaDoOleoDoMotor>
        <tempoDuracaoAscendenteFxAmarela>0</tempoDuracaoAscendenteFxAmarela>
        <tempoDuracaoAscendenteFxFxMarchaLenta>0</tempoDuracaoAscendenteFxFxMarchaLenta>
        <tempoDuracaoAscendenteFxFxPerigo>0</tempoDuracaoAscendenteFxFxPerigo>
        <tempoDuracaoAscendenteFxFxTransic>0</tempoDuracaoAscendenteFxFxTransic>
        <tempoDuracaoAscendenteFxFxVerde>0</tempoDuracaoAscendenteFxFxVerde>
        <tempoDuracaoAscendenteFxFxVerde_ext>0</tempoDuracaoAscendenteFxFxVerde_ext>
        <tempoDuracaoDescendenteFxAmarela>0</tempoDuracaoDescendenteFxAmarela>
        <tempoDuracaoDescendenteFxFxMarchaLenta>0</tempoDuracaoDescendenteFxFxMarchaLenta>
        <tempoDuracaoDescendenteFxFxPerigo>0</tempoDuracaoDescendenteFxFxPerigo>
        <tempoDuracaoDescendenteFxFxTransic>0</tempoDuracaoDescendenteFxFxTransic>
        <tempoDuracaoDescendenteFxFxVerde>0</tempoDuracaoDescendenteFxFxVerde>
        <tempoDuracaoDescendenteFxFxVerde_ext>0</tempoDuracaoDescendenteFxFxVerde_ext>
        <tempoDuracaoEstavelFxAmarela>0</tempoDuracaoEstavelFxAmarela>
        <tempoDuracaoEstavelFxFxMarchaLenta>0</tempoDuracaoEstavelFxFxMarchaLenta>
        <tempoDuracaoEstavelFxFxPerigo>0</tempoDuracaoEstavelFxFxPerigo>
        <tempoDuracaoEstavelFxFxTransic>0</tempoDuracaoEstavelFxFxTransic>
      </return>
    </ns0:obterDeltaTelemetriaIntegracaoInerciaResponse>
  </S:Body>
</S:Envelope>
```

```

<tempoDuracaoEstavelFxVerde>0</tempoDuracaoEstavelFxVerde>
<tempoDuracaoEstavelFxVerde_ext>0</tempoDuracaoEstavelFxVerde_ext>
<tempoDuracaoFaixaAmarelaComInercia>0</tempoDuracaoFaixaAmarelaComInercia>
<tempoDuracaoFaixaAmarelaSemInercia>0</tempoDuracaoFaixaAmarelaSemInercia>
<tempoDuracaoFaixaDePerigoComInercia>0</tempoDuracaoFaixaDePerigoComInercia>
<tempoDuracaoFaixaDePerigoSemInercia>0</tempoDuracaoFaixaDePerigoSemInercia>
<tempoDuracaoFaixaDeTransicaoComInercia>0</tempoDuracaoFaixaDeTransicaoComInercia>
<tempoDuracaoFaixaDeTransicaoSemInercia>0</tempoDuracaoFaixaDeTransicaoSemInercia>
<tempoDuracaoFaixaMarchaLenta>0</tempoDuracaoFaixaMarchaLenta>
<tempoDuracaoFaixaMarchaLentaComVelocidade>
0</tempoDuracaoFaixaMarchaLentaComVelocidade>
<tempoDuracaoFaixaVerdeComInercia>0</tempoDuracaoFaixaVerdeComInercia>
<tempoDuracaoFaixaVerdeEconomicaComInercia>
0</tempoDuracaoFaixaVerdeEconomicaComInercia>
<tempoDuracaoFaixaVerdeEconomicaSemInercia>
0</tempoDuracaoFaixaVerdeEconomicaSemInercia>
<tempoDuracaoFaixaVerdeSemInercia>0</tempoDuracaoFaixaVerdeSemInercia>
<tempoDuracaoFreioMotor>0</tempoDuracaoFreioMotor>
<tempoDuracaoGiroMotor>0</tempoDuracaoGiroMotor>
<tempoDuracaoMovimento>0</tempoDuracaoMovimento>
<tempoDuracaoParado>0</tempoDuracaoParado>
<tempoDuracaoTotal>0</tempoDuracaoTotal>
<tempoTotalComCinto>0</tempoTotalComCinto>
<tempoTotalFxAscendente>0</tempoTotalFxAscendente>
<tempoTotalFxDescendente>0</tempoTotalFxDescendente>
<tempoTotalFxEstavel>0</tempoTotalFxEstavel>
<tensaoMediaDaBateria>0</tensaoMediaDaBateria>
<tipoDelta>254</tipoDelta>
<velocidadeMaximaFaixaAmarela>0</velocidadeMaximaFaixaAmarela>
<velocidadeMedia>0</velocidadeMedia>
</return>
</ns0:obterDeltaTelemetriaIntegracaoInerciaResponse>
</S:Body>
</S:Envelope>

```

7.4. ObterDeltaTelemetriaIntegracaoDataChegada



Necessário possuir Obrigação Financeira de Software para consumo dos Deltas.

Descrição

Serviço responsável por disponibilizar aos clientes os Deltas de Telemetria, respeitando as regras de negócio previamente estabelecidas ao INTEGRADOR.

Declaração do Método

```
List obterDeltaTelemetriaIntegracaoDataChegada(String usuario, String senha, String dataInicio, String dataFinal, Integer idVeiculo, String dataInicioChegada, String dataFinalChegada)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
dataInicio	String	Não	Data inicial para a consulta Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataFinal	String	Não	Data final para a consulta Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
idVeiculo	Integer	Sim	Id do Veículo
dataInicioChegada	String	Sim	Data Início de registro no sistema Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataFinalChegada	String	Sim	Data final de registro no sistema Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.

Retorno

Nome	Tipo	Descrição
idVeiculo	Integer	Id do veículo
idMotorista	Integer	Id do motorista
loginMotorista	Integer	Login do motorista
latitude	Integer	Latitude

Nome	Tipo	Descrição
longitude	Integer	Longitude
dataPosicao	String	Data do GPS (GMT-0) no momento do envio do pacote
deltmDeltaGiroMotor	Integer	Tempo em segundos(s) que houve giro de motor
deltmDeltaIntervalo	Integer	Tempo em segundos(s) decorrido do início ao fechamento do delta
tempoDuracaoMovimento	Integer	Tempo em segundos(s) que o veículo se movimentou
deltmDeltaParado	Integer	Tempo em segundos(s) que o veículo ficou parado
tempoDuracaoFaixaAmarela	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa amarela
deltmFxAmareloVelMax	Integer	Velocidade máxima em que o veículo permaneceu na faixa amarela
tempoDuracaoFaixaAzul	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa azul
tempoDuracaoFaixaVerde	Integer	Tempo em segundos(s) que o veículo permaneceu na verde
tempoDuracaoFaixaVermelha	Integer	Tempo em segundos(s) que o veículo permaneceu na vermelha
tipoDelta	Integer	Tipo do Delta: – Ignição ON/OFF – Login/ Logoff
horimetro	Integer	Horímetro final (no momento do fechamento do delta)
odometro	Integer	Hodômetro final (no momento do fechamento do delta)
deltmHodometroDelta	Integer	Distância percorrida (da abertura ao fechamento do delta)
velocidadeMedia	Integer	Velocidade média
rpmMaximo	Integer	RPM máximo atingido entre abertura e fechamento de delta
rpmMedia	Integer	Média de RPM (entre abertura e fechamento de delta)
tempoAcionamentoFreioMotor	Integer	Tempo de acionamento (em segundos) do freio motor
distanciaPercorridaEmbreagemA cionada	Integer	Distância (em km) que o veículo percorreu com embreagem acionada
distanciaPercorridaFreioAccionado	Integer	Distância (em km) que o veículo percorreu com freio acionado
consumoCombustivel	Integer	Consumo (em mililitros) de combustível desde o início até o fim do delta. Nas situações onde o Delta de Telemetria não possui a informação de Consumo de Combustível, esse campo não será apresentado no XML de Retorno.
dataChegada	String	Data da chegada do pacote.
nomeMotorista	String	Nome do Motorista

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterDeltaTelemetriaIntegracaoDataChegada>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <dataInicio>2023-10-17 00:00:00</dataInicio>
      <dataFinal>2023-10-17 23:59:00</dataFinal>
      <idVeiculo>881456</idVeiculo>
      <dataInicioChegada>2023-10-17 00:00:00</dataInicioChegada>
      <dataFinalChegada>2023-10-17 00:00:00</dataFinalChegada>
    </web:obterDeltaTelemetriaIntegracaoDataChegada>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterDeltaTelemetriaIntegracaoDataChegadaResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <consumoCombustivel>0</consumoCombustivel>
        <dataPosicao>17/10/2023 07:11:46</dataPosicao>
        <distanciaPercorrida>0</distanciaPercorrida>
        <distanciaPercorridaEmbreagemAcionada>0</distanciaPercorridaEmbreagemAcionada>
        <distanciaPercorridaFreioAcionado>0</distanciaPercorridaFreioAcionado>
        <horimetro>535443</horimetro>
        <idMotorista>0</idMotorista>
        <idVeiculo>881456</idVeiculo>
        <latitude>-22.7543</latitude>
        <loginMotorista>0</loginMotorista>
        <longitude>-43.3877</longitude>
        <nomeMotorista>Sem motorista</nomeMotorista>
        <odometro>4294967</odometro>
        <rpmMaximo>0</rpmMaximo>
        <rpmMedia>0</rpmMedia>
        <tempoDuracaoFaixaAmarela>0</tempoDuracaoFaixaAmarela>
        <tempoDuracaoFaixaAzul>0</tempoDuracaoFaixaAzul>
        <tempoDuracaoFaixaVerde>0</tempoDuracaoFaixaVerde>
        <tempoDuracaoFaixaVermelha>0</tempoDuracaoFaixaVermelha>
        <tempoDuracaoFreioMotor>0</tempoDuracaoFreioMotor>
        <tempoDuracaoGiroMotor>0</tempoDuracaoGiroMotor>
        <tempoDuracaoMovimento>0</tempoDuracaoMovimento>
        <tempoDuracaoParado>0</tempoDuracaoParado>
        <tempoDuracaoTotal>0</tempoDuracaoTotal>
        <tipoDelta>254</tipoDelta>
        <velocidadeMaximaFaixaAmarela>0</velocidadeMaximaFaixaAmarela>
        <velocidadeMedia>0</velocidadeMedia>
        <dataChegada>17/10/2023 07:11:46</dataChegada>
      </return>
    </ns0:obterDeltaTelemetriaIntegracaoDataChegadaResponse>
  </S:Body>
</S:Envelope>
```


7.5. ObterDeltaTelemetriaIntegracaoInerciaDataChegada



Necessário possuir Obrigação Financeira de Software para consumo dos Deltas.

Descrição

Serviço responsável por disponibilizar aos clientes os Deltas de Telemetria, respeitando as regras de negócio previamente estabelecidas ao INTEGRADOR

Declaração do Método

```
List obterDeltaTelemetriaIntegracaoInerciaDataChegada(String usuario, String senha, Integer idVeiculo, String dataInicio, String dataFinal, String dataChegadaInicio, String dataChegadaFinal)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idVeiculo	Integer	Sim	Id do veículo
dataInicio	String	Sim	Data inicial para a consulta Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataFinal	String	Sim	Data final para a consulta Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataChegadaInicio	String	Sim	Data Início de registro no sistema Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataChegadaFinal	String	Sim	Data final de registro no sistema Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.

Retorno

Nome	Tipo	Descrição
deltmIdVeiculo	Integer	Id do veículo
deltmIdMotorista	Integer	Id do motorista
deltmloginMotorista	Integer	Login do motorista
deltmLatitude	Integer	Latitude

Nome	Tipo	Descrição
deltmLongitude	Integer	Longitude
deltmDataPosicao	String	Data do GPS (GMT-0) no momento do envio do pacote
deltmDeltaGiroMotor	Integer	Tempo em segundos(s) que houve giro de motor
deltmDeltaIntervalo	Integer	Tempo em segundos(s) decorrido do início ao fechamento do delta
deltmDeltaMovimento	Integer	Tempo em segundos(s) que o veículo se movimentou
deltmDeltaParado	Integer	Tempo em segundos(s) que o veículo ficou parado
deltmFxAmareloVelMax	Integer	Velocidade máxima em que o veículo permaneceu na faixa amarela
deltmFxVermelhoTipo	Integer	Tipo do Delta: – Ignição ON/OFF – Login/ Logoff
tempoDuracaoFaixaMachaLenta	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de marcha lenta
tempoDuracaoFaixaMachaLentaComVelocidade	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de marcha lenta estando em velocidade.
tempoDuracaoFaixaDeTransicaoComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de transição com inércia.
tempoDuracaoFaixaDeTransicaoSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de transição sem inércia.
tempoDuracaoFaixaVerdeEconomicaComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa verde econômica com inércia.
tempoDuracaoFaixaVerdeEconomicaSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa verde econômica sem inércia.
tempoDuracaoFaixaVerdeComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa verde com inércia.
tempoDuracaoFaixaVerdeSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa verde sem inércia.
tempoDuracaoFaixaAmarelaSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa amarela com inércia.
tempoDuracaoFaixaAmarelaComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa amarela sem inércia.
tempoDuracaoFaixaDePerigoComInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de perigo com inércia.
tempoDuracaoFaixaDePerigoSemInercia	Integer	Tempo em segundos(s) que o veículo permaneceu na faixa de perigo sem inércia.
deltmHorimetro	Integer	Horímetro final (no momento do fechamento do delta)
deltmHodometro	Integer	Hodômetro final (no momento do fechamento do delta)
deltmHodometroDelta	Integer	Distância percorrida (da abertura ao fechamento do delta)

Nome	Tipo	Descrição
deltmHodometroVelocidadeMedia	Integer	Velocidade média
deltmRpmMax	Integer	RPM máximo atingido entre abertura e fechamento de delta
deltmRpmMed	Integer	Média de RPM (entre abertura e fechamento de delta)
deltmTempoFreioMotor	Integer	Tempo de acionamento (em segundos) do freio motor
deltmUsoEmbreagemKm	Integer	Distância (em km) que o veículo percorreu com embreagem acionada
deltmUsoFreioKm	Integer	Distância (em km) que o veículo percorreu com freio acionado
delconsumo_combustivel	Integer	Consumo (em mililitros) de combustível desde o início até o fim do delta. Nas situações onde o Delta de Telemetria não possui a informação de Consumo de Combustível, esse campo não será apresentado no XML de Retorno.
distPercorridaAscendenteFxAmarela	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa Amarela
distPercorridaAscendenteFxMarchaLenta	Integer	Distância (em km) percorrida pelo veículo em ascendente na Marcha Lenta
distPercorridaAscendenteFxPerigo	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa de Perigo
distPercorridaAscendenteFxTransic	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa de Transição
distPercorridaAscendenteFxVerde	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa Verde
distPercorridaAscendenteFxVerde_ext	Integer	Distância (em km) percorrida pelo veículo em ascendente na Faixa Verde Econômica
distPercorridaDescendenteFxAmarela	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa Amarela
distPercorridaDescendenteFxMarchaLenta	Integer	Distância (em km) percorrida pelo veículo em descendente na Marcha Lenta
distPercorridaDescendenteFxPerigo	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa de Perigo
distPercorridaDescendenteFxTransic	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa de Transição
distPercorridaDescendenteFxVerde	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa Verde
distPercorridaDescendenteFxVerde_ext	Integer	Distância (em km) percorrida pelo veículo em descendente na Faixa Verde Econômica
distPercorridaEstavelFxAmarela	Integer	Distância (em km) percorrida pelo veículo estável na Faixa Amarela

Nome	Tipo	Descrição
distPercorridaEstavelFxMarcha Lenta	Integer	Distância (em km) percorrida pelo veículo estável na Marcha Lenta
distPercorridaEstavelFxPerigo	Integer	Distância (em km) percorrida pelo veículo estável na Faixa de Perigo
distPercorridaEstavelFxTransi c	Integer	Distância (em km) percorrida pelo veículo estável na Faixa de Transição
distPercorridaEstavelFxVerde	Integer	Distância (em km) percorrida pelo veículo estável na Faixa Verde
distPercorridaEstavelFxVerde_ext	Integer	Distância (em km) percorrida pelo veículo estável na Faixa Verde Econômica
distTotalFxAscendente	Integer	Distância Total (em km) percorrida pelo veículo em ascendente
distTotalFxDescendente	Integer	Distância Total (em km) percorrida pelo veículo em descendente
distTotalFxEstavel	Integer	Distância Total (em km) percorrida pelo veículo estável
tempoDuracaoAscendenteFxAmare la	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa Amarela
tempoDuracaoAscendenteFxMarch aLenta	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Marcha Lenta
tempoDuracaoAscendenteFxPerig o	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa de Perigo
tempoDuracaoAscendenteFxTrans ic	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa de Transição
tempoDuracaoAscendenteFxVerde	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa Verde
tempoDuracaoAscendenteFxVerde _ext	Integer	Tempo em segundos(s) que o veículo permaneceu em ascendente na Faixa Verde Econômica
tempoDuracaoDescendenteFxAmar ela	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa Amarela
tempoDuracaoDescendenteFxMarc haLenta	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Marcha Lenta
tempoDuracaoDescendenteFxPeri go	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa de Perigo
tempoDuracaoDescendenteFxTran sic	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa de Transição
tempoDuracaoDescendenteFxVerd e	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa Verde
tempoDuracaoDescendenteFxVerd e_ext	Integer	Tempo em segundos(s) que o veículo permaneceu em descendente na Faixa Verde Econômica

Nome	Tipo	Descrição
tempoDuracaoEstavelFxAmarela	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa Amarela
tempoDuracaoEstavelFxMarchaLenta	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Marcha Lenta
tempoDuracaoEstavelFxPerigo	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa de Perigo
tempoDuracaoEstavelFxTransic	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa de Transição
tempoDuracaoEstavelFxVerde	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa Verde
tempoDuracaoEstavelFxVerde_ext	Integer	Tempo em segundos(s) que o veículo permaneceu estável na Faixa Verde Econômica
tempoTotalFxAscendente	Integer	Tempo Total em segundos(s) que o veículo permaneceu em ascendente
tempoTotalFxDescendente	Integer	Tempo Total em segundos(s) que o veículo permaneceu em descendente
tempoTotalFxEstavel	Integer	Tempo Total em segundos(s) que o veículo permaneceu estável
pressaoMediaDoOleoDoMotor	Integer	Pressão média do óleo
pressaoMediaDoSistemaDeFreioAAr	Integer	Pressão Média do Sistema de Freios a Ar
temperaturaMediaDoArrefecimento	Integer	Temperatura Média de Arrefecimento
temperaturaMediaDoCombustivel	Integer	Temperatura Média do Combustível
temperaturaMediaDoOleoDoMotor	Integer	Temperatura Média do Óleo
tempoTotalComCinto	Integer	Tempo Total com Cinto
tensaoMediaDaBateria	Integer	Tensão Média da Bateria
dataChegada	String	Data da chegada do pacote

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterDeltaTelemetriaIntegracaoInerciaDataChegada>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <dataInicio>2023-10-17 00:00:00</dataInicio>
    </web:obterDeltaTelemetriaIntegracaoInerciaDataChegada>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<dataFinal>2023-10-17 23:59:00</dataFinal>
<dataChegadaInicio>2023-10-17 00:00:00</dataChegadaInicio>
<dataChegadaFinal>2023-10-17 00:00:00</dataChegadaFinal>
<idVeiculo>881456</idVeiculo>
</web:obterDeltaTelemetriaIntegracaoInerciaDataChegada>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <!-- Resposta truncada ou simplificada -->
  </S:Body>
</S:Envelope>
```

7.6. ObterEventoTelemetriaIntegracaoDataChegada



Necessário possuir Obrigação Financeira de Software para consumo dos Eventos.

Descrição

Serviço responsável por disponibilizar aos clientes os Eventos de Telemetria, respeitando as regras de negócio previamente estabelecidas ao INTEGRADOR.

Declaração do Método

```
List obterEventoTelemetriaIntegracaoDataChegada(String usuario, String senha, Integer idVeiculo, Integer idEventoList, String dataInicio, String dataFinal, String dataChegadaInicio, String dataChegadaFinal)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
idVeiculo	Integer	Sim	Id do veículo
idEventoList	Integer	Não	Id do(s) evento(s) Ex: 1 2
dataInicio	String	Não	Data inicial para a consulta Padrão: AAAA-MM-DDTHH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataFinal	String	Não	Data final para a consulta Padrão: AAAA-MM-DDTHH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataChegadaInicio	String	Sim	Data Início de registro no sistema Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.
dataChegadaFinal	String	Sim	Data final de registro no sistema Padrão: AAAA-MM-DD HH:MM:SS Regra: Período máximo de extração de dados deve ser 1 (um) dia.

Retorno

Nome	Tipo	Descrição
evetmIdVeiculo	Integer	Id do veículo
evetmIdMotorista	Integer	Id do motorista

Nome	Tipo	Descrição
evetmloginMotorista	Integer	Login do motorista
evetmlatitude	Integer	Latitude
evetmlongitude	Integer	Longitude
evetmDataPacote	String	Data do pacote do Evento
evetmIdTelemetria	Integer	Id do evento telemetria: • 210 — Sistema de Freio a Ar • 211 — Cinto • 214 — Força G Lateral Fraca • 215 — Força G Lateral Média - Força G Lateral Forte - Tempo parado - Pressão de óleo - Tensão do alternador - Temperatura do líquido de arrefecimento - Excesso de rotação - Excesso velocidade trecho rodoviário com chuva - Excesso velocidade trecho rodoviário seco - Excesso velocidade trecho urbano com chuva - Excesso velocidade trecho urbano seco - Tempo de marcha lenta excessivo - Tempo longo de ignição sem giro de motor - Pé na embreagem - Arrancada brusca - Freada brusca - Banguela - Arrancada em segunda marcha - Eventos de percurso - Entradas e saídas - Horímetro veículo movimento/ hodômetro - Horímetro motor ligado/ veículo parado - Horímetro fx verde/vermelha - Horímetro fx azul/amarela
evetmhodometro	Integer	Hodômetro no momento do Evento.
evetmhorimetro	Integer	Horímetro no momento do Evento.
evetmTempoDuracao	Integer	Tempo de Duração do Evento.
velocidadeMaximaEvento	Integer	Velocidade máxima atingida durante o evento.
velocidadeReferencia	Integer	Velocidade de limite cadastrado no evento.
dataChegada	String	Data da chegada do Evento

Exemplo XML

Request:

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/"
>
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterEventoTelemetriaIntegracaoDataChegada>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <dataInicio>2023-10-17 00:00:00</dataInicio>
```



```
<dataFinal>2023-10-17 23:59:00</dataFinal>
<dataChegadaInicio>2023-10-17 00:00:00</dataChegadaInicio>
<dataChegadaFinal>2023-10-17 00:00:00</dataChegadaFinal>
<idVeiculo>881456</idVeiculo>
<!--Zero or more repetitions:-->
<idEventoList>?</idEventoList>
</web:obterEventoTelemetriaIntegracaoDataChegada>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterEventoTelemetriaIntegracaoDataChegadaResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <dataPosicao>17/10/2023 05:43:03</dataPosicao>
        <horimetro>217006</horimetro>
        <idEvento>210</idEvento>
        <idMotorista>0</idMotorista>
        <idVeiculo>1207216</idVeiculo>
        <latitude>-26.2539</latitude>
        <loginMotorista>0</loginMotorista>
        <longitude>-48.61469</longitude>
        <nomeMotorista>Sem motorista</nomeMotorista>
        <odometro>484878</odometro>
        <tempoDuracao>0</tempoDuracao>
        <valorConfigurado>600</valorConfigurado>
        <valorVioladoFreioAr>0</valorVioladoFreioAr>
        <velocidadeMaximaEvento>0</velocidadeMaximaEvento>
        <velocidadeReferencia>0</velocidadeReferencia>
        <dataChegada>17/10/2023 05:43:04</dataChegada>
      </return>
    </ns0:obterEventoTelemetriaIntegracaoDataChegadaResponse>
  </S:Body>
</S:Envelope>
```

7.7. getSmartCamerasEvents

Descrição

Método para consultar os eventos de câmeras inteligentes dos veículos, para visualização da gerenciadora.



Regra de restrição

Para obtenção de datas, o método usa o timestamp com UTC 0 (zero).

Declaração do Método

```
List getSmartCamerasEvents(String agrupador)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
agrupador	String	Sim	Usuário que visualizará as informações das empresas que concederam acesso aos veículos
senha	String	Sim	Senha do integrador para acesso ao sistema
offset	Integer	Sim	Define a paginação dos resultados
limit	Integer	Sim	Usado para recurso de paginação dos resultados
motoristas	String	Não	Lista os ids de motoristas no formato "1,2,3,4,5" (sem aspas)
veículos	String	Não	Lista de placas a serem filtradas no formato "ABC123,CDE456,FGH789" (sem aspas)
dataInicio	String	Sim	Data de início do filtro no formato DD-MM-YYYY
dataFim	String	Sim	Data de fim do filtro no formato DD-MM-YYYY
tipoEvento	Integer	Não	Evento apresentado pelo veículo
criticidade	Integer	Não	Lista de códigos de criticidade a serem filtrados no formato "1,2,3" (sem aspas), os códigos devem ser os mesmos códigos utilizados no sistema SMART CAMERAS
turno	String	Não	Lista de horários a serem filtrados no formato "hh:mm-hh:mm;hh:mm-hh:mm;hh:mm-hh:mm; esses horários são referentes às opções pré existentes na aplicação Câmeras, a saber: "Madrugada" (compreenderá horários entre 00h e 06h), "Manhã" (compreenderá horários entre 06h:01min e 12h), "Tarde" (compreenderá horários entre 12h:01min e 18h) e "Noite" (compreenderá horário entre 18h:01min e 23h:59min). Poderá ser selecionado mais de um turno;

Nome	Tipo	Obrigatório	Descrição
diaSemana	Integer	Não	lista com dias da semana no formato "1,2,3,4,5,6,7" (sem aspas), é utilizado o padrão ISO-8601, onde Segunda-feira=1, Terça-feira=2, Quarta-feira=3, Quinta-feira=4, Sexta-feira=5, Sábado=6, Domingo=7
quantidade	Integer	Não	Quantidade que será considerada para mostrar os eventos; por exemplo, se for informado o valor 2, apenas os veículos que tiveram dois eventos iguais serão retornados, se for informado o valor 4, apenas os veículos que tiveram quatro eventos iguais serão retornados.
status	String	Não	Lista com status a serem filtrados no formato " 'validated','not_validated' " (sem aspas dupla, apenas aspas simples). Os status devem ser os seguintes: - Validado: 'validated' - Não validado: 'not_validated' - Aguardando vídeo: 'waiting_video' - Falha: 'failed'

Retorno

Retorna uma lista de objetos do tipo '[SmartCamerasEvento](#)' com as propriedades abaixo:

Nome	Tipo	Descrição
deviceId	Integer	Reservado do sistema
driver	String	Lista os motoristas
cpf	String	Número do documento do motorista
nome	String	Nome do motorista
registration	String	Matrícula do motorista
eventType	String	Tipo de evento realizado pelo veículo
hwType	String	Reservado do sistema
id	Integer	Reservado do sistema
messageId	String	Reservado do sistema
event_id	Integer	Código do evento
gps_valid	Integer	Reservado do sistema
lat	Double	Valor da latitude do veículo que gerou o evento
lon	Double	Valor da longitude do veículo que gerou o evento
timestamp	String	Data e hora do evento em UTC 0
vel	Integer	Velocidade do veículo
pkDeviceDate	Integer	Reservado do sistema

Nome	Tipo	Descrição
plate	String	Placa do veículo que gerou o evento



OBS.: Informações serão obtidas somente se houver pelo menos uma nota cadastrada. Se não forem passados os parâmetros opcionais, serão retornados todos os eventos disponíveis.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getSmartCamerasEvents>
      <!--Optional:-->
      <usuario>LOGIN</usuario>
      <!--Optional:-->
      <agrupador>LOGINAGRUPADOR</agrupador>
      <!--Optional:-->
      <senha>SENHA</senha>
      <offset>0</offset>
      <limit>100</limit>
      <!--Optional:-->
      <motoristas/>
      <!--Optional:-->
      <veiculos/>
      <!--Optional:-->
      <dataInicio>17-06-2021</dataInicio>
      <!--Optional:-->
      <dataFim>17-06-2021</dataFim>
      <!--Optional:-->
      <tipoEvento/>
      <!--Optional:-->
      <criticidade/>
      <!--Optional:-->
      <turno/>
      <!--Optional:-->
      <diaSemana/>
      <quantidade>2</quantidade>
      <!--Optional:-->
      <status/>
    </web:getSmartCamerasEvents>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <work:WorkContext xmlns:work="http://oracle.com/weblogic/soap/workarea/">
      XYZ</work:WorkContext>
    </S:Header>
  <S:Body>
    <ns0:getSmartCamerasEventsResponse xmlns:ns0=
"http://webservice.web.integracao.sascar.com.br/">
      <return>
        <deviceId>1265698</deviceId>
        <driver>
          <cpf/>
          <nome>No driver</nome>
          <registration/>
        </driver>
        <eventType>1</eventType>
        <hwType>streamax</hwType>
        <id>STREAMAX_ALARMID_2312615</id>
        <messageId>2312615</messageId>
        <payload>
          <event_id>56006</event_id>
          <gps_valid>1</gps_valid>
          <lat>-22.04364</lat>
          <lon>-47.430843</lon>
          <timestamp>2021-06-17 00:00:18</timestamp>
          <vel>45</vel>
        </payload>
        <pkDeviceDate>STREAMAX_2021-06-17_1265698</pkDeviceDate>
        <plate>GFA3536</plate>
        <timestamp>2021-06-17 00:00:18</timestamp>
      </return>
      <return>
        <deviceId>1265698</deviceId>
        <driver>
          <cpf/>
          <nome>No driver</nome>
          <registration/>
        </driver>
        <eventType>1</eventType>
        <hwType>streamax</hwType>
        <id>STREAMAX_ALARMID_2312617</id>
        <messageId>2312617</messageId>
        <payload>
          <event_id>56006</event_id>
          <gps_valid>1</gps_valid>
          <lat>-22.043132</lat>
          <lon>-47.430967</lon>
          <timestamp>2021-06-17 00:00:23</timestamp>
          <vel>41</vel>
        </payload>
        <pkDeviceDate>STREAMAX_2021-06-17_1265698</pkDeviceDate>
        <plate>GFA3536</plate>
        <timestamp>2021-06-17 00:00:23</timestamp>
      </return>
    </ns0:getSmartCamerasEventsResponse>
  </S:Body>
</S:Envelope>
```

7.8. SolicitarEventosCaixaPreta



Este método está desativado, sem previsão de liberação.

Descrição

Método que permite solicitar a recuperação de eventos caixa preta armazenados no sdcard do equipamento para gravação dos mesmos em banco de dados.

Declaração do Método

```
String solicitarEventosCaixaPreta(String usuario, String senha, String veioId, String placa, String dataPosicaoInicial, String dataPosicaoFinal)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
veioId	String	Sim ¹	Id do veículo para o qual deseja-se recuperar os eventos caixa preta.
placa	String	Sim ¹	Placa do veículo para o qual deseja-se recuperar os eventos caixa preta.
dataPosicaoInicial	String	Não ²	Indica a data inicial para o intervalo do qual deseja-se obter os eventos caixa preta.
dataPosicaoFinal	String	Não ²	Indica a data final para o intervalo do qual deseja-se obter os eventos caixa preta.



Caso o campo “veioId” seja informado, o campo “placa” torna-se opcional. Caso o campo “placa” seja informado, o campo “veioId” torna-se opcional.



Não é possível informar apenas o campo “dataPosicaoInicial” ou o campo “dataPosicaoFinal”, porém, pode-se omitir ambos e o serviço enviará comando solicitando os últimos 10 minutos armazenados.

Retorno

Retorna o indicativo de execução do comando que solicita a recuperação dos dados de caixa preta armazenados no sd-card do rastreador:

Nome	Tipo	Descrição
Código de retorno	Integer	Indica se o comando foi executado: 7: comando executado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:solicitarEventosCaixaPreta>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <idVeiculo>1575379</idVeiculo>
      <!--Optional:-->
      <placa>?</placa>
      <!--Optional:-->
      <dataPosicaoInicial>?</dataPosicaoInicial>
      <!--Optional:-->
      <dataPosicaoFinal>?</dataPosicaoFinal>
      <!--Optional:-->
      <ticket>?</ticket>
    </web:solicitarEventosCaixaPreta>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:Fault xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:ns1="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>ns0:Server</faultcode>
      <faultstring>Atencao: Método desativado. Sem previsão de liberação.</faultstring>
    </ns0:Fault>
  </S:Body>
</S:Envelope>
```

7.9. RecuperarEventosCaixaPreta

Descrição

Método que permite consultar os eventos caixa preta já obtidos do rastreador e armazenados em base de dados.

Declaração do Método

```
CaixaPretaList recuperarEventosCaixaPreta(String usuario, String senha, String placa, String veioId, String dataPosicao)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
placa	String	Sim ¹	Placa do veículo para o qual deseja-se recuperar os eventos caixa preta.
veioId	String	Sim ¹	Id do veículo para o qual deseja-se recuperar os eventos caixa preta.
dataPosicao	String	Sim	Indica a data inicial para o intervalo do qual deseja-se obter os eventos caixa preta.



1: Caso o campo “veioId” seja informado, o campo “placa” torna-se opcional. Caso o campo “placa” seja informado, o campo “veioId” torna-se opcional.

Retorno

Retorna uma instância de CaixaPretaList, que encapsula a lista de eventos caixa preta contidos no intervalo de 10 minutos contados a partir do valor do campo “dataPosicao”:

Nome	Tipo	Descrição
caixaPretaList	List<Caixa Preta>	Elemento que encapsula a lista contendo os eventos de caixa preta obtidos
dataEvento	String	Representa a data em que o evento ocorreu, inclusive minutos e segundos
latitude	Double	Representa a latitude onde o evento ocorreu
longitude	Double	Representa a longitude onde o evento ocorreu
idOperador	Integer	Representa o id do motorista registrado como condutor no momento em que o evento ocorreu

Nome	Tipo	Descrição
velocidade	Integer	Representa a velocidade em que o veículo se encontrava no momento em que o evento ocorreu
rpm	Integer	Representa o valor das rotações no momento em que o evento ocorreu
ignicao	Integer	Representa o estado da ignição no momento em que o evento ocorreu: 0: Desligada 1: Ligada
freio	Integer	Representa o estado do freio no momento em que o evento ocorreu: 0: Desativado 1: Ativado
limpador	Integer	Representa o estado do limpador no momento em que o evento ocorreu: 0: Desativado 1: Ativado
buzzer	Integer	Representa o estado do buzzer no momento em que o evento ocorreu: 0: Desativado 1: Ativado
embreagem	Integer	Representa o estado da embreagem no momento em que o evento ocorreu: 0: Desativado 1: Ativado

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:recuperarEventosCaixaPreta>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <placa?></placa>
      <!--Optional:-->
      <idVeiculo>1903888</idVeiculo>
      <!--Optional:-->
      <dataPosicao?></dataPosicao>
    </web:recuperarEventosCaixaPreta>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:recuperarEventosCaixaPretaResponse
xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        <caixaPreta>
          <dataEvento>26/02/2016 23:08:00</dataEvento>
          <latitude>-25.323203</latitude>
          <longitude>-49.119539</longitude>
          <idOperador>1</idOperador>
          <velocidade>0</velocidade>
          <rpm>807</rpm>
          <ignicao>-1</ignicao>
          <freio>1</freio>
          <limpador>1</limpador>
          <buzzer>1</buzzer>
          <embreagem>1</embreagem>
        </caixaPreta>
        <caixaPreta>
          <dataEvento>26/02/2016 23:08:01</dataEvento>
          <latitude>-25.323203</latitude>
          <longitude>-49.119539</longitude>
          <idOperador>1</idOperador>
          <velocidade>0</velocidade>
          <rpm>807</rpm>
          <ignicao>-1</ignicao>
          <freio>1</freio>
          <limpador>1</limpador>
          <buzzer>1</buzzer>
          <embreagem>1</embreagem>
        </caixaPreta>
        <caixaPreta>
          <dataEvento>26/02/2016 23:08:02</dataEvento>
          <latitude>-25.323203</latitude>
          <longitude>-49.119539</longitude>
          <idOperador>1</idOperador>
          <velocidade>0</velocidade>
          <rpm>807</rpm>
          <ignicao>-1</ignicao>
          <freio>1</freio>
          <limpador>1</limpador>
          <buzzer>1</buzzer>
          <embreagem>1</embreagem>
        </caixaPreta>
        <caixaPreta>
          <dataEvento>26/02/2016 23:08:03</dataEvento>
          <latitude>-25.323203</latitude>
          <longitude>-49.119539</longitude>
          <idOperador>1</idOperador>
          <velocidade>0</velocidade>
          <rpm>807</rpm>
          <ignicao>-1</ignicao>
          <freio>1</freio>
          <limpador>1</limpador>
          <buzzer>1</buzzer>
          <embreagem>1</embreagem>
        </caixaPreta>
      </return>
    </ns0:recuperarEventosCaixaPretaResponse>
  </S:Body>
</S:Envelope>
```

26/02/2016 23:08:04 -25.323203 -49.119539 1 0 807 -1 1 1 1 1

8. Métodos – Motoristas

8.1. ObterMotoristas

Descrição

Método para consulta dos dados cadastrais de motoristas pertencentes ao INTEGRADOR.



Caso cliente esteja com restrição, não será possível consultar motoristas do mesmo.

Declaração do Método

```
List obterMotoristas(String usuario, String senha, Integer quantidade, Integer idMotorista)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Sim	Quantidade máxima de registros da consulta, sendo o limite de 1000 registros. Para obter as informações pertinentes apenas à um registro de motorista, preencha o campo quantidade com o valor 1 (um) e indique o id do motorista desejado no campo idMotorista.
idMotorista	Integer	Não	O Limite máximo de resultados do método é 1000 registros. Neste caso para obter o restante dos registros, foi implementado um sistema de paginação, onde o INTEGRADOR deve seguir as seguintes regras: Para obter a primeira página com os 1000 registros iniciais enviar 0 (zero) como parâmetro; Para obter a próxima página, enviar como parâmetro o último id do motorista da última página consultada. Para obter as informações pertinentes apenas à um registro de motorista, enviar apenas o id do motorista desejado e o no campo quantidade o valor 1 (um).

Retorno

Retorna uma lista de objetos do tipo 'Motorista' com as propriedades abaixo:

Nome	Tipo	Descrição
idMotorista	Integer	Id do motorista.
nome	String	Nome do motorista.

Nome	Tipo	Descrição
tipoMotorista	Char	Indica o tipo do motorista. A (Agregado) T (Terceiro) F (Funcionario)
dataContratacao	Calendar	Data da contratação do motorista.
tipoDocumento	Char	Tipo do documento utilizado para identificação. C (CNH) R (RG) F (CPF)
numeroDocumento	String	Número do documento utilizado para identificação.
tipoCNH	String	Tipo / Categoria da CNH do motorista.
vencimentoCNH	Calendar	Data vencimento da CNH do motorista.
telefone	String	Telefone principal do motorista.
celular	String	Celular principal do motorista.
login	String	Login do motorista.
senha	String	Senha do motorista.
generico	Boolean	Indica se o motorista é genérico à gerenciadora. true (Motorista é generico sem vinculo a cliente) false (Motorista tem vinculo a cliente)

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/" >
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterMotoristas>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>Quantidade</quantidade>
      <!--Optional:-->
      <idMotorista>IDMOTORISTA</idMotorista>
    </web:obterMotoristas>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterMotoristasResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/" >
      <return>
        <celular>(77) 7777-7777</celular>
        <dataContratacao>2004-12-12T00:00:00.000-02:00</dataContratacao>
        <generico>>false</generico>
      </return>
    </ns0:obterMotoristasResponse>
  </S:Body>
</S:Envelope>
```

```
<idMotorista>6530</idMotorista>
<login>2424</login>
<nome>Ruy Leal</nome>
<numeroDocumento>4322222432222222222</numeroDocumento>
<senha>2424</senha>
<telefone>(77) 7777-7777</telefone>
<tipoCNH>AB</tipoCNH>
<tipoDocumento>R</tipoDocumento>
<vencimentoCNH>2009-09-28T00:00:00.000-03:00</vencimentoCNH>
</return>
</ns0:obterMotoristasResponse>
</S:Body>
</S:Envelope>
```

8.2. ObterMotoristasJson

Descrição

Método para consulta dos dados cadastrais de motoristas pertencentes ao INTEGRADOR, com retorno no formato JSON. Equivalente ao método `obterMotoristas`, com a adição dos campos `ativo` e `dataInativacao`.



Motoristas excluídos ou pertencentes a clientes sem sinal direcionado não são retornados. Clientes com restrição não permitem a consulta de seus motoristas.

Declaração do Método

```
List obterMotoristasJson(String usuario, String senha, Integer quantidade, Integer idMotorista)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
<code>usuario</code>	String	Sim	Login do integrador para acesso ao sistema.
<code>senha</code>	String	Sim	Senha do integrador para acesso ao sistema.
<code>quantidade</code>	Integer	Sim	Quantidade máxima de registros da consulta, sendo o limite de 1000 registros. Para obter as informações pertinentes apenas a um registro de motorista, preencha o campo <code>quantidade</code> com o valor <code>1</code> e indique o id no campo <code>idMotorista</code> .
<code>idMotorista</code>	Integer	Não	Paginação: enviar <code>0</code> para obter os primeiros 1000 registros; para as páginas seguintes, enviar o último <code>idMotorista</code> da página anterior. Para consultar um motorista específico, enviar seu id com <code>quantidade = 1</code> .

Retorno

Retorna uma lista de objetos do tipo `Motorista` com as propriedades abaixo:

Nome	Tipo	Descrição
<code>idMotorista</code>	Integer	Id do motorista.
<code>nome</code>	String	Nome do motorista.
<code>tipoMotorista</code>	Char	Tipo do motorista: <code>A</code> (Agregado), <code>T</code> (Terceiro), <code>F</code> (Funcionário).
<code>dataContratacao</code>	Calendar	Data da contratação do motorista.
<code>tipoDocumento</code>	Char	Tipo do documento: <code>C</code> (CNH), <code>R</code> (RG), <code>F</code> (CPF).
<code>numeroDocumento</code>	String	Número do documento utilizado para identificação.

Nome	Tipo	Descrição
tipoCNH	String	Tipo / Categoria da CNH do motorista.
vencimentoCNH	Calendar	Data de vencimento da CNH do motorista.
telefone	String	Telefone principal do motorista.
celular	String	Celular principal do motorista.
login	String	Login do motorista.
senha	String	Senha do motorista.
generico	Boolean	true se o motorista é genérico sem vínculo a cliente; false se possui vínculo.
ativo	Boolean	Situação cadastral do motorista: • true — motorista ativo • false — motorista inativo
dataInativacao	Calendar	Data de inativação do motorista. Preenchida quando ativo = false ; null quando o motorista está ativo.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:obterMotoristasJson>
      <usuario>Usuário</usuario>
      <senha>Senha</senha>
      <!--Optional:-->
      <quantidade>1000</quantidade>
      <!--Optional:-->
      <idMotorista>0</idMotorista>
    </web:obterMotoristasJson>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterMotoristasJsonResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/">
      <return>
        [
          {
            "idMotorista": 6530,
            "nome": "Ruy Leal",
            "tipoMotorista": "F",
```



```
"dataContratacao": "2004-12-12T00:00:00.000-02:00",
"tipoDocumento": "R",
"numeroDocumento": "4322222432222222222",
"tipoCNH": "AB",
"vencimentoCNH": "2009-09-28T00:00:00.000-03:00",
"telefone": "(77) 7777-7777",
"celular": "(77) 7777-7777",
"login": "2424",
"senha": "2424",
"generico": false,
"ativo": true,
"dataInativacao": null
}
]
</return>
</ns0:obterMotoristasJsonResponse>
</S:Body>
</S:Envelope>
```

8.3. ObterMotoristasVeiculos

Descrição

Método para consulta dos dados cadastrais de vínculos entre motoristas e veículos pertencentes ao INTEGRADOR.

Declaração do Método

```
List obterMotoristasVeiculos(String usuario, String senha, Integer quantidade, Long idMotoristaVeiculo)
```

Parâmetros

Nome	Tipo	Obrigatório	Descrição
usuario	String	Sim	Login do integrador para acesso ao sistema.
senha	String	Sim	Senha do integrador para acesso ao sistema.
quantidade	Integer	Sim	Quantidade máxima de registros da consulta, sendo o limite de 1000 registros.
idMotoristaVeiculo	Long	Não	O Limite máximo de resultados do método é 1000 registros. Neste caso para obter o restante dos registros, foi implementado um sistema de paginação, onde o INTEGRADOR deve seguir as seguintes regras: Para obter a primeira página com os 1000 registros iniciais enviar 0 (zero) como parâmetro; Para obter a próxima página, enviar como parâmetro o último id do motorista da última página consultada.

Retorno

Retorna uma lista de objetos do tipo 'MotoristaVeiculo' com as propriedades abaixo:

Nome	Tipo	Descrição
idMotoristaVeiculo	Long	Id do registro de motorista e veículo
idMotorista	Integer	Id do motorista.
idVeiculo	Integer	Id do veículo.

Exemplo XML

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://webservice.web.integracao.sascar.com.br/" >
  <soapenv:Header/>
  <soapenv:Body>
```

```
<web:obterMotoristasVeiculos>
  <usuario>Usuário</usuario>
  <senha>Senha</senha>
  <!--Optional:-->
  <quantidade>QuantidadeDeRetorno</quantidade>
  <!--Optional:-->
  <idMotoristaVeiculo>idMotoristaVeiculo</idMotoristaVeiculo>
</web:obterMotoristasVeiculos>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <WorkContext xmlns="http://oracle.com/weblogic/soap/workarea/">XYZ</WorkContext>
  </S:Header>
  <S:Body>
    <ns0:obterMotoristasVeiculosResponse xmlns:ns0="http://webservice.web.integracao.sascar.com.br/" >
      <return>
        <idMotorista>202</idMotorista>
        <idMotoristaVeiculo>202113516</idMotoristaVeiculo>
        <idVeiculo>113516</idVeiculo>
      </return>
    </ns0:obterMotoristasVeiculosResponse>
  </S:Body>
</S:Envelope>
```

9. Referência – Estrutura de Eventos e Exceções

9.1. Estrutura Atual

O sistema de integração atual possui uma estrutura, que dificulta a identificação de qual atuador foi o responsável por gerar o evento disparado pelo equipamento. Essa limitação ocorre, pois a identificação do atuador não é informada, sendo somente o status de cada entrada/saída como demonstrado na tabela abaixo:

Nome	Código Status ON	Código Status OFF	Descrição
Entrada 1	17	21	Evento gerado pelo dispositivo da entrada 1, sendo 17 para entrada ativada e 21 para desativada.
Entrada 2	18	22	Evento gerado pelo dispositivo da entrada 2, sendo 18 para entrada ativada e 22 para desativada.
Entrada 3	19	23	Evento gerado pelo dispositivo da entrada 3, sendo 19 para entrada ativada e 23 para desativada.
Entrada 4	20	24	Evento gerado pelo dispositivo da entrada 4, sendo 20 para entrada ativada e 24 para desativada.
Entrada 5	40	44	Evento gerado pelo dispositivo da entrada 5, sendo 40 para entrada ativada e 44 para desativada.
Entrada 6	41	45	Evento gerado pelo dispositivo da entrada 6, sendo 41 para entrada ativada e 45 para desativada.
Entrada 7	42	46	Evento gerado pelo dispositivo da entrada 7, sendo 42 para entrada ativada e 46 para desativada.
Entrada 8	43	47	Evento gerado pelo dispositivo da entrada 8, sendo 43 para entrada ativada e 47 para desativada.
Saída1	25	29	Evento gerado pelo dispositivo da entrada 1, sendo 25 para entrada ativada e 29 para desativada.
Saída2	26	30	Evento gerado pelo dispositivo da entrada 2, sendo 26 para entrada ativada e 30 para desativada.
Saída3	27	31	Evento gerado pelo dispositivo da entrada 3, sendo 27 para entrada ativada e 31 para desativada.
Saída4	28	32	Evento gerado pelo dispositivo da entrada 4, sendo 28 para entrada ativada e 32 para desativada.
Saída5	48	52	Evento gerado pelo dispositivo da entrada 5, sendo 48 para entrada ativada e 52 para desativada.
Saída6	49	53	Evento gerado pelo dispositivo da entrada 6, sendo 49 para entrada ativada e 53 para desativada.
Saída7	50	54	Evento gerado pelo dispositivo da entrada 7, sendo 50 para entrada ativada e 54 para desativada.

Nome	Código Status ON	Código Status OFF	Descrição
Saída8	51	55	Evento gerado pelo dispositivo da entrada 8, sendo 51 para entrada ativada e 55 para desativada.
Bloqueio	33	34	Evento de bloqueio sendo 33 para entrada ativo e 34 bloqueio desativado.
GPS	35	36	Evento de bloqueio sendo 35 para entrada ativo e 36 bloqueio desativado.
Ignição	37	38	Evento de bloqueio sendo 37 para entrada ativo e 38 bloqueio desativado.
Pânico	1	-	Evento de código 1 indicando a ocorrência de pânico. Este tipo de evento não possui o status OFF.
Alerta Interno	3	-	Evento de código 3 indicando a ocorrência de alerta interno. Este tipo de evento não possui o status OFF.
Alerta Bateria	4	-	Evento de código 4 indicando a ocorrência de alerta de bateria. Este tipo de evento não possui o status OFF.
Velocidade	5	-	Evento de código 5 indicando a ocorrência de alerta de excesso de velocidade. Este tipo de evento não possui o status OFF.
Chegada Ponto de Referência	8	-	Evento de código 8 indicando a ocorrência de chegada ponto de referência. Este tipo de evento não possui o status OFF.
Saída Ponto de Referência	9	-	Evento de código 9 indicando a ocorrência de saída ponto de referência. Este tipo de evento não possui o status OFF.
Excesso Tempo	15	-	Evento de código 15 indicando a ocorrência de excesso de tempo. Este tipo de evento não possui o status OFF.
Ponto Referência	16	-	Evento de código 16 indicando a ocorrência de ponto de referência. Este tipo de evento não possui o status OFF.

9.2. Nova Estrutura de Eventos

A nova estrutura proporcionará uma forma mais eficiente de identificação para os eventos gerados pelos equipamentos, pois será fornecido o id do atuador como demonstrado no exemplo abaixo:

Id do Atuador	Descrição
231	O código 231 é a identificação do sensor de violação de painel. Neste caso o número positivo indica que o painel foi violado. Caso a violação seja corrigida, um novo evento será gerado com o número negativo.
-249	O código 249 é a identificação do sensor do baú. Neste caso o número negativo indica que o baú foi fechado. Caso o baú seja aberto, um novo evento será gerado com o número positivo.

O id do evento gerado será a identificação exata do atuador instalado no equipamento e poderá ser relacionado com o resultado do método `ObterGrupoAtuadores()`, que possui a relação e o detalhamento de todos os atuadores, sensores e acessórios passíveis de instalação pela Sascar.

9.3. Exemplos da Nova Estrutura de Eventos

Para exemplificar o funcionamento da nova estrutura de eventos é necessário observar o campo evento **em uma das posições obtidas pelo método obterPacotePosicoes**. Este campo é a listagem dos objetos Eventos ocorridos no espaço de tempo entre a posição analisada e a anterior. A lista contém o campo código do atuador alterado e seu estado indicado pelo sinal do código, sendo positivo quando foi ativado e negativo quando foi desativado.

Lista de eventos

Código	Descrição
231	Sensor de violação. (Ativado)
-249	Sensor de baú. (Desativado)
251	Sensor de Porta de Motorista.(Ativado)
-241	Desengate.(Desativado)

Neste exemplo, o veículo teve seu sensor ativado, o sensor de baú desativado, o sensor de porta do motorista ativado e desengate desativado. Caso, seja alterado qualquer estado do sensor será enviado na próxima posição.

9.4. Mensagens de Exceção

Mensagem	Motivo
Atenção: Veículo sem permissão!	Contrato sem obrigações financeiras
Atenção: Veículo sem dispositivo instalado!	Contrato sem telemetria instalada
Atenção: Veículo indisponível para a Integradora!	Veículo não direcionado

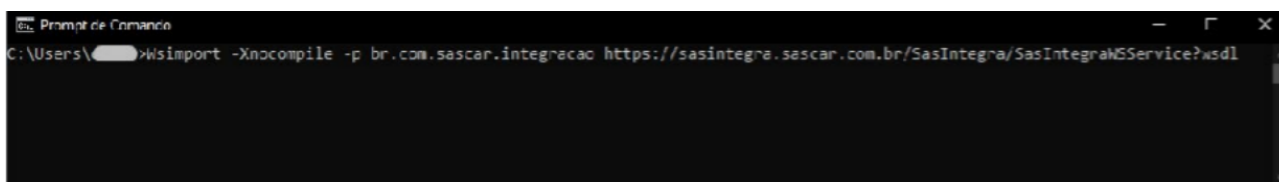
10. Anexo

10.1. Criação de Cliente para Conexão com o Web Service SasIntegra

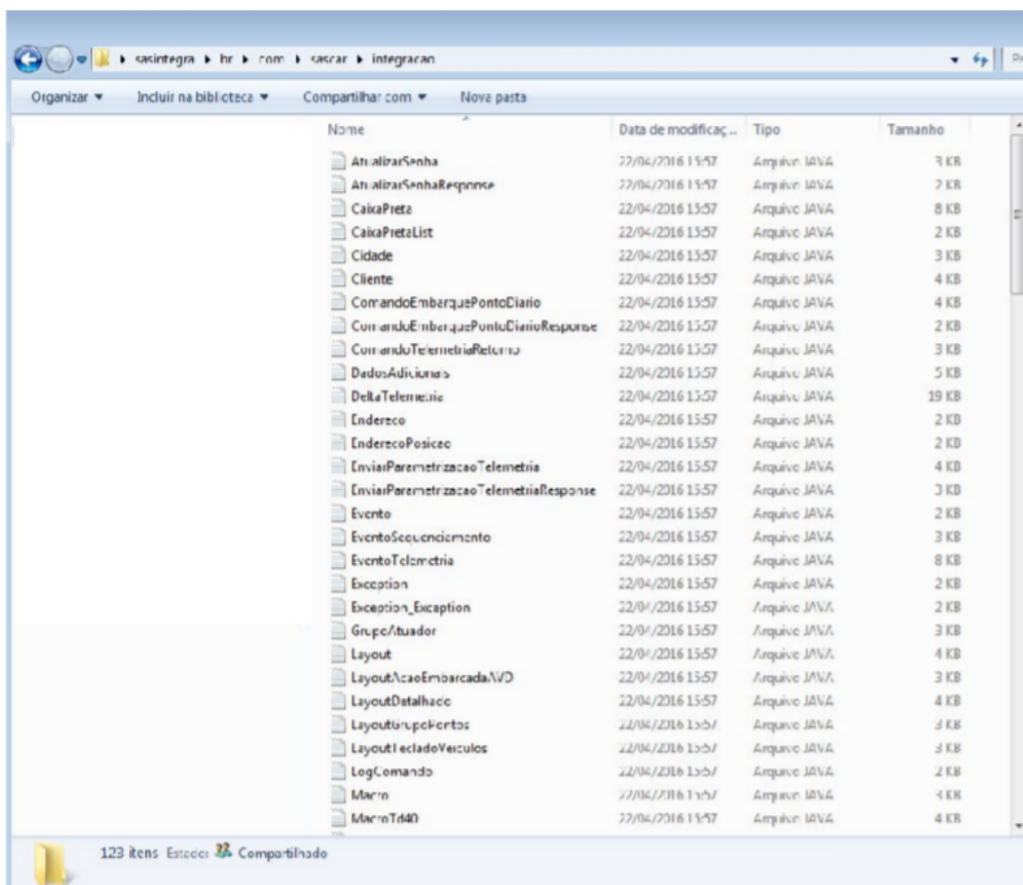
Para criar um cliente para conexão e uso dos métodos disponibilizados no SasIntegra, apresentamos um exemplo abaixo de um cliente desenvolvido em linguagem Java, para leitura do método de posições.

Primeiro é necessário importar os métodos do web service para a sua aplicação, através do import por WSDL, conforme comando de importação utilizado abaixo.

```
wsimport -Xnocompile -p br.com.sascar.integracao  
https://sasintegra.sascar.com.br/SasIntegra/SasIntegraWSService?wsdl
```



Pasta contendo o arquivo gerado pelo wsimport:



Após os métodos estarem importados em classes da linguagem Java, é necessário utilizar os trechos de código abaixo para disparo de uma requisição para pedido de pacotes de posição.

```
1 package br.com.sascar.integracao.main;
2
3 import java.util.List;
4
5 public class Main {
6
7     public static SasIntegraWS port;
8     public static List<PacotePosicao> posicaoList = null;
9
10    public static void main(String[] args) {
11        // para versões inferiores de Java JDK 6u11 o TLSv1.2 não é suportado
12        // para versões iguais ou acima do JDK 6u11 o TLSv1.2 é suportado mas precisa ser explicitado, assim como para o Java 7
13        //System.setProperty("https.protocols", "TLSv1.2");
14
15        SasIntegraWSService service = new SasIntegraWSService();
16        port = service.getSasIntegraWSPort();
17
18        try {
19            posicaoList = Main.port.obterPacotePosicoes("usuario", "senha", 300);
20            for (PacotePosicao pos : posicaoList) {
21                System.out.println("IdPacote: " + pos.getIdPacote());
22                System.out.println("IDVeiculo: " + pos.getIdVeiculo());
23                System.out.println("DataPacote: " + pos.getDataPacote());
24            }
25        } catch (Exception e) {
26
27        }
28    }
29 }
30
31
32
33 }
```

Na linha 19, o método `Main` realiza a conexão com o web service. Após isto, é realizado o disparo para a leitura de 300 pacotes de posições, conforme o método acima.

Na linha 23 é realizada a chamada do método `ObterPacotePosicoes`, onde deve ser informado o usuário e senha do cliente e a quantidade de pacotes de posições que se deseja receber no XML de retorno.

O objeto `posicao` pode ser iterado dentro do laço `for`, para seu devido tratamento — por exemplo, gravar cada posição contida no objeto no banco de dados, entre outras possibilidades.