

# API de Integração

- [Api de Rastreamento](#)
- [Webservice de Roteirização](#)
- [FAQ - Webservice Integração - Importação de Pedidos](#)
- [FAQ - Webservice Integração - Importação de Despesas](#)
- [FAQ - Webservice Integração - Automatizador de Roteirização](#)

# Api de Rastreamento

## 1. Introdução

A API de Rastreamento é uma interface de programação que permite o acesso programático aos dados e funcionalidades do sistema de gerenciamento de frota GPSCorporativo. Esta API fornece endpoints REST para autenticação, monitoramento de veículos em tempo real, recuperação de históricos de percursos, telemetria avançada, envio de mensagens para dispositivos de bordo, e controle de atuadores do veículo.

A API utiliza o formato JSON para comunicação de dados e segue o padrão REST com métodos HTTP como GET e POST. Ela requer autenticação via login e implementa controles anti-abuso que limitam a frequência de requisições.

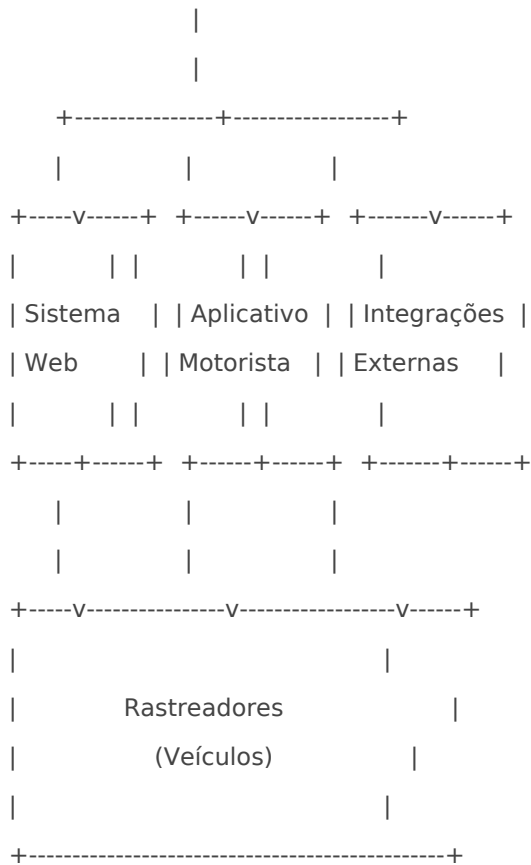
Esta documentação descreve uma coleção do Postman denominada "Testes", que contém requisições para interagir com a API do sistema de rastreamento GPSCorporativo. A coleção foi projetada para facilitar o teste e a integração com os serviços de rastreamento, monitoramento e comunicação com veículos através de uma interface REST.

## Informações da Coleção do Postman

- **Nome da Coleção:** Testes
- **ID Postman:** 65270104-d3fd-4853-a9e9-6f3062fbfb22
- **Esquema:** <https://schema.getpostman.com/json/collection/v2.1.0/collection.json>
- **Servidor Base:** rastreamento.concept.inf.br (definido como variável de ambiente)
- **Base Path:** /GPSCorporativo/rest/

## 2. Diagrama Conceitual do Sistema

```
+-----+
|       |
| Servidor API |
| Rastreamento |
|       |
+-----+
```



O diagrama acima ilustra como os diferentes componentes do sistema GPSCorporativo se comunicam entre si. A API de Rastreamento atua como interface central que permite a comunicação bidirecional entre:

1. O Sistema Web (usado pelos gestores de frota)
2. O Aplicativo do Motorista (usado pelos condutores dos veículos)
3. Integrações Externas (sistemas de terceiros)
4. Os Rastreadores dos Veículos (que enviam dados de telemetria e recebem comandos)

Este fluxo de comunicação permite tanto o monitoramento quanto o controle remoto da frota.

### 3. Glossário Técnico

Termo	Definição
<b>Rotograma</b>	Planejamento detalhado de rota que inclui pontos de parada, tempos estimados e distâncias a percorrer.
<b>Itinerário</b>	Sequência ordenada de locais a serem visitados pelo veículo, com horários previstos.
<b>Roteirização</b>	Processo de planejamento e otimização de rotas para uma frota de veículos.

Termo	Definição
<b>Telemetria</b>	Conjunto de dados técnicos transmitidos pelo veículo, como velocidade, RPM, nível de combustível, etc.
<b>Atuador</b>	Dispositivo que controla mecanismos do veículo remotamente (ex: bloqueio de motor).
<b>Ponto de Interesse (POI)</b>	Local geograficamente definido que possui relevância para a operação.
<b>Teclado</b>	Dispositivo de bordo que permite comunicação bidirecional entre a central e o motorista.
<b>JSESSIONID</b>	Token de sessão usado para autenticação nas requisições após o login inicial.
<b>Avisos</b>	Notificações automáticas geradas pelo sistema (ex: excesso de velocidade, desvio de rota).
<b>Identificador de Posição</b>	Código único que identifica um registro específico de localização no sistema.

## 4. Arquitetura e Características Técnicas

### 4.1. Características Gerais

- **Tipo:** REST API
- **Formato de Dados:** JSON
- **Autenticação:** Baseada em sessão (JSESSIONID)
- **Tempo da sessão:** 120 segundos. Após 120 segundos, a sessão é expirada e precisa de uma nova autenticação.
- **Controle de Taxa:** Implementado em todos os endpoints (rate limiting)
- **Versão Atual:** 1.5 (Setembro 2024)

### 4.2. Controle Anti-Abuso

A API implementa um mecanismo de controle anti-abuso que limita as requisições por usuário:

- Armazena timestamps das últimas requisições em um hashmap
- Verifica o intervalo entre requisições (varia de 5 a 60 segundos dependendo do endpoint)
- Retorna mensagem de erro se novas requisições forem feitas dentro do intervalo mínimo

## 4.3. Fluxo de Trabalho da API

1. **Autenticação:** O cliente deve primeiro se autenticar usando o endpoint de autenticação
2. **Obtenção do Token de Sessão:** Após autenticação bem-sucedida, um JSESSIONID é retornado
3. **Chamadas às APIs:** O cliente pode fazer chamadas aos diversos endpoints usando o JSESSIONID
4. **Controle de Sessão:** A sessão tem duração limitada e pode precisar ser renovada

# 5. Endpoints e Operações

## 5.1. Autenticação

**Descrição:** Autentica o usuário no sistema de rastreamento, obrigatório para utilização dos demais métodos.

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/autenticar>

**Método:** POST

**Content-Type:** application/x-www-form-urlencoded

**Parâmetros:**

- `login` (String): CPF ou CNPJ do cliente
- `senha` (String): Senha de acesso ao sistema
- `tipoAcesso` (String): Tipo de acesso ("master" ou "grupo")

**Validações:**

- Login precisa ser um CPF ou CNPJ válido
- Intervalos entre requisições deve ser superior a 5 segundos
- Intervalos entre autenticações do mesmo usuário deve ser superior a 120 segundos

**Retorno em caso de sucesso:**

```
{
  "id": 123,
  "nome": "Nome do Cliente",
  "idVeiculo": 456,
  "idGrupoVeiculo": 789,
```

```
"mensagemParaCliente": "Mensagem opcional para o cliente",
"jsessionId": "ABC123XYZ"
}
```

#### Retorno em caso de erro:

```
{
  "id": 0
}
```

#### Instruções passo a passo:

1. Prepare os parâmetros `login`, `senha` e `tipoAcesso`
2. Envie uma requisição POST para o endpoint de autenticação
3. Armazene o valor de `jsessionId` retornado para utilizar nas próximas requisições
4. Inclua o `jsessionId` como cookie HTTP em todas as requisições subsequentes

## 5.2. Posição Atual dos Veículos

**Descrição:** Retorna a posição atual de todos os veículos associados ao usuário autenticado.

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/rastrearVeiculos>

**Método:** GET

**Content-Type:** application/x-www-form-urlencoded

**Parâmetros:** Nenhum (utiliza o JSESSIONID para identificação)

#### Validações:

- Intervalo entre requisições deve ser superior a 30 segundos
- Usuário deve estar autenticado

#### Retorno em caso de sucesso:

```
[
  {
    "idVeiculo": "123",
    "latitude": "-19.9322",
    "longitude": "-43.9352",
    "ignicao": "Ligado",
    "modelo": "F-4000",
```

```
"tipoVeiculo": "CAMINHAOA",
"velocidadeAtual": "60",
"dataUltimaAtualizacao": "10:25 12/03/2025",
"tempoParado": "00:00:00",
"condutor": "João da Silva",
"cpfCondutor": "123.456.789-00",
"placa": "ABC1234",
"ultimaLeituraSensorRS232": "Sensor data",
"ultimoOdometro": "12345",
"ultimoHorimetro": "987",
"statusEntrada1": "false",
"statusEntrada2": "true",
"statusEntrada3": "false",
"statusEntrada4": "false",
"statusSaida1": "true",
"statusSaida2": "false",
"statusSaida3": "false"
}
]
```

#### Retorno em caso de erro:

```
{
  "codigo": 99
}
```

#### Instruções passo a passo:

1. Certifique-se de estar autenticado e possuir um JSESSIONID válido
2. Envie uma requisição GET para o endpoint
3. Processe a lista de veículos retornada
4. Respeite o intervalo mínimo entre requisições (30 segundos)

## 5.3. Percursos Anteriores

**Descrição:** Recupera o histórico de percurso de um veículo específico em um intervalo de datas.

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/rastrearVeiculoPorData>

**Método:** POST

**Content-Type:** application/x-www-form-urlencoded

## Parâmetros:

- `idVeiculo` (String): Identificador do veículo
- `dataHoraInicio` (String): Data e hora de início no formato YYYYMMDDHHmmss
- `dataHoraFim` (String): Data e hora de fim no formato YYYYMMDDHHmmss
- `somenteParadas` (String, opcional): "true" para retornar apenas paradas

## Validações:

- Intervalo entre requisições deve ser superior a 60 segundos
- `idVeiculo` deve ser um número inteiro válido
- Datas devem estar no formato YYYYMMDDHHmmss
- `dataHoraInicio` deve ser anterior a `dataHoraFim`
- O intervalo máximo permitido é de 7 dias

## Retorno em caso de sucesso:

```
[
  {
    "indice": "1",
    "latitude": "-19.9322",
    "longitude": "-43.9352",
    "dataHoraGPS": "10:25 12/03/2025",
    "velocidadeAtual": "60",
    "odometro": "12345",
    "horimetro": "987",
    "tempoParado": "00:00:00",
    "ignicao": "Ligado",
    "descricaoPol": "Nome do ponto de interesse"
  }
]
```

## Retorno em caso de erro:

```
{
  "mensagem": "Mensagem de erro específica"
}
```

## Instruções passo a passo:

1. Prepare os parâmetros `idVeiculo`, `dataHoraInicio` e `dataHoraFim`
2. Envie uma requisição POST para o endpoint
3. Processe a lista de pontos de rastreamento retornada

4. Atenção ao limite de 7 dias para o intervalo de pesquisa

## 5.4. Consulta de Posição por Identificador

**Descrição:** Recupera registros de rastreamento a partir de um identificador de posição para um ou mais veículos.

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/rastrearPorIdentificador>

**Método:** POST

**Content-Type:** application/x-www-form-urlencoded

### Parâmetros:

- `idVeiculo` (String, opcional): Identificador do veículo
- `identificadorPosicao` (String): Identificador único da posição

### Validações:

- Intervalo entre requisições deve ser superior a 30 segundos
- `identificadorPosicao` deve ser fornecido
- Se `idVeiculo` for fornecido, deve ser um número inteiro

### Retorno em caso de sucesso:

```
[
  [
    {
      "indice": "1",
      "idVeiculo": "123",
      "identificadorPos": "789012345",
      "latitude": "-19.9322",
      "longitude": "-43.9352",
      "dataRecebimentoGMT_3": "10:25 12/03/2025",
      "dataHoraGPSGMT_3": "10:24 12/03/2025",
      "velocidade": "60",
      "ignicao": "Ligado",
      "odometro": "12345",
      "horimetro": "987",
      "orientacao": "180"
    }
  ]
]
```

```
] ]
```

### Retorno em caso de erro:

```
{  
  "mensagem": "Mensagem de erro específica"  
}
```

### Instruções passo a passo:

1. Prepare o parâmetro `identificadorPosicao` (e opcionalmente `idVeiculo`)
2. Envie uma requisição POST para o endpoint
3. Processe a matriz de pontos de rastreamento retornada
4. Observe que o retorno é uma matriz de matrizes, com cada matriz interna representando um veículo

## 5.5. Consulta de Dados de Telemetria

**Descrição:** Recupera dados detalhados de telemetria de um veículo em um intervalo de datas.

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/listarTelemetriaPorData>

**Método:** POST

**Content-Type:** application/x-www-form-urlencoded

### Parâmetros:

- `idVeiculo` (String): Identificador do veículo
- `dataHoraInicio` (String): Data e hora de início no formato YYYYMMDDHHmmss
- `dataHoraFim` (String): Data e hora de fim no formato YYYYMMDDHHmmss

### Validações:

- Intervalo entre requisições deve ser superior a 30 segundos
- Todos os parâmetros são obrigatórios
- `idVeiculo` deve ser um número inteiro
- Datas devem estar no formato YYYYMMDDHHmmss
- `dataHoraInicio` deve ser anterior a `dataHoraFim`
- O intervalo máximo permitido é de 7 dias

### Retorno em caso de sucesso:

```
[
  {
    "indice": "1",
    "latitude": "-19.9322",
    "longitude": "-43.9352",
    "dataHoraGPS": "10:25 12/03/2025",
    "velocidadeAtual": "60",
    "odometro": "12345",
    "horimetro": "987",
    "ignicao": "Ligado",
    "rpm": "1800",
    "temperaturaMotor": "90",
    "combustivelUtilizado": "1.2",
    "nivelCombustivel": "45",
    "nivelCombustivelPercentual": "75",
    "pressaoPedalAcelerador": "80",
    "tempoEmMovimentoTotal": "120",
    "totalCombustivelUtilizado": "230.5",
    "totalCombustivelUtilizadoParado": "12.3",
    "pesoEixos": "2500",
    "informacaoTacografo": "Info do tacógrafo",
    "indicadores": "Indicadores diversos",
    "portaMotoristaAberta": "false",
    "portaPassageiroAberta": "false",
    "portaTraseiraEsquerdaAberta": "false",
    "portaTraseiraDireitaAberta": "false",
    "portaMalasAberto": "false",
    "caputAberto": "false",
    "cintoSegurancaLigado": "true",
    "arCondicionadoLigado": "true",
    "farolLigado": "true",
    "isExcessoVelocidade": "false",
    "corHexaFaixaRPM": "#00FF00",
    "pressaoOleoMotor": "3.5"
  }
]
```

**Retorno em caso de erro:**

```
{
  "mensagem": "Mensagem de erro específica"
}
```

### Instruções passo a passo:

1. Prepare os parâmetros `idVeiculo`, `dataHoraInicio` e `dataHoraFim`
2. Envie uma requisição POST para o endpoint
3. Processe a lista de dados de telemetria retornada
4. Atenção ao limite de 7 dias para o intervalo de pesquisa

## 5.6. Listar Avisos por Veículo e Data

**Descrição:** Lista os avisos gerados por um veículo em uma data específica.

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/listarAvisos>

**Método:** POST

**Content-Type:** application/x-www-form-urlencoded

### Parâmetros:

- `idVeiculo` (String): Identificador do veículo
- `data` (String): Data no formato YYYYMMDDHHmmss

### Validações:

- Intervalo entre requisições deve ser superior a 5 segundos
- Ambos os parâmetros são obrigatórios

### Retorno em caso de sucesso:

```
{
  "lista": [
    {
      "id": "123",
      "tipo": "ExcessoVelocidade",
      "tipoDescricao": "Excesso de Velocidade",
      "condutor": "João da Silva",
      "dataHora": "10:25 12/03/2025",
      "latitude": "-19.9322",
      "longitude": "-43.9352",
    }
  ]
}
```

```
"descricao": "Veículo ultrapassou o limite de velocidade",  
"dataHoraEnvioEmail": "10:26 12/03/2025"  
}  
]  
}
```

#### Retorno em caso de erro:

```
{  
  "codigo": 99  
}
```

#### Instruções passo a passo:

1. Prepare os parâmetros `idVeiculo` e `data`
2. Envie uma requisição POST para o endpoint
3. Processe a lista de avisos retornada
4. O retorno inclui um resumo e detalhes dos avisos no período

## 5.7. Acionar Atuadores

**Descrição:** Controla atuadores do veículo (bloqueio, saída 2 e saída 3).

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/acionarAtuadores>

**Método:** POST

**Content-Type:** application/x-www-form-urlencoded

#### Parâmetros:

- `idVeiculo` (String): Identificador do veículo
- `comando` (String): Comando a ser enviado
- `senhaContato` (String): Senha de contato para autorização

#### Comandos disponíveis:

- Saída 1 (Bloqueio): "BLOQUEAR" / "DESBLOQUEAR"
- Saída 2: "SPC\_OUTPUT2\_ACTIVATE" / "SPC\_OUTPUT2\_DEACTIVATE"
- Saída 3: "SPC\_OUTPUT3\_ACTIVATE" / "SPC\_OUTPUT3\_DEACTIVATE"

#### Validações:

- Intervalo entre requisições deve ser superior a 5 segundos

- Todos os parâmetros são obrigatórios
- Comando deve ser um dos valores permitidos

#### Retorno em caso de sucesso:

```
{
  "mensagem": "Comando enviado com sucesso"
}
```

#### Retorno em caso de erro:

```
{
  "codigo": 99
}
```

#### Instruções passo a passo:

1. Prepare os parâmetros `idVeiculo`, `comando` e `senhaContato`
2. Envie uma requisição POST para o endpoint
3. Verifique a mensagem de retorno para confirmar o envio
4. Aguarde pelo menos 5 segundos antes de enviar outro comando

## 5.8. Listar Mensagens do Teclado

**Descrição:** Recupera mensagens trocadas com o dispositivo de bordo (teclado).

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/listarMensagensTeclado>

**Método:** POST

**Content-Type:** application/x-www-form-urlencoded

#### Parâmetros:

- `idVeiculo` (String): Identificador do veículo
- `cpfMotorista` (String, opcional): CPF do motorista
- `dataInicioMensagem` (String, opcional): Data de início da mensagem (YYYYMMDDHHmmss)
- `dataFimMensagem` (String, opcional): Data de fim da mensagem (YYYYMMDDHHmmss)
- `dataInicioRecebidaMensagem` (String, opcional): Data de início do recebimento (YYYYMMDDHHmmss)
- `dataFimRecebidaMensagem` (String, opcional): Data de fim do recebimento (YYYYMMDDHHmmss)

#### Validações:

- Intervalo entre requisições deve ser superior a 5 segundos
- `idVeiculo` é obrigatório
- Pelo menos um conjunto de datas (mensagem ou recebimento) deve ser informado
- Datas devem estar no formato correto

#### Retorno em caso de sucesso:

```
[
  {
    "placa": "ABC1234",
    "dataHora": "10:25 12/03/2025",
    "codigo": "123",
    "mensagem": "Texto da mensagem",
    "origem": "Motorista",
    "statusComandoMensagem": "Entregue",
    "odometroMensagem": "12345",
    "horimetroMensagem": "987",
    "latitudeMensagem": "-19.9322",
    "longitudeMensagem": "-43.9352",
    "localizacaoMensagem": "Av. Amazonas, 1234",
    "cpfMotorista": "123.456.789-00",
    "nomeMotorista": "João da Silva",
    "codigoidentificacaoLeitor": "XYZ123",
    "matriculaMotorista": "M1234"
  }
]
```

#### Retorno em caso de erro:

```
{
  "mensagem": "Mensagem de erro específica"
}
```

#### Instruções passo a passo:

1. Prepare o parâmetro `idVeiculo` e pelo menos um conjunto de datas
2. Envie uma requisição POST para o endpoint
3. Processe a lista de mensagens retornada
4. Observe que o veículo precisa estar associado ao cliente autenticado

## 5.9. Enviar Mensagem para Teclado

**Descrição:** Envia uma mensagem para o dispositivo de bordo (teclado) do veículo.

**URL:** <http://rastreamento.concept.inf.br/GPSCorporativo/rest/enviarMensagemTeclado>

**Método:** POST

**Content-Type:** application/x-www-form-urlencoded

**Parâmetros:**

- `idVeiculo` (String): Identificador do veículo
- `mensagem` (String): Texto da mensagem a ser enviada

**Validações:**

- Intervalo entre requisições deve ser superior a 5 segundos
- Ambos os parâmetros são obrigatórios

**Retorno em caso de sucesso:**

```
{
  "mensagem": "Mensagem enviada com sucesso"
}
```

**Retorno em caso de erro:**

```
{
  "codigo": 99
}
```

**Instruções passo a passo:**

1. Prepare os parâmetros `idVeiculo` e `mensagem`
2. Envie uma requisição POST para o endpoint
3. Verifique a mensagem de retorno para confirmar o envio
4. Aguarde pelo menos 5 segundos antes de enviar outra mensagem

## 6. Regras de Negócio e Validações

### 6.1. Regras Gerais

1. **Autenticação e Autorização:**

- Usuário deve estar autenticado para acessar qualquer endpoint (exceto o de autenticação)
  - Usuário só pode acessar veículos associados à sua conta
  - Existem dois tipos de acesso: "master" e "grupo"
- 2. Controle de Taxa de Requisições:**
    - Cada endpoint tem um limite de frequência de chamadas
    - O IP e o login do usuário são usados para identificação
    - Requisições feitas antes do intervalo mínimo são rejeitadas
  - 3. Validação de Datas:**
    - Datas devem seguir o formato YYYYMMDDHHmmss
    - Data inicial deve ser anterior à data final
    - Intervalo máximo em consultas históricas é de 7 dias
    - Datas são armazenadas e processadas em UTC mas exibidas em GMT-3
  - 4. Formato de IDs e Parâmetros:**
    - IDs de veículos devem ser números inteiros
    - Identificadores de posição devem ser números inteiros ou strings
    - Comandos de atuadores devem ser um dos valores predefinidos

## 6.2. Validações Específicas

- 1. Bloqueio de Veículo:**
  - Requer senha de contato para autorização
  - A operação é registrada em log para auditoria
  - Limitado a uma operação a cada 5 segundos
- 2. Mensagens para Teclado:**
  - Tamanho da mensagem pode ser limitado pelo dispositivo de bordo
  - Enviadas em tempo real quando o veículo está conectado
- 3. Consulta de Histórico:**
  - Opção para filtrar apenas paradas (economiza dados)
  - Limite de 7 dias para evitar sobrecarga do sistema

# 7. Troubleshooting e Resolução de Problemas

## 7.1. Diagnóstico de Problemas Comuns

Problema	Possíveis Causas	Soluções
----------	------------------	----------

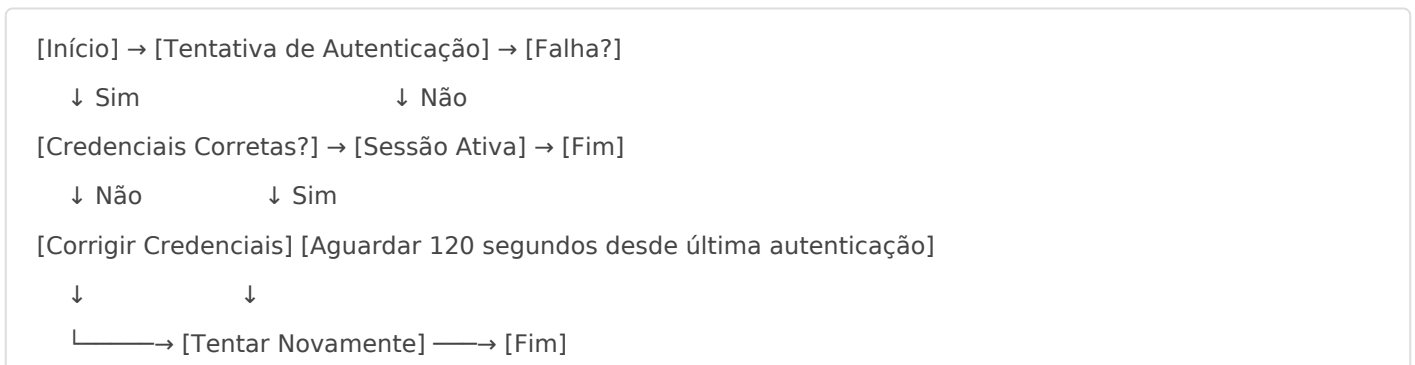
<b>Autenticação sempre retorna 0</b>	<ol style="list-style-type: none"> <li>1. CPF ou CNPJ errados</li> <li>2. Senha incorreta</li> <li>3. Não informou os pontos, hífen e barra no CPF ou CNPJ</li> </ol>	<ol style="list-style-type: none"> <li>1. Informar o CPF e CNPJ com pontos, hífen e barra</li> <li>2. Confirmar a senha na opções da plataforma WEB em Cofnguração -&gt; Controle de Acesso</li> <li>3. Verificar se o tipo usado é grupo para as senhas criadas no controle de aceso</li> </ol>
--------------------------------------	---	--

## 7.2. Tabela de Códigos de Erro

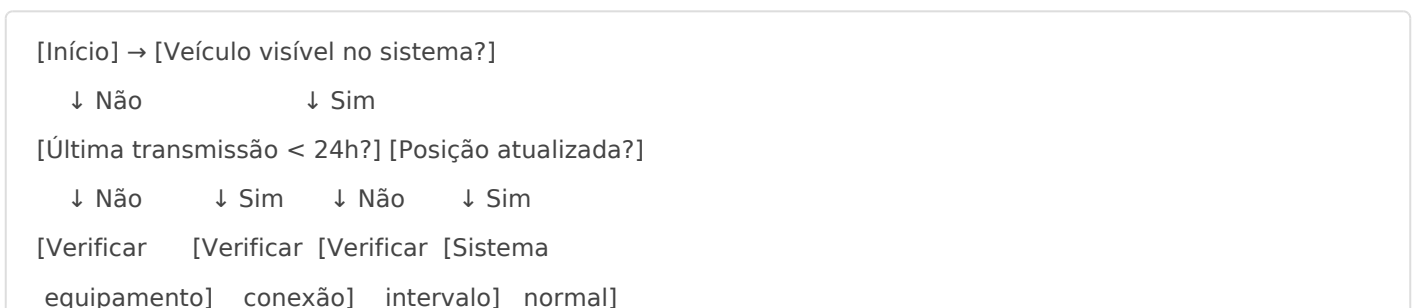
Código	Descrição	Procedimento de Resolução
99	Usuário não autenticado ou sessão expirada	Realizar nova autenticação através do endpoint correspondente
101	Múltiplas requisições em intervalo proibido	Aguardar o tempo mínimo entre requisições para o endpoint específico
102	Parâmetros inválidos ou ausentes	Verificar documentação e corrigir parâmetros da requisição

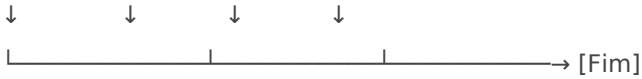
## 7.3. Fluxogramas de Decisão

### 7.3.1. Diagnóstico de Falhas de Autenticação



### 7.3.2. Resolução de Problemas de Rastreamento





## 8. Versões

Versão	Data de Lançamento	Principais Mudanças
1.5	Setembro 2024	Adicionado suporte para sensores adicionais de telemetria e melhorias na segurança
1.4	Março 2024	Implementada telemetria avançada e relatórios de consumo de combustível
1.3	Outubro 2023	Adicionado controle remoto de atuadores e comunicação bidirecional
1.2	Mai 2023	Melhorias na performance e suporte a múltiplos veículos por requisição
1.1	Janeiro 2023	Correções de bugs e expansão da documentação
1.0	Setembro 2022	Versão inicial com funcionalidades básicas de rastreamento

## 9. Limitações Conhecidas do Sistema

- Intervalo de Consultas Históricas:** Consultas de histórico são limitadas a um período máximo de 7 dias para evitar sobrecarga do sistema.
- Frequência de Atualização:** A posição atual dos veículos é atualizada conforme a configuração do dispositivo, tipicamente entre 30 segundos e 5 minutos.
- Conexão com Dispositivos:** Comandos enviados para veículos só são processados quando o dispositivo está online e com conectividade.
- Tamanho de Mensagens:** Mensagens enviadas para o teclado do motorista são limitadas a 160 caracteres por limitações de hardware.
- Simultaneidade de Comandos:** O sistema não permite o envio simultâneo de múltiplos comandos para o mesmo veículo. É necessário aguardar a confirmação do primeiro comando.
- Retenção de Dados:** Os dados históricos detalhados ficam disponíveis por até 5 anos. Depende do plano contratado pelo cliente. Após esse período, os dados são apagados.

7. **Precisão Geográfica:** A precisão da localização depende do dispositivo GPS instalado no veículo e das condições ambientais, podendo variar entre 2 e 15 metros.
8. **Compatibilidade de Navegadores:** A interface web é otimizada para Chrome, Firefox e Edge. Outros navegadores podem apresentar limitações visuais.
9. **Atuação em Áreas sem Cobertura:** Veículos em áreas sem cobertura móvel não receberão comandos até retornarem a uma área com sinal.

## 10. Perguntas Frequentes (FAQ)

1. **Como obtenho o identificador (ID) de um veículo?**
  - O ID do veículo pode ser obtido através do endpoint "Posição Atual dos Veículos" ou ao realizar a autenticação, caso o usuário tenha um veículo único associado.
2. **Quanto tempo dura uma sessão após autenticação?**
  - A sessão tem duração de 120 segundos conforme documentado na seção 4.1. Após esse período, é necessário autenticar-se novamente.
3. **Por que recebo um erro de "múltiplas requisições"?**
  - Cada endpoint tem um limite de frequência de chamadas. Você deve respeitar o intervalo mínimo entre requisições (de 5 a 60 segundos, dependendo do endpoint).
4. **Como enviar um comando de bloqueio/desbloqueio de veículo?**
  - Utilize o endpoint "Acionar Atuadores" com o parâmetro `comando` igual a "BLOQUEAR" ou "DESBLOQUEAR" e forneça a senha de contato para autorização.
5. **Quais formatos de data são aceitos pela API?**
  - A API aceita apenas datas no formato YYYYMMDDHHmmss (exemplo: 20250312102530 para 12/03/2025 10:25:30).
6. **Por que minha consulta de histórico retorna "Rota máxima pesquisável é no intervalo de 7 dias"?**
  - Para evitar sobrecarga do sistema, as consultas de histórico estão limitadas a um intervalo máximo de 7 dias.
7. **O que significa o código de erro 99?**
  - O código 99 indica que o usuário não está autenticado ou que a sessão expirou.
8. **Como faço para consultar os avisos de um veículo?**
  - Utilize o endpoint "Listar Avisos por Data" fornecendo o ID do veículo e a data desejada.
9. **Posso monitorar múltiplos veículos simultaneamente?**
  - Sim, o endpoint "Posição Atual dos Veículos" retorna informações de todos os veículos associados ao usuário autenticado.
10. **Quais dados de telemetria estão disponíveis?**
  - A API fornece dados detalhados como velocidade, RPM, temperatura do motor, nível de combustível, estado das portas, uso do cinto de segurança, entre outros.
11. **O que fazer quando um veículo não aparece no mapa?**
  - Verifique se o dispositivo está ligado, se o veículo tem permissão associada ao seu usuário e consulte a última transmissão para determinar se há problemas de conectividade.

12. **Como resolver erros de sincronização no aplicativo do motorista?**
  - Verifique a versão do aplicativo, a conexão de internet do dispositivo e tente limpar o cache. Se persistir, reinstale a aplicação.
13. **Por que o roteirizador não está considerando todas as restrições configuradas?**
  - Verifique se não há conflitos entre restrições e se todas estão corretamente configuradas. Algumas restrições podem ter precedência sobre outras.
14. **Como posso saber se um comando foi recebido pelo veículo?**
  - Os comandos enviados são registrados no sistema e você pode verificar o status através do histórico de comandos na interface web ou consultando os logs do sistema.

# 11. Exemplos de Implementação

## 11.1. Exemplo de Rastreamento em Tempo Real

```
// Exemplo de código JavaScript para rastreamento em tempo real
async function monitorarVeiculosEmTempoReal() {
  // 1. Autenticar no sistema
  const credenciais = {
    login: "empresa@exemplo.com.br",
    senha: "senhaSegura123",
    tipoAcesso: "grupo"
  };

  const respAuth = await fetch('http://rastreamento.concept.inf.br/GPSCorporativo/rest/autenticar', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/x-www-form-urlencoded'
    },
    body: new URLSearchParams(credenciais)
  });

  const authData = await respAuth.json();
  const jsessionId = authData.jsessionId;

  // 2. Configurar intervalo para atualização a cada 30 segundos
```

```

setInterval(async () => {
  // 3. Obter posição atual dos veículos
  const respVeiculos = await fetch('http://rastreamento.concept.inf.br/GPSCorporativo/rest/rastrearVeiculos',
  {
    method: 'GET',
    headers: {
      'Cookie': `JSESSIONID=${jsessionId}`
    }
  });

  const veiculos = await respVeiculos.json();

  // 4. Processar e exibir os dados
  veiculos.forEach(veiculo => {
    console.log(`Veículo ${veiculo.placa} - Posição: ${veiculo.latitude},${veiculo.longitude}`);
    console.log(`Velocidade: ${veiculo.velocidadeAtual} km/h - Status: ${veiculo.ignicao}`);

    // Aqui você adicionaria código para atualizar um mapa ou interface
  });
}, 30000); // 30 segundos
}

monitorarVeiculosEmTempoReal();

```

## 11.2. Exemplo de Análise Histórica

```

// Exemplo de código JavaScript para análise histórica de percurso
async function analisarPercursoHistorico(idVeiculo, dataInicio, dataFim) {
  // 1. Autenticar no sistema
  const credenciais = {
    login: "empresa@exemplo.com.br",
    senha: "senhaSegura123",
    tipoAcesso: "grupo"
  };

  const respAuth = await fetch('http://rastreamento.concept.inf.br/GPSCorporativo/rest/autenticar', {
    method: 'POST',
    headers: {

```

```
        'Content-Type': 'application/x-www-form-urlencoded'
    },
    body: new URLSearchParams(credenciais)
  });

const authData = await respAuth.json();
const jsessionId = authData.jsessionId;

// 2. Consultar histórico de percurso
const params = {
  idVeiculo: idVeiculo,
  dataHoraInicio: dataInicio, // Formato: YYYYMMDDHHmmss
  dataHoraFim: dataFim,      // Formato: YYYYMMDDHHmmss
  somenteParadas: "false"
};

const respHistorico = await
fetch('http://rastreamento.concept.inf.br/GPSCorporativo/rest/rastrearVeiculoPorData', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded',
    'Cookie': `JSESSIONID=${jsessionId}`
  },
  body: new URLSearchParams(params)
});

const historico = await respHistorico.json();

// 3. Processar os dados históricos
let distanciaTotal = 0;
let tempoMovimento = 0;
let tempoParado = 0;

for (let i = 1; i < historico.length; i++) {
  const pontoAtual = historico[i];
  const pontoAnterior = historico[i-1];

  // Calcular distância entre pontos (exemplo simplificado)
  const distancia = calcularDistanciaEntrePontos(
```

```

        pontoAnterior.latitude, pontoAnterior.longitude,
        pontoAtual.latitude, pontoAtual.longitude
    );

    distanciaTotal += distancia;

    // Analisar tempos de movimento e parada
    if (pontoAtual.velocidadeAtual > 0) {
        tempoMovimento++;
    } else {
        tempoParado++;
    }
}

// 4. Retornar análise
return {
    idVeiculo: idVeiculo,
    periodo: {
        inicio: dataInicio,
        fim: dataFim
    },
    pontos: historico.length,
    distanciaTotal: distanciaTotal.toFixed(2) + " km",
    tempoMovimento: formatarTempo(tempoMovimento),
    tempoParado: formatarTempo(tempoParado),
    velocidadeMedia: (distanciaTotal / tempoMovimento * 3600).toFixed(2) + " km/h"
};
}

// Funções auxiliares
function calcularDistanciaEntrePontos(lat1, lon1, lat2, lon2) {
    // Implementação da fórmula de Haversine para cálculo de distância
    // entre coordenadas geográficas
    // Código simplificado para o exemplo
    return 0.5; // km
}

function formatarTempo(minutos) {
    const horas = Math.floor(minutos / 60);

```

```
const min = minutos % 60;
return `${horas}h ${min}min`;
}

// Exemplo de uso
analisarPercursoHistorico(
  "3375",
  "20250310000000", // 10/03/2025 00:00:00
  "20250310235959" // 10/03/2025 23:59:59
).then(analise => {
  console.log("Análise do percurso histórico:", analise);
});
```

## 12. Referências Cruzadas

Componente	Documentação Relacionada	Seção Relevante
Sistema Web	Manual do Sistema Web	Capítulo 4: Monitoramento de Frota
Aplicativo do Motorista	Guia do Aplicativo	Seção 2.3: Comunicação com a Central
Roteirizador	Manual do Roteirizador	Capítulo 5: Integração com Rastreamento
API de Integração	Documentação de Integração	Seção 3.1: Autenticação e Segurança

## Vínculos entre Componentes e Funcionalidades

- **Monitoramento em Tempo Real:** Para implementações avançadas de monitoramento em tempo real, consulte a seção 5.2 deste documento e a seção 4.2 do Manual do Sistema Web.
- **Roteirização com Rastreamento:** Para otimizar rotas com base em dados históricos, consulte a seção 5.3 deste documento e o Capítulo 3 do Manual do Roteirizador.
- **Comunicação com Motoristas:** Para configurar mensagens e alertas automáticos, consulte as seções 5.8 e 5.9 deste documento e a Seção 2.3 do Guia do Aplicativo.
- **Telemetria e Manutenção Preventiva:** Para implementar manutenção baseada em dados de telemetria, consulte a seção 5.5 deste documento e o Capítulo 6 do Manual do Sistema Web.

# 13. Representações Estruturadas das Entidades

## 13.1. Estrutura de Veículo (JSON)

```
{
  "id": 3375,
  "placa": "ABC1234",
  "tipo": "CAMINHAOA",
  "modelo": "F-4000",
  "ano": 2022,
  "chassi": "9BFFT35P3CD123456",
  "capacidade": {
    "peso": 3500,
    "volume": 24,
    "passageiros": 3
  },
  "dispositivo": {
    "id": 98765,
    "modelo": "GP1500",
    "imei": "123456789012345",
    "dataInstalacao": "20230215"
  },
  "telemetria": {
    "sensores": [
      "rpm",
      "temperatura",
      "combustivel",
      "pressao_oleo",
      "cinto",
      "portas"
    ],
    "intervaloPadrao": 30,
    "intervaloParado": 300
  },
  "limites": {
```

```
"velocidadeMaxima": 80,  
"rpmMaximo": 2500,  
"horasOperacao": 12  
}  
}
```

## 13.2. Estrutura de Motorista (JSON)

```
{  
  "id": 1234,  
  "nome": "João da Silva",  
  "cpf": "123.456.789-00",  
  "matricula": "M1234",  
  "cnh": {  
    "numero": "12345678900",  
    "categoria": "D",  
    "validade": "20260531"  
  },  
  "contato": {  
    "telefone": "11999998888",  
    "email": "joao.silva@email.com"  
  },  
  "acessoApp": {  
    "login": "joao.silva",  
    "ultimoAcesso": "20250311103045",  
    "versaoApp": "2.3.7"  
  },  
  "veiculoAtual": 3375,  
  "status": "EmRota"  
}
```

## 13.3. Estrutura de Rota (JSON)

```
{  
  "id": 5678,  
  "nome": "Entrega Centro - 12/03/2025",  
  "veiculo": 3375,  
  "motorista": 1234,
```

```
"status": "EmAndamento",
"criacao": "20250311153045",
"inicio": {
  "planejado": "20250312080000",
  "real": "20250312075532"
},
"fim": {
  "planejado": "20250312180000",
  "real": null
},
"paradas": [
  {
    "ordem": 1,
    "cliente": "Empresa A",
    "endereco": "Av. Amazonas, 1234",
    "coordenadas": {
      "latitude": -19.9322,
      "longitude": -43.9352
    },
    "horario": {
      "planejado": "20250312090000",
      "previsto": "20250312092500",
      "real": "20250312093012"
    },
    "status": "Concluido"
  },
  {
    "ordem": 2,
    "cliente": "Empresa B",
    "endereco": "Rua dos Carijós, 456",
    "coordenadas": {
      "latitude": -19.9182,
      "longitude": -43.9372
    },
    "horario": {
      "planejado": "20250312110000",
      "previsto": "20250312114500",
      "real": null
    },
    "status": "EmAndamento"
```

```
}  
],  
"metricas": {  
  "distanciaPlanejada": 28.5,  
  "distanciaPercorrida": 15.7,  
  "consumoPlanejado": 8.5,  
  "consumoReal": 9.2  
}  
}
```

# Webservice de Roteirização

## Introdução

A API da Concept Tecnologia oferece uma solução completa para gerenciamento de entregas, roteamento de veículos e controle de despesas de frota. Esta plataforma permite que empresas automatizem o processo de entrega, desde a importação de pedidos até a roteirização eficiente e o monitoramento das despesas associadas aos veículos.

O sistema foi projetado para otimizar rotas de entrega, reduzir custos operacionais e aumentar a eficiência logística, utilizando tecnologias de geolocalização e algoritmos de roteirização para determinar os melhores caminhos para seus veículos de entrega, considerando diversos fatores como distância, tempo de atendimento e restrições de horários.

## Componentes Principais da API

A API é dividida em três componentes principais, cada um responsável por um aspecto diferente do processo logístico:

### 1. Importador de Pedidos

Este componente permite a inserção e consulta de pedidos no sistema. É o ponto de entrada para todos os dados de entregas que serão posteriormente roteirizados.

### 2. Roteirizador de Pedidos (Automatizador)

Responsável pela criação de rotas otimizadas, agrupando pedidos por região, veículo e outros critérios, determinando a sequência ideal de entregas.

### 3. Importador de Despesas

Gerencia os custos operacionais relacionados aos veículos, como abastecimentos, manutenções e outras despesas, permitindo um controle financeiro da frota.

# Detalhamento dos Componentes e Métodos

## 1. Importador de Pedidos (ImportadorPedidos)

**Endpoint WSDL:** <http://52.6.27.50:8181/importadorPedidos?wsdl>

### Métodos Disponíveis:

**importarPedidos:** Permite inserir novos pedidos no sistema para posterior roteirização.

- Recebe uma lista de objetos `PedidoRotaVO` contendo informações como número do pedido, endereço, coordenadas geográficas, data/hora, zona, loja, vendedor, entre outros.
- É possível também cadastrar pontos de interesse (locais específicos) durante a importação.

**consultarStatusEntregaPedido:** Verifica o status atual de um pedido no sistema.

- Informa se um pedido foi entregue, está em rota, pendente, ou teve problemas durante a entrega.

**converterEnderecoUsandoCEP:** Converte um CEP em coordenadas geográficas e endereço completo.

- Útil para pedidos que não possuem latitude/longitude definidas.

**listarItinerariosRoterizado:** Lista todos os itinerários (paradas) de uma rota já criada.

- Mostra a sequência de entregas, horários previstos e status.

**listarItinerariosLote:** Lista itinerários relacionados a um determinado número de lote.

**listarItinerariosCarregamento:** Lista itinerários relacionados a um carregamento específico.

**listarPoIRoterizado:** Lista os pontos de interesse incluídos em uma rota.

**atualizarNumeroCarregamento:** Atualiza o número de carregamento de uma viagem.

**listarPedidosPorStatus:** Recupera pedidos que estão em um determinado status (ex: separados, pendentes).

Métodos para alteração de status de pedidos:

- **alterarStatusVendaFinalizadoParaLiberadoSeparacao**
- **alterarStatusLiberadoSeparacaParaEmSeparacao**
- **alterarStatusEmSeparacaoParaSeparados**
- **alterarStatusSeparadosParaLiberadosRoteirizacao**
- **alterarStatusNaoEntreguesParaLiberadoSeparacao**
- **alterarStatusNaoEntreguesParaLiberadoRoteirizacao**

Métodos para cancelamento de status:

- **cancelar\_LiberadoRoteirizacao**
- **cancelar\_Separados**
- **cancelar\_EmSeparacao**
- **cancelar\_LiberadosSeparacao**

## 2. Roteirizador de Pedidos (AutomatizadorFachada)

**Endpoint WSDL:** <http://52.6.27.50:8181/automatizador?wsdl>

### Métodos Disponíveis:

**roteirizarPedidos:** Cria uma rota otimizada para entrega de pedidos.

- Recebe parâmetros como lista de carregamentos, ponto de origem, ponto de destino, veículo, motorista e data/hora de início.
- Utiliza algoritmos de otimização para definir a melhor sequência de entregas.
- Considera janelas de tempo, capacidade do veículo, tempo de atendimento e outros fatores.

**listarMotoristas:** Lista todos os motoristas cadastrados no sistema.

**listarVeiculos:** Lista todos os veículos disponíveis para roteirização.

**cadastrarZona:** Cadastra ou atualiza uma zona de entrega.

- As zonas são utilizadas para agrupar pedidos em regiões geográficas.

**cadaststrarLoja:** Cadastra ou atualiza uma loja no sistema.

**cadaststrarVendedor:** Cadastra ou atualiza um vendedor.

**cadaststrarMotorista:** Cadastra ou atualiza um motorista.

## 3. Importador de Despesas (ImportadorDespesas)

**Endpoint WSDL:** <http://52.6.27.50:8181/importadorDespesas?wsdl>

### Métodos Disponíveis:

**importarDespesas:** Permite inserir despesas relacionadas a veículos no sistema.

- Recebe informações como tipo de despesa, valor, quantidade, veículo, motorista, data/hora, etc.
- Suporta diversos tipos de despesas, incluindo abastecimentos, manutenções e outras.

**consultarDespesas:** Consulta despesas registradas por período e/ou veículo.

- Permite filtrar despesas por placa de veículo e intervalo de datas.

## Autenticação e Segurança

A autenticação na API é realizada por meio de três parâmetros presentes em praticamente todos os métodos:

1. **cpfCNPJ:** CPF ou CNPJ do cliente que está acessando a API. Deve estar cadastrado no sistema da central de rastreamento.
2. **senhaCliente:** Senha fornecida pela Concept Tecnologia para o cliente.
3. **senhaCentral:** Senha fornecida pela Concept Tecnologia para acesso à central.

É importante manter essas credenciais em segurança, utilizando variáveis de ambiente ou sistemas de gestão de segredos.

## Limitações e Boas Práticas

1. A API utiliza SOAP (Simple Object Access Protocol), requerendo bibliotecas específicas para sua integração.
2. Todas as datas devem ser enviadas no formato correto - para campos de data use objetos Date, para campos de hora use strings no formato "HH:MM".
3. Ao trabalhar com coordenadas geográficas, certifique-se de que os valores estão dentro dos limites válidos (latitude entre -90 e 90, longitude entre -180 e 180).
4. O sistema tem limitações na quantidade de pontos que podem ser roteirizados em uma única requisição (máximo de 23 pontos por chunk).
5. Sempre verifique o retorno das chamadas para garantir que a operação foi bem-sucedida.
6. Quando um pedido não tem coordenadas geográficas definidas, é possível obter essas informações através do código do ponto de interesse, endereço ou CEP.

## Fluxos de Integração Típicos

### Fluxo de Pedidos Completo

1. **Importação de Pedidos:** Utilize o método `importarPedidos` para inserir novos pedidos no sistema.
2. **Liberação para Separação:** Altere o status dos pedidos para "Liberado para Separação" usando `alterarStatusVendaFinalizadoParaLiberadoSeparacao`.
3. **Início de Separação:** Quando a separação dos produtos começar, altere o status para "Em Separação" com `alterarStatusLiberadoSeparacaParaEmSeparacao`.
4. **Finalização da Separação:** Após a separação completa, use `alterarStatusEmSeparacaoParaSeparados`.
5. **Liberação para Roteirização:** Quando os pedidos estiverem prontos para entrega, use `alterarStatusSeparadosParaLiberadosRoteirizacao`.
6. **Roteirização:** Utilize o método `roteirizarPedidos` para criar uma rota otimizada.
7. **Acompanhamento:** Use `consultarStatusEntregaPedido` ou `listarItinerariosRoterizado` para acompanhar o status de entrega.

### Fluxo de Gestão de Despesas

1. **Registro de Despesas:** Utilize o método `importarDespesas` para registrar despesas de veículos.
2. **Consulta de Despesas:** Use `consultarDespesas` para analisar os gastos por veículo ou período.

## Glossário de Termos

- **Pedido:** Solicitação de entrega de produtos a um cliente.
- **Carregamento:** Agrupamento de pedidos que serão entregues em uma mesma rota ou viagem.
- **Lote:** Agrupamento de pedidos para fins de controle interno.
- **Itinerário:** Cada parada (entrega) de uma rota.
- **Rota** ou **Viagem:** Sequência planejada de entregas para um veículo.
- **Ponto de Interesse (Pol):** Local geograficamente identificado, como endereço de cliente, depósito, etc.
- **Zona:** Região geográfica que agrupa pedidos para facilitar a roteirização.
- **Loja:** Estabelecimento a partir do qual os pedidos são despachados.
- **Vendedor:** Profissional responsável pela venda que gerou o pedido.
- **Status do Pedido:** Estado atual do pedido (finalizado, liberado para separação, em separação, separado, liberado para roteirização, etc.).
- **Tempo de Atendimento:** Tempo estimado para realizar a entrega em cada parada.
- **Janela de Entrega:** Intervalo de horário permitido para realização da entrega.

# Principais Objetos e Modelos de Dados

## PedidoRotaVO

Representa um pedido de entrega, contendo:

- **numeroPedido:** Identificador único do pedido
- **dataPedido:** Data em que o pedido foi realizado
- **horaPedido:** Hora em que o pedido foi realizado
- **valorPedido:** Valor monetário do pedido
- **endereco:** Endereço de entrega
- **latitude/longitude:** Coordenadas geográficas do endereço
- **codigoPontoInteresse:** Código do ponto de interesse (opcional)
- **descricao:** Descrição adicional do pedido
- **peso/volume:** Peso e volume total dos itens do pedido
- **horaEntregaInicial/horaEntregaFinal:** Janela de horário para entrega
- **tempoAtendimento:** Tempo estimado para realizar a entrega (em minutos)
- **zona/loja/vendedor:** Informações sobre zona de entrega, loja de origem e vendedor
- **numeroCarregamento:** Identificador do carregamento
- **numeroNotaFiscal:** Número da nota fiscal
- **numeroLote:** Identificador do lote
- **numeroGSM:** Número de celular para envio de alertas

- **dataCompromissoEntrega/horaCompromissoEntrega:** Data e hora prometidas para entrega
- **observacaoPedido:** Observações adicionais sobre o pedido

## ItinerarioViagemVO

Representa uma parada em uma rota de entrega:

- **numeroPedido:** Pedido associado à parada
- **dataItinerarioViagem:** Data prevista para a entrega
- **horaItinerarioViagem:** Hora prevista para a entrega
- **status:** Status atual da entrega (aberto, entregue, endereço fechado, etc.)
- **horaEntrega:** Hora em que a entrega foi realizada
- **latitude/longitude:** Localização da entrega
- **dataHoraCheckin/dataHoraCheckout:** Momento em que o motorista chegou e saiu do local
- **tempoAtendimento:** Tempo gasto na entrega

## ViagemVO

Representa uma rota completa de entregas:

- **id:** Identificador único da viagem
- **data:** Data da viagem
- **horaInicio:** Hora de início prevista
- **dataHoraFim:** Data e hora previstas para o término
- **pontoSaida/pontoRetorno:** Locais de origem e destino da rota
- **veiculo:** Veículo que realizará a viagem
- **condutorViagem:** Nome do motorista
- **qtdItinerarios:** Quantidade de paradas
- **distanciaEstimadaEmKM:** Distância total da rota
- **tempoEstimadoEmSegundos:** Tempo estimado total da rota
- **polylineViagem:** Representação codificada do trajeto para exibição em mapas

## DespesaVO

Representa uma despesa relacionada a um veículo:

- **condutor:** Nome do motorista
- **placa:** Placa do veículo
- **data/hora:** Data e hora da despesa
- **nome/descricao:** Título e descrição da despesa

- **odometro:** Odômetro do veículo no momento da despesa
- **quantidade/valor:** Quantidade e valor unitário
- **tipo:** Tipo de despesa (abastecimento, manutenção, etc.)
- **tipoCombustível:** Tipo de combustível (se aplicável)
- **fornecedor:** Informações sobre o fornecedor do serviço ou produto

# Tratamento de Erros Comuns

1. **Erro de Autenticação:** Verifique se o CPF/CNPJ e senhas estão corretos.
2. **Erro de Dados Inválidos:** Verifique se todos os campos obrigatórios estão preenchidos corretamente.
3. **Erro de Geocodificação:** Se as coordenadas não puderem ser encontradas, tente fornecer um endereço mais completo ou um CEP válido.
4. **Erro de Roteirização:** Verifique se todos os parâmetros necessários foram informados e se o veículo tem capacidade suficiente para os pedidos.
5. **Erro de Status Inválido:** Certifique-se de que os pedidos estão no status correto antes de tentar alterá-los.
6. **Erro de Entidade Não Encontrada:** Verifique se zonas, lojas, vendedores e motoristas estão previamente cadastrados no sistema.

# Integração com Java

Como a API utiliza SOAP, a integração com Java é facilitada pelo uso da ferramenta wsimport para gerar os stubs de cliente. O documento fornece um script ANT para gerar automaticamente as classes necessárias:

```
<project default="wsimport">
  <target name="wsimport">
    <exec executable="/usr/lib/jvm/jdk1.6.0_45/bin/wsimport">
      <arg line="-keep -s ./src -p concept.gps.wsclient -d ./bin http://52.6.27.50:8181/importadorPedidos?wsdl"/>
    </exec>
  </target>
</project>
```

Após gerar os stubs, você pode usar as classes para acessar a API, como mostrado nos exemplos de código fornecidos na documentação.

# Conclusão

A API de Roteamento de Entregas e Gestão de Frotas da Concept Tecnologia é uma solução completa que permite automatizar o processo logístico, desde a importação de pedidos até a roteirização e controle de despesas. Com uma integração correta, é possível otimizar rotas, reduzir custos operacionais e melhorar a experiência de entrega para os clientes.

Para começar a utilizar a API, é necessário obter as credenciais com a Concept Tecnologia e configurar a integração seguindo as boas práticas mencionadas neste documento. Em caso de dúvidas ou problemas durante a integração, o suporte técnico da Concept está à disposição para ajudar.

# FAQ - Webservice Integração - Importação de Pedidos

Esta documentação detalha as validações implementadas em cada método em *ImportadorPedidos*. Em cada seção, as validações são enumeradas e, em seguida, são apresentadas 15 perguntas e respostas frequentes (FAQ) que abordam dúvidas comuns, sempre fazendo referência ao método e à validação correspondente.

---

## 1. Método:

### `converterEnderecoUsandoCEP`

#### Validações do método

### `converterEnderecoUsandoCEP`

1. **Lista de Pedidos:** Verifica se a lista de pedidos não é nula e contém pelo menos um elemento.
2. **Senha da Central:** Valida a senha informada da central; se inválida, retorna erro.
3. **Identificação do Cliente:** Usa o CPF/CNPJ e senha do cliente para identificar o cliente cadastrado.
4. **CEP do Pedido:** Para cada pedido, verifica se o CEP está informado; caso contrário, retorna erro específico para o pedido.
5. **Conversão do Endereço:** Utilizado para converter o CEP em latitude, longitude e endereço formatado; se não encontrado, retorna erro.

## FAQ – `converterEnderecoUsandoCEP`

**Q1:** O que acontece se a lista de pedidos estiver vazia no método `converterEnderecoUsandoCEP`?

**A1:** O método retorna um objeto de resultado com sucesso falso e a mensagem “Erro: a lista de pedidos deve conter pelo menos um elemento.”, impedindo a conversão sem pedidos.

**Q2:** Como o método *converterEnderecoUsandoCEP* valida a senha da central?

**A2:** Ele utiliza o cadastro da Empresa para validar a senha; se a central não for identificada, retorna erro "Erro: Senha invalida."

**Q3:** O que ocorre se o cliente não for encontrado no método *converterEnderecoUsandoCEP*?

**A3:** O método retorna um erro informando que o cliente com o CPF/CNPJ informado não está cadastrado, indicando o formato esperado para o CNPJ.

**Q4:** Qual a validação aplicada quanto ao CEP em cada pedido?

**A4:** O método verifica se o CEP foi informado; se estiver ausente, retorna erro indicando "Pedido [número] sem o número do CEP."

**Q5:** Como o método trata a conversão do endereço a partir do CEP?

**A5:** Utiliza o cadastro para recuperar um objeto de endereço. Se o endereço não for encontrado, retorna erro especificando que o CEP pesquisado não gerou resultado.

**Q6:** O método *converterEnderecoUsandoCEP* permite pedidos sem CEP?

**A6:** Não; se o CEP estiver vazio para algum pedido, o método interrompe a conversão e retorna a mensagem de erro apropriada.

**Q7:** Há alguma verificação adicional de formatação ou dados no CEP?

**A7:** A validação se concentra em verificar se o campo CEP não está vazio, utilizando-o para recuperar dados no cadastro.

**Q8:** Qual é o retorno do método caso todos os pedidos sejam convertidos com sucesso?

**A8:** Retorna um objeto do tipo *ResultadoConversaoEnderecoPeloCEPVO* com sucesso verdadeiro e uma mensagem informando quantos pedidos foram convertidos.

**Q9:** Em caso de exceção durante a conversão, qual é a resposta do método *converterEnderecoUsandoCEP*?

**A9:** O método captura a exceção e retorna um objeto de resultado com sucesso falso e a mensagem "Erro inesperado na conversão."

**Q10:** O que acontece se a senha da central estiver incorreta?

**A10:** A validação falha e o método retorna "Erro: Senha invalida." sem prosseguir para as demais etapas.

**Q11:** Como é tratado um CEP que não gera resultado no banco de dados?

**A11:** Se o CEP pesquisado não retornar um endereço, o método lança uma mensagem de erro específica para o pedido em questão.

**Q12:** Existe validação de formato para o CEP?

**A12:** A verificação consiste em confirmar que o CEP não está vazio; detalhes de formatação são tratados internamente pelo cadastro responsável pela conversão.

**Q13:** O método *converterEnderecoUsandoCEP* realiza validações por pedido individualmente?

**A13:** Sim, cada pedido é validado individualmente quanto à presença do CEP e a conversão do

endereço.

**Q14:** É possível converter apenas alguns pedidos e ignorar os sem CEP?

**A14:** Não; se qualquer pedido não tiver o CEP informado, o método interrompe e retorna erro para o primeiro pedido sem CEP.

**Q15:** Quais parâmetros são obrigatórios para a execução do método *converterEnderecoUsandoCEP* ?

**A15:** São obrigatórios: uma lista de pedidos com pelo menos um elemento, CPF/CNPJ, senha do cliente e senha da central, além de cada pedido conter o CEP informado.

---

## 2. Método: importarPedidos

### Validações do método importarPedidos

1. **Lista de Pedidos:** Verifica se a lista não é nula e contém ao menos um pedido.
2. **Senha da Central:** Valida a senha da central; se inválida, retorna erro.
3. **Identificação do Cliente:** Valida o CPF/CNPJ e senha do cliente para identificar o cliente.
4. **Ponto de Interesse (POI):** Se o pedido requer cadastro de POI, o método chama a função *validarPoi* para conferir a existência e integridade dos dados do POI.
5. **Loja:** Verifica se a loja está informada e se é válida (pelo código ou nome); caso contrário, retorna erro.
6. **Número do Pedido:** Confere se o número do pedido não é nulo, não vazio e não duplicado conforme regras de data e loja.
7. **Data e Hora do Pedido:** Valida se a data e a hora do pedido estão corretas e se a hora está no formato HH:mm.
8. **Zona e Vendedor:** Valida se a zona e o vendedor estão informados e se correspondem a registros válidos.
9. **Coordenadas Geográficas:** Verifica se latitude e longitude são numéricas e estão dentro dos limites válidos.
10. **Endereço e Geocoding:** Caso não haja coordenadas, o método tenta obter geocoding a partir do endereço; se falhar, retorna erro.
11. **Horários de Entrega:** Valida se a hora inicial e final de entrega estão informadas, corretas e se a final não ocorre antes da inicial.
12. **Tempo de Atendimento:** Verifica se o tempo de atendimento foi informado e é maior que zero.
13. **Número GSM:** Se informado, o número deve estar no formato de celular.
14. **Nota Fiscal, Número de Carregamento e Lote:** Verifica o tamanho máximo permitido (até 45 dígitos) para estes campos.
15. **Data e Hora do Compromisso de Entrega:** Devem estar informados e no formato adequado.

# FAQ – importarPedidos

**Q1:** O que acontece se a lista de pedidos estiver vazia no método *importarPedidos*?

**A1:** O método retorna um `ResultadoImportacaoVO` com sucesso falso e a mensagem “Erro: a lista de pedidos deve conter pelo menos um elemento.”

**Q2:** Como o método *importarPedidos* valida a senha da central?

**A2:** Ele utiliza o cadastro da Empresa para validar a senha; se a central não for encontrada, retorna “Erro: Senha invalida.”

**Q3:** O que ocorre se o cliente não for identificado?

**A3:** O método retorna um erro informando que o cliente com o CPF/CNPJ informado não está cadastrado, especificando o formato esperado do CNPJ.

**Q4:** Como são validados os dados do Ponto de Interesse (POI) no método *importarPedidos*?

**A4:** Se o pedido exigir o cadastro do POI, o método chama a função *validarPoi*, que checa a presença do código, nome, descrição, grupo, raio e, em alguns casos, geocoding ou endereço.

**Q5:** Qual é a validação para a loja no método *importarPedidos*?

**A5:** O método verifica se a loja está informada e se é encontrada (por código ou nome). Caso contrário, retorna erro solicitando o cadastro da loja.

**Q6:** O que acontece se o número do pedido já estiver cadastrado?

**A6:** Dependendo da configuração do cliente (por data ou loja), o método retorna um erro informando que o pedido já foi cadastrado.

**Q7:** Como o método *importarPedidos* valida a data e hora do pedido?

**A7:** Verifica se a data e a hora estão informadas e se a hora está no formato HH:mm; se não, retorna erro indicando “Hora do pedido inválida.”

**Q8:** O que é verificado quanto à zona e ao vendedor do pedido?

**A8:** O método valida se ambos estão informados e se correspondem a registros válidos no sistema; senão, retorna erro com a mensagem apropriada.

**Q9:** Quais validações são realizadas nas coordenadas geográficas?

**A9:** São verificados se latitude e longitude estão preenchidos, se podem ser convertidos para números e se estão dentro dos limites (latitude entre -90 e 90 e longitude entre -180 e 180).

**Q10:** Como o método trata pedidos sem coordenadas mas com endereço?

**A10:** O método tenta obter o geocoding usando o endereço (e, em alguns casos, pelo CEP) para preencher latitude e longitude; se não conseguir, retorna erro.

**Q11:** Como são validados os horários de entrega?

**A11:** Verifica se a hora inicial e final de entrega estão informadas, no formato correto e que a hora final não seja anterior à hora inicial.

**Q12:** Qual é a validação aplicada ao tempo de atendimento?

**A12:** O tempo de atendimento deve ser maior que zero; caso contrário, o método retorna erro indicando a necessidade de informar esse dado.

**Q13:** O número GSM é validado?

**A13:** Sim, se o número GSM estiver informado, ele deve estar no formato de um número de celular válido; senão, ocorre erro.

**Q14:** Como são tratados os campos de Nota Fiscal, Número de Carregamento e Lote?

**A14:** Cada um desses campos é verificado para que não ultrapasse 45 dígitos; se exceder, o método retorna um erro específico.

**Q15:** Quais os requisitos para os dados do compromisso de entrega?

**A15:** É necessário que a data e a hora do compromisso de entrega estejam informadas e que a hora esteja no formato HH:MM, caso contrário, retorna erro.

---

## 3. Método:

# consultarStatusEntregaPedido

## Validações do método

# consultarStatusEntregaPedido

1. **Senha da Central:** Validação inicial da senha da central.
2. **Identificação do Cliente:** Verifica se o cliente existe com base no CPF/CNPJ e senha.
3. **Consulta de Itinerário:** Verifica se há um itinerário associado ao pedido e, se não houver, informa que o pedido nunca foi roteirizado.
4. **Recuperação de Erros:** Caso o itinerário seja encontrado, o método recupera os erros associados ao mesmo.

## FAQ – consultarStatusEntregaPedido

**Q1:** O que acontece se a senha da central for inválida no método *consultarStatusEntregaPedido*?

**A1:** O método retorna um `ResultadoConsultaStatusVO` com sucesso falso e a mensagem “Erro: Senha invalida.”

**Q2:** Como o método valida o cliente?

**A2:** Ele utiliza o cadastro do Cliente para identificar o cliente com CPF/CNPJ e senha; se não

encontrar o cliente, retorna erro indicando que o cliente não está cadastrado.

**Q3:** O que significa se nenhum itinerário for encontrado para o pedido?

**A3:** A mensagem “Pedido nunca foi roteirizado para entrega” é retornada, indicando que o pedido ainda não passou pelo processo de roteirização.

**Q4:** Como o método *consultarStatusEntregaPedido* lida com erros associados ao itinerário?

**A4:** Após recuperar o itinerário, o método chama uma rotina para recuperar erros e os inclui na resposta.

**Q5:** É necessário informar todos os parâmetros para consultar o status de entrega?

**A5:** Sim, é obrigatório informar o pedido, CPF/CNPJ, senha do cliente e senha da central.

**Q6:** Qual é o tipo de retorno quando o status é consultado com sucesso?

**A6:** Retorna um objeto *ResultadoConsultaStatusVO* com sucesso verdadeiro, a mensagem “Pedido encontrado.” e o itinerário de viagem.

**Q7:** O que ocorre se ocorrer uma exceção durante a consulta do status?

**A7:** O método captura a exceção, registra o erro e retorna um *ResultadoConsultaStatusVO* com sucesso falso e uma mensagem genérica de erro.

**Q8:** Posso consultar o status de qualquer pedido?

**A8:** Sim, desde que o pedido possua um registro de itinerário ou erros de roteirização no sistema.

**Q9:** Como o método trata a conexão com o banco?

**A9:** Ele obtém uma nova conexão por meio do *FabricaConexao* para cada operação de validação e consulta.

**Q10:** Qual a importância do itinerário na consulta do status de entrega?

**A10:** O itinerário contém o histórico de status e erros que permitem identificar o estágio atual do pedido na entrega.

**Q11:** O que é necessário para que a consulta retorne “Pedido encontrado.”?

**A11:** É necessário que exista um itinerário previamente cadastrado para o pedido e que a conexão e validação dos dados tenham sido bem-sucedidas.

**Q12:** Como é validada a conexão com o banco de dados no método?

**A12:** O método utiliza o *FabricaConexao* para obter a conexão, assegurando que cada chamada aos cadastros possua conexão ativa.

**Q13:** Existe alguma validação específica do objeto “pedido”?

**A13:** A validação se concentra na existência do itinerário relacionado ao pedido; os dados do pedido já devem estar previamente validados na etapa de importação.

**Q14:** O método pode retornar informações parciais em caso de erro?

**A14:** Não; se ocorrer qualquer falha na validação ou consulta, o método retorna um objeto de resultado com sucesso falso e mensagem de erro.

**Q15:** Quais parâmetros devo conferir se a consulta não retornar o status esperado?

**A15:** Verifique se o CPF/CNPJ, senha do cliente, senha da central e os dados do pedido estão corretos, pois qualquer inconsistência impedirá a consulta do itinerário.

---

## 4. Método:

# listarItinerariosCarregamento

## Validações do método

# listarItinerariosCarregamento

1. **Senha da Central:** Verifica se a senha da central é válida.
2. **Identificação do Cliente:** Confirma a existência do cliente com os dados fornecidos.
3. **Recuperação da Lista:** Utiliza o cadastro de Itinerário para recuperar a lista de itinerários com base no número de carregamento.

## FAQ – listarItinerariosCarregamento

**Q1:** Quais parâmetros são obrigatórios no método *listarItinerariosCarregamento*?

**A1:** É necessário informar o número do carregamento, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** O que acontece se a senha da central for inválida?

**A2:** O método retorna um ResultadoConsultaListaltinerariosVO com sucesso falso e a mensagem “Erro: Senha invalida.”

**Q3:** Como o método identifica o cliente?

**A3:** Utiliza o cadastro do Cliente para validar o CPF/CNPJ e senha; se o cliente não for encontrado, retorna o erro correspondente.

**Q4:** Qual é o critério de busca para os itinerários?

**A4:** A busca é feita pelo número do carregamento associado aos pedidos.

**Q5:** O que ocorre se nenhum itinerário for encontrado para o carregamento?

**A5:** O método retorna uma lista vazia com uma mensagem de sucesso indicando que a lista foi recuperada, mas o usuário pode interpretar que não há itinerários cadastrados.

**Q6:** Posso utilizar o método para buscar itinerários de vários carregamentos?

**A6:** Este método foi projetado para filtrar por um único número de carregamento.

**Q7:** Como é tratada a conexão com o banco?

**A7:** O método obtém a conexão por meio do FabricaConexao e a repassa para os cadastros utilizados.

**Q8:** Existe tratamento de exceções no método?

**A8:** Sim, se ocorrer alguma exceção, o método captura o erro e retorna uma mensagem genérica de falha na consulta dos itinerários.

**Q9:** O que o método retorna em caso de sucesso?

**A9:** Um objeto ResultadoConsultaListaltinerariosVO com sucesso verdadeiro, uma mensagem de sucesso e a lista de itinerários encontrados.

**Q10:** É possível que a lista de itinerários contenha erros?

**A10:** A função se preocupa em recuperar a lista; a eventual existência de erros nos itinerários é tratada em métodos complementares de recuperação.

**Q11:** Quais tipos de itinerários são listados?

**A11:** São listados itinerários associados ao número de carregamento informado e pertencentes ao cliente autenticado.

**Q12:** O método realiza alguma validação sobre o formato do número de carregamento?

**A12:** A validação principal é a existência do número; detalhes adicionais devem estar conformes ao cadastro do cliente.

**Q13:** Como é informado o usuário se não houver itinerários?

**A13:** A mensagem de retorno será "Lista de Itinerarios recuperada com sucesso.", porém a lista estará vazia.

**Q14:** Existe limite para o número de itinerários retornados?

**A14:** Não há um limite explícito; todos os itinerários correspondentes são retornados.

**Q15:** Quais cadastros são utilizados na execução deste método?

**A15:** São utilizados o cadastro da Empresa para validação, o cadastro do Cliente para identificação e o cadastro de Itinerário para a recuperação dos itinerários.

---

## 5. Método: listarItinerariosLote

### Validações do método listarItinerariosLote

1. **Senha da Central:** Verifica se a senha da central está correta.
2. **Identificação do Cliente:** Confirma se o cliente existe.
3. **Busca pelo Lote:** Recupera a lista de itinerários com base no número do lote.
4. **Validação de Lista Vazia:** Se a lista estiver vazia, atualiza a mensagem informando “Nenhuma roteirização ou itinerário encontrado.”

## FAQ – listarItinerariosLote

**Q1:** Quais dados são necessários para executar o método *listarItinerariosLote*?

**A1:** É necessário informar o número do lote, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Como o método trata a senha da central?

**A2:** Valida a senha utilizando o cadastro da Empresa; se inválida, retorna “Erro: Senha invalida.”

**Q3:** O que acontece se o cliente não for identificado?

**A3:** O método retorna um erro informando que o cliente não está cadastrado.

**Q4:** Como é realizada a busca dos itinerários por lote?

**A4:** Através do cadastro de Itinerário, filtrando pelo número do lote e associado ao cliente autenticado.

**Q5:** Qual é a mensagem retornada se nenhum itinerário for encontrado?

**A5:** A mensagem será “Nenhuma roteirização ou itinerário encontrado.”

**Q6:** O método retorna um objeto de sucesso mesmo com a lista vazia?

**A6:** Sim, o objeto é retornado com sucesso verdadeiro, mas com uma mensagem específica e a lista vazia.

**Q7:** Quais conexões são utilizadas na execução do método?

**A7:** São utilizadas conexões obtidas via FabricaConexao para todos os cadastros envolvidos.

**Q8:** Existe validação sobre o formato do número do lote?

**A8:** A validação se limita a verificar se o número foi informado; o formato deve estar conforme o cadastro do cliente.

**Q9:** Posso buscar itinerários de lotes parciais?

**A9:** O método busca por correspondência exata do número do lote informado.

**Q10:** Qual cadastro é responsável por recuperar a lista de itinerários?

**A10:** O cadastro de Itinerário é utilizado para essa recuperação.

**Q11:** O método realiza tratamento de exceções?

**A11:** Sim, qualquer exceção é capturada e o método retorna uma mensagem de erro genérica.

**Q12:** É possível filtrar itinerários por mais de um lote neste método?

**A12:** Este método aceita apenas um número de lote por chamada.

**Q13:** Como é informado o usuário se a senha estiver incorreta?

**A13:** Retorna “Erro: Senha invalida.” sem prosseguir com a consulta.

**Q14:** O que fazer se a lista de itinerários estiver vazia?

**A14:** O usuário deverá verificar se o número do lote está correto ou se há itinerários cadastrados para aquele lote.

**Q15:** Quais são os principais cadastros utilizados neste método?

**A15:** São utilizados os cadastros da Empresa, Cliente e ItinerarioViagem.

---

## 6. Método: listarItinerariosRoterizado

### Validações do método

#### listarItinerariosRoterizado

1. **Senha da Central:** Valida a senha informada.
2. **Identificação do Cliente:** Confirma se o cliente está cadastrado.
3. **Verificação da Viagem:** Verifica se existe uma viagem com o ID informado para o cliente.
4. **Recuperação dos Itinerários:** Utiliza o cadastro de Itinerário para recuperar a lista com base na viagem.
5. **Recuperação de Erros:** Chama a rotina para atualizar a lista de ocorrências (erros) para cada itinerário.

## FAQ – listarItinerariosRoterizado

**Q1:** Quais parâmetros são obrigatórios no método *listarItinerariosRoterizado*?

**A1:** É necessário informar o ID da viagem, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** O que acontece se a senha da central for inválida?

**A2:** O método retorna um ResultadoConsultaListaltinerariosVO com sucesso falso e a mensagem “Erro: Senha invalida.”

**Q3:** Como o método valida a existência da viagem?

**A3:** Ele utiliza o cadastro de Viagem para recuperar a viagem com o ID informado; se não existir, retorna erro “Erro: Nao existe viagem com o identificador informado.”

**Q4:** O que é retornado se a viagem for encontrada?

**A4:** Retorna uma lista de itinerários associados à viagem e, se houver ocorrências, estas são recuperadas e associadas a cada itinerário.

**Q5:** Como é tratada a conexão com o banco?

**A5:** São utilizadas conexões obtidas via FabricaConexao para cada operação de validação e recuperação.

**Q6:** O que ocorre se o cliente não for identificado?

**A6:** O método retorna erro informando que o cliente não está cadastrado.

**Q7:** Qual cadastro é responsável por recuperar os itinerários?

**A7:** O cadastro é o responsável pela recuperação dos itinerários.

**Q8:** Como são atualizados os erros associados aos itinerários?

**A8:** Para cada itinerário, o método chama a função de recuperação de erros e atualiza a lista de ocorrências.

**Q9:** Existe algum limite para o número de itinerários retornados?

**A9:** Não há limite explícito; todos os itinerários associados à viagem são retornados.

**Q10:** O que acontece se ocorrer uma exceção durante a recuperação?

**A10:** A exceção é capturada, o erro é logado e o método retorna um objeto de resultado com sucesso falso e mensagem genérica de erro.

**Q11:** O método pode ser usado para consultar itinerários de viagens finalizadas?

**A11:** Sim, contanto que o ID da viagem esteja corretamente informado e associada ao cliente.

**Q12:** Há alguma validação específica do formato do ID da viagem?

**A12:** A validação principal é a existência da viagem para o cliente; o formato do ID deve estar conforme o cadastro no sistema.

**Q13:** Como saber se a consulta foi realizada com sucesso?

**A13:** O objeto retornado terá o campo "sucesso" verdadeiro e uma mensagem "Lista de Itinerarios recuperada com sucesso."

**Q14:** Quais informações adicionais podem ser encontradas nos itinerários?

**A14:** Além dos dados do pedido, os itinerários podem conter a lista de ocorrências (erros) associadas.

**Q15:** É necessário tratar manualmente os erros de itinerários após a consulta?

**A15:** Não; o método já recupera e associa os erros a cada itinerário automaticamente.

---

# 7. Método: listarPolRoterizado

## Validações do método listarPolRoterizado

1. **Senha da Central:** Valida a senha informada.
2. **Identificação do Cliente:** Confirma se o cliente está cadastrado.
3. **Verificação da Viagem:** Confirma a existência da viagem com o ID informado.
4. **Recuperação dos Itinerários:** Obtém os itinerários da viagem.
5. **Extração de POIs:** A partir dos itinerários, extrai os pontos de interesse (POI) sem repetições.

## FAQ – listarPolRoterizado

**Q1:** Quais parâmetros são necessários para o método *listarPolRoterizado*?

**A1:** É preciso informar o ID da viagem, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** O que acontece se a senha da central for inválida?

**A2:** O método retorna um ResultadoConsultaListaPolRoteirizadosVO com sucesso falso e a mensagem “Erro: Senha invalida.”

**Q3:** Como o método valida a existência da viagem?

**A3:** Utiliza o cadastro de Viagem para recuperar a viagem com o ID informado; se não for encontrada, retorna erro.

**Q4:** Como são extraídos os POIs dos itinerários?

**A4:** O método percorre a lista de itinerários e, para cada novo POI (não repetido), adiciona-o à lista final.

**Q5:** O que é retornado se nenhum POI for encontrado?

**A5:** Se a lista de itinerários estiver vazia ou não houver POIs distintos, a lista retornada ficará vazia.

**Q6:** Qual é a importância do campo “idPontoInteresse” nesta consulta?

**A6:** Ele é utilizado para identificar e evitar a duplicação dos pontos de interesse na lista final.

**Q7:** O que ocorre se o cliente não for identificado?

**A7:** O método retorna erro informando que o cliente não está cadastrado.

**Q8:** Existe alguma validação adicional quanto aos dados dos itinerários?

**A8:** A validação foca na existência dos itinerários e na extração única dos POIs.

**Q9:** Como é tratada a conexão com o banco neste método?

**A9:** A conexão é obtida via `FabricaConexao` e utilizada nos cadastros de Empresa, Cliente e Viagem.

**Q10:** Quais mensagens de erro podem ser retornadas neste método?

**A10:** Podem ocorrer mensagens “Erro: Senha invalida.”, “Erro: cliente com CPF/CNPJ ... nao cadastrado.” ou “Erro: Nao existe viagem com o identificador informado.”

**Q11:** Posso usar este método para verificar se um POI já foi roteirizado?

**A11:** Sim, ele retorna os pontos de interesse que já foram associados a itinerários de uma viagem.

**Q12:** O método lida com duplicatas de POI?

**A12:** Sim, a lógica compara o ID dos POIs para garantir que cada ponto seja listado apenas uma vez.

**Q13:** Qual cadastro é utilizado para recuperar os itinerários?

**A13:** O cadastro do `ItinerarioViagem` é utilizado para obter a lista de itinerários.

**Q14:** O método retorna informações adicionais sobre cada POI?

**A14:** Retorna os dados básicos do POI, como o ID e o código, conforme extraídos dos itinerários.

**Q15:** Como saber se a operação foi realizada com sucesso?

**A15:** O objeto retornado terá o campo “sucesso” verdadeiro e a mensagem “Lista de Pontos de Interesse recuperada com sucesso.”

---

## 8. Método: `listarPedidosPorStatus`

### Validações do método

### `listarPedidosPorStatus`

1. **Senha da Central:** Verifica se a senha informada é válida.
2. **Identificação do Cliente:** Confirma se o cliente está cadastrado.
3. **Filtragem por Status:** Utiliza o cadastro de Pedidos para recuperar a lista de pedidos com o status informado.

### FAQ – `listarPedidosPorStatus`

**Q1:** Quais parâmetros são obrigatórios no método `listarPedidosPorStatus`?

**A1:** É necessário informar o status do pedido, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Como é validada a senha da central?

**A2:** Através do cadastro da Empresa; se inválida, retorna “Erro: Senha invalida.”

**Q3:** O que acontece se o cliente não for identificado?

**A3:** O método retorna erro informando que o cliente com o CPF/CNPJ informado não está cadastrado.

**Q4:** Qual é o critério de filtragem para os pedidos?

**A4:** A filtragem é feita com base no status do pedido (por exemplo, VENDA\_FINALIZADA).

**Q5:** O método pode retornar uma lista vazia?

**A5:** Sim, se nenhum pedido corresponder ao status informado, a lista retornada ficará vazia.

**Q6:** Qual cadastro é utilizado para recuperar os pedidos?

**A6:** O método utiliza o cadastro de PedidosRotas para listar os pedidos conforme o status.

**Q7:** Existe tratamento para exceções durante a consulta?

**A7:** Sim, exceções são capturadas e o método retorna um resultado com sucesso falso e uma mensagem de erro.

**Q8:** Como é informado o usuário se a operação for bem-sucedida?

**A8:** O método retorna um objeto com sucesso verdadeiro e a mensagem “Lista de Pedidos recuperada com sucesso.”

**Q9:** O status informado é case sensitive?

**A9:** O status deve ser informado conforme definido na enumeração StatusPedido do sistema.

**Q10:** É possível filtrar pedidos por múltiplos status com este método?

**A10:** Este método aceita apenas um status por chamada.

**Q11:** Como o método lida com conexões ao banco?

**A11:** Ele obtém conexões via FabricaConexao para os cadastros de Empresa, Cliente e Pedidos.

**Q12:** Posso usar este método para monitorar o andamento dos pedidos?

**A12:** Sim, ele é ideal para recuperar pedidos em um status específico e monitorar sua evolução.

**Q13:** O que devo fazer se nenhum pedido for retornado?

**A13:** Verifique se o status informado está correto e se há pedidos cadastrados com aquele status.

**Q14:** Qual a utilidade deste método para o cliente?

**A14:** Ele permite ao cliente acompanhar a situação dos seus pedidos, filtrando por status como “VENDA\_FINALIZADA”, entre outros.

**Q15:** Quais são os possíveis status que podem ser consultados?

**A15:** Os status são definidos na enumeração StatusPedido e podem incluir VENDA\_FINALIZADA, LIBERADO\_PARA\_SEPARACAO, NAO\_ENTREGUE, etc.

---

## 9. Método:

# alterarStatusVendaFinalizadoParaLiberadoSeparacao

## Validações do método

# alterarStatusVendaFinalizadoParaLiberadoSeparacao

1. **Validação de Acesso:** Verifica se a senha da central e os dados do cliente estão corretos.
2. **Status Atual:** Confirma que o status do pedido está como VENDA\_FINALIZADA.
3. **Chamada à Fachada:** Encaminha a alteração de status para o GerenciadorPedidosFachada, que realiza as validações internas.

## FAQ –

# alterarStatusVendaFinalizadoParaLiberadoSeparacao

**Q1:** Quais parâmetros são obrigatórios no método *alterarStatusVendaFinalizadoParaLiberadoSeparacao*?

**A1:** É preciso informar a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** O que acontece se a senha da central for inválida?

**A2:** O método retorna um *ResultadoAlteracaoStatusPedidosVO* com sucesso falso e a mensagem “Erro: Senha invalida.”

**Q3:** Como é verificado o status atual do pedido?

**A3:** A validação é realizada internamente pelo *GerenciadorPedidosFachada*, que confirma que o status atual é VENDA\_FINALIZADA.

**Q4:** Qual o novo status aplicado pelo método?

**A4:** O método altera o status de VENDA\_FINALIZADA para LIBERADO\_PARA\_SEPARACAO.

**Q5:** O que ocorre se o pedido não estiver com o status VENDA\_FINALIZADA?

**A5:** O GerenciadorPedidosFachada deverá identificar a inconsistência e retornar um erro apropriado.

**Q6:** O método altera o status de todos os pedidos da lista?

**A6:** Sim, todos os pedidos contidos na lista que estiverem com o status correto serão alterados.

**Q7:** Quais validações de acesso são realizadas?

**A7:** São validados a senha da central e a identificação do cliente, conforme nos métodos anteriores.

**Q8:** Se ocorrer um erro na alteração, qual é a mensagem retornada?

**A8:** Em caso de exceção, o método retorna "Erro geral : alterarStatusVendaFinalizadoParaLiberadoSeparacao."

**Q9:** O método realiza alterações parciais ou tudo ou nada?

**A9:** A alteração é realizada para todos os pedidos que cumpram a validação; se houver erro em algum, geralmente a transação é abortada.

**Q10:** Como o GerenciadorPedidosFachada é utilizado neste método?

**A10:** Ele é instanciado e chamado para executar a alteração do status, centralizando a lógica de negócios.

**Q11:** Existe alguma validação quanto à lista de pedidos?

**A11:** Sim, a lista não pode ser nula e deve conter pelo menos um pedido.

**Q12:** Posso usar este método para alterar pedidos com status diferente?

**A12:** Não, este método é específico para alterar de VENDA\_FINALIZADA para LIBERADO\_PARA\_SEPARACAO.

**Q13:** Quais cadastros são invocados durante a alteração?

**A13:** A alteração é realizada pela fachada (GerenciadorPedidosFachada), que internamente utiliza os cadastros de pedidos.

**Q14:** Qual é o retorno esperado em caso de sucesso?

**A14:** Um ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e mensagem confirmando a atualização do status.

**Q15:** É possível acompanhar quais pedidos foram alterados?

**A15:** O retorno inclui informações sobre a quantidade de pedidos atualizados, permitindo a verificação da operação.

---

# 10. Método:

## alterarStatusLiberadoSeparacaParaEmSeparacao

### Validações do método

## alterarStatusLiberadoSeparacaParaEmSeparacao

1. **Acesso e Cliente:** Valida senha da central e identificação do cliente.
2. **Status Atual:** Confirma que o status atual é LIBERADO\_PARA\_SEPARACAO.
3. **Parametro Compromisso:** Verifica se o parâmetro compromissoSeparacao foi informado corretamente.
4. **Chamada à Fachada:** Encaminha a alteração para o GerenciadorPedidosFachada, que realiza a mudança para EM\_SEPARACAO.

### FAQ –

## alterarStatusLiberadoSeparacaParaEmSeparacao

**Q1:** Quais parâmetros são obrigatórios no método *alterarStatusLiberadoSeparacaParaEmSeparacao*?

**A1:** São necessários a lista de pedidos, o compromisso de separação, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** O que acontece se o parâmetro compromissoSeparacao estiver ausente?

**A2:** O método pode retornar erro ou falhar na alteração, pois esse parâmetro é necessário para a alteração de status.

**Q3:** Qual o status atual exigido para a alteração neste método?

**A3:** Os pedidos devem estar com o status LIBERADO\_PARA\_SEPARACAO.

**Q4:** Qual o novo status que será aplicado?

**A4:** O status dos pedidos será alterado para EM\_SEPARACAO.

**Q5:** Como é verificada a senha da central?

**A5:** Por meio do cadastro da Empresa; se inválida, o método retorna “Erro: Senha invalida.”

**Q6:** O que acontece se o cliente não for identificado?

**A6:** O método retorna erro informando que o cliente não está cadastrado.

**Q7:** A alteração é realizada individualmente para cada pedido?

**A7:** Sim, a alteração é aplicada a cada pedido na lista que passar nas validações.

**Q8:** Como o GerenciadorPedidosFachada atua neste método?

**A8:** Ele recebe os parâmetros e efetua a alteração de status, realizando as validações internas necessárias.

**Q9:** O que acontece se algum pedido não estiver no status esperado?

**A9:** O GerenciadorPedidosFachada deverá retornar um erro para o(s) pedido(s) que não atenderem ao status atual exigido.

**Q10:** Qual é o retorno do método em caso de sucesso?

**A10:** Um objeto ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e uma mensagem informando a alteração para EM\_SEPARACAO.

**Q11:** Como é tratada uma exceção durante o processo?

**A11:** Se ocorrer uma exceção, o método captura o erro e retorna “Erro geral : alterarStatusLiberadoSeparacaParaEmSeparacao.”

**Q12:** O método permite alterar apenas pedidos com status LIBERADO\_PARA\_SEPARACAO?

**A12:** Sim, a alteração só ocorre se o status atual for o esperado.

**Q13:** Quais validações de acesso são realizadas?

**A13:** São verificadas a senha da central e a existência do cliente com base nos dados fornecidos.

**Q14:** Posso usar este método para alterar pedidos sem compromisso de separação?

**A14:** Não, o parâmetro compromissoSeparacao é obrigatório para a transição para EM\_SEPARACAO.

**Q15:** Onde posso acompanhar os pedidos atualizados?

**A15:** O retorno do método informa quantos pedidos foram atualizados, permitindo a verificação da operação.

---

## 11. Método:

# alterarStatusEmSeparacaoParaSep

# arados

## Validações do método

### alterarStatusEmSeparacaoParaSeparados

1. **Acesso e Cliente:** Valida a senha da central e identifica o cliente.
2. **Status Atual:** Verifica se o status atual é EM\_SEPARACAO.
3. **Chamada à Fachada:** Encaminha para o GerenciadorPedidosFachada para alterar o status para SEPARADOS.

## FAQ –

### alterarStatusEmSeparacaoParaSeparados

**Q1:** Quais parâmetros são obrigatórios no método *alterarStatusEmSeparacaoParaSeparados*?

**A1:** É necessário informar a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Qual o status esperado dos pedidos para esta alteração?

**A2:** Os pedidos devem estar no status EM\_SEPARACAO.

**Q3:** Qual o novo status aplicado por este método?

**A3:** O status será alterado para SEPARADOS.

**Q4:** Como é validada a senha da central?

**A4:** Através do cadastro da Empresa; se a senha for inválida, retorna “Erro: Senha invalida.”

**Q5:** O que ocorre se o cliente não for encontrado?

**A5:** O método retorna erro informando que o cliente não está cadastrado.

**Q6:** Como a alteração de status é processada?

**A6:** O GerenciadorPedidosFachada realiza a alteração após confirmar que os pedidos estão com o status EM\_SEPARACAO.

**Q7:** É possível que apenas alguns pedidos sejam alterados?

**A7:** A alteração é aplicada a todos os pedidos que cumpram as validações; caso algum não esteja no status correto, a transação poderá ser abortada.

**Q8:** Qual é o retorno em caso de sucesso?

**A8:** Retorna um ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e uma mensagem confirmando a alteração para SEPARADOS.

**Q9:** O método trata exceções?

**A9:** Sim, exceções são capturadas e o método retorna “Erro geral : alterarStatusEmSeparacaoParaSeparados.”

**Q10:** É possível alterar o status se o pedido não estiver em EM\_SEPARACAO?

**A10:** Não, o método só altera pedidos que estejam no status EM\_SEPARACAO.

**Q11:** Quais cadastros são invocados na alteração?

**A11:** A operação é realizada internamente pelo GerenciadorPedidosFachada, que utiliza os cadastros de pedidos.

**Q12:** Há alguma validação sobre a lista de pedidos?

**A12:** Sim, a lista deve conter pelo menos um pedido, caso contrário, a operação não é realizada.

**Q13:** Como é informada a quantidade de pedidos atualizados?

**A13:** O retorno do método inclui a quantidade de pedidos que tiveram o status alterado.

**Q14:** O que devo fazer se a alteração não ocorrer?

**A14:** Verifique se os pedidos estão no status correto e se os parâmetros de acesso (senha, CPF/CNPJ) estão corretos.

**Q15:** Posso usar este método para monitorar a separação dos pedidos?

**A15:** Sim, ele é utilizado para marcar os pedidos que passaram da etapa de separação para o status SEPARADOS.

---

## 12. Método:

# alterarStatusSeparadosParaLiberadosRoteirizacao

## Validações do método

# alterarStatusSeparadosParaLiberadosRoteirizacao

1. **Acesso e Cliente:** Verifica senha da central e identificação do cliente.
2. **Status Atual:** Confirma que os pedidos estão com status SEPARADOS.

3. **Chamada à Fachada:** Encaminha para o GerenciadorPedidosFachada que altera o status para LIBERADOS\_PARA\_ROTUIZACAO.

## FAQ –

# alterarStatusSeparadosParaLiberadosRoteirizacao

**Q1:** Quais parâmetros são obrigatórios no método *alterarStatusSeparadosParaLiberadosRoteirizacao*?

**A1:** É necessário informar a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Qual o status atual exigido para esta alteração?

**A2:** Os pedidos devem estar no status SEPARADOS.

**Q3:** Qual o novo status aplicado?

**A3:** O status será alterado para LIBERADOS\_PARA\_ROTUIZACAO.

**Q4:** Como é validada a senha da central?

**A4:** A senha é verificada via cadastro da Empresa; se estiver incorreta, retorna “Erro: Senha invalida.”

**Q5:** O que ocorre se o cliente não for identificado?

**A5:** Retorna erro informando que o cliente não está cadastrado.

**Q6:** Como é executada a alteração de status?

**A6:** Por meio do GerenciadorPedidosFachada, que valida o status atual e efetua a mudança.

**Q7:** O método processa todos os pedidos da lista?

**A7:** Sim, desde que todos os pedidos estejam no status SEPARADOS.

**Q8:** Qual é o retorno em caso de sucesso?

**A8:** Um objeto ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e mensagem informando a alteração.

**Q9:** E se algum pedido não estiver no status SEPARADOS?

**A9:** A alteração não será efetuada para aquele pedido e o método poderá retornar erro para a transação.

**Q10:** Como são tratadas exceções?

**A10:** Exceções são capturadas e o método retorna “Erro geral : *alterarStatusSeparadosParaLiberadosRoteirizacao*.”

**Q11:** O método altera o status individualmente ou em lote?

**A11:** A alteração é aplicada em lote a todos os pedidos que cumpram as condições.

**Q12:** Posso reexecutar o método caso a alteração falhe?

**A12:** Sim, desde que os pedidos estejam novamente com o status SEPARADOS e os parâmetros de acesso estejam corretos.

**Q13:** Quais cadastros são utilizados durante a operação?

**A13:** A operação é realizada pela fachada que utiliza internamente os cadastros relacionados aos pedidos.

**Q14:** Como posso confirmar a quantidade de pedidos alterados?

**A14:** O retorno do método informa quantos pedidos tiveram seu status atualizado.

**Q15:** Este método pode ser usado para integrar com outros sistemas de roteirização?

**A15:** Sim, ao liberar os pedidos para roteirização, o sistema permite que outras integrações processem a rota de entrega.

---

## 13. Método:

# alterarStatusNaoEntreguesParaLiberadoSeparacao

## Validações do método

# alterarStatusNaoEntreguesParaLiberadoSeparacao

1. **Acesso e Cliente:** Valida a senha da central e identifica o cliente.
2. **Status Atual:** Verifica se os pedidos estão com status NAO\_ENTREGUE.
3. **Chamada à Fachada:** Encaminha para o GerenciadorPedidosFachada, que altera o status para LIBERADO\_PARA\_SEPARACAO.

## FAQ –

# alterarStatusNaoEntreguesParaLiberadoSe

# paracao

**Q1:** Quais parâmetros são necessários no método *alterarStatusNaoEntreguesParaLiberadoSeparacao*?

**A1:** É obrigatório informar a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Qual o status atual exigido para a alteração?

**A2:** Os pedidos devem estar com o status NAO\_ENTREGUE.

**Q3:** Qual o novo status que será aplicado?

**A3:** O novo status será LIBERADO\_PARA\_SEPARACAO.

**Q4:** Como é validada a senha da central?

**A4:** Utilizando o cadastro da Empresa; se a senha estiver incorreta, retorna “Erro: Senha invalida.”

**Q5:** O que ocorre se o cliente não for identificado?

**A5:** O método retorna erro informando que o cliente não está cadastrado.

**Q6:** Qual é o papel do GerenciadorPedidosFachada neste método?

**A6:** Ele centraliza a lógica de alteração, validando o status atual e efetuando a mudança.

**Q7:** Se algum pedido não estiver com status NAO\_ENTREGUE, o que acontece?

**A7:** A alteração só será efetuada para os pedidos que cumpram a condição; os demais gerarão erro.

**Q8:** Qual o retorno em caso de sucesso?

**A8:** Um objeto ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e mensagem confirmando a atualização.

**Q9:** Como são tratadas exceções?

**A9:** Em caso de exceção, o método retorna “Erro geral : alterarStatusNaoEntreguesParaLiberadoSeparacao.”

**Q10:** A alteração é feita individualmente ou em lote?

**A10:** A alteração é aplicada em lote para todos os pedidos válidos na lista.

**Q11:** É possível reexecutar a operação se ocorrer um erro?

**A11:** Sim, desde que os pedidos estejam corretamente com o status NAO\_ENTREGUE e os dados de acesso estejam corretos.

**Q12:** Quais cadastros são utilizados na alteração?

**A12:** A alteração é realizada pela fachada, que utiliza os cadastros de pedidos conforme necessário.

**Q13:** O método permite acompanhar quantos pedidos foram atualizados?

**A13:** Sim, o retorno inclui a quantidade de pedidos que tiveram seu status alterado.

**Q14:** Existe alguma verificação especial para a lista de pedidos?

**A14:** Sim, a lista não pode ser nula e deve conter ao menos um pedido.

**Q15:** Como posso confirmar se a alteração foi realizada corretamente?

**A15:** Verifique a mensagem de retorno e a quantidade de pedidos atualizados informados no objeto de resultado.

---

## 14. Método:

# alterarStatusNaoEntreguesParaLiberadoRoteirizacao

## Validações do método

# alterarStatusNaoEntreguesParaLiberadoRoteirizacao

1. **Acesso e Cliente:** Valida senha da central e identificação do cliente.
2. **Status Atual:** Confirma que os pedidos estão com status NAO\_ENTREGUE.
3. **Chamada à Fachada:** Encaminha para o GerenciadorPedidosFachada que altera o status para LIBERADOS\_PARA\_ROTUIRIZACAO.

## FAQ –

# alterarStatusNaoEntreguesParaLiberadoRoteirizacao

**Q1:** Quais parâmetros são necessários no método *alterarStatusNaoEntreguesParaLiberadoRoteirizacao*?

**A1:** É necessário fornecer a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Qual o status atual exigido para a alteração?

**A2:** Os pedidos devem estar com o status NAO\_ENTREGUE.

**Q3:** Qual o novo status aplicado por este método?

**A3:** O status será alterado para LIBERADOS\_PARA\_ROTUIRIZACAO.

**Q4:** Como é validada a senha da central?

**A4:** A senha é validada utilizando o cadastro da Empresa; se inválida, retorna "Erro: Senha inválida."

**Q5:** O que ocorre se o cliente não for identificado?

**A5:** O método retorna erro indicando que o cliente não está cadastrado.

**Q6:** Qual é o papel do GerenciadorPedidosFachada neste processo?

**A6:** Ele é responsável por validar o status atual e realizar a alteração para LIBERADOS\_PARA\_ROTUIRIZACAO.

**Q7:** Se algum pedido não estiver com status NAO\_ENTREGUE, o que acontece?

**A7:** Apenas os pedidos com status NAO\_ENTREGUE serão alterados; os demais serão ignorados ou geram erro.

**Q8:** Qual é o retorno do método em caso de sucesso?

**A8:** Um objeto ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e mensagem confirmando a atualização.

**Q9:** Como o método lida com exceções?

**A9:** Se ocorrer uma exceção, o método captura e retorna "Erro geral : alterarStatusNaoEntreguesParaLiberadoRoteirizacao."

**Q10:** A operação é aplicada a cada pedido individualmente?

**A10:** A alteração é feita em lote para todos os pedidos que atendam às condições.

**Q11:** Quais validações de acesso são realizadas?

**A11:** São validadas a senha da central e a identificação do cliente.

**Q12:** O método permite reexecutar a alteração se houver erro?

**A12:** Sim, desde que os pedidos estejam com o status correto e os parâmetros estejam corretos.

**Q13:** Quais cadastros são utilizados nesta operação?

**A13:** A operação é realizada pela fachada, que utiliza internamente os cadastros de pedidos.

**Q14:** É possível acompanhar quantos pedidos foram atualizados?

**A14:** Sim, o objeto de resultado informa a quantidade de pedidos alterados.

**Q15:** Qual a principal diferença entre este método e o anterior de não entregues para separação?

**A15:** A diferença está no status de destino: este método altera para LIBERADOS\_PARA\_ROTUIRIZACAO, enquanto o outro altera para LIBERADO\_PARA\_SEPARACAO.

---

# 15. Método:

## cancelar\_LiberadoRoteirizacao

### Validações do método

## cancelar\_LiberadoRoteirizacao

1. **Acesso e Cliente:** Valida a senha da central e identifica o cliente.
2. **Status Atual:** Verifica que os pedidos estão com status LIBERADOS\_PARA\_ROTEIRIZACAO.
3. **Chamada à Fachada:** Chama o GerenciadorPedidosFachada para reverter o status para SEPARADOS.

## FAQ – cancelar\_LiberadoRoteirizacao

**Q1:** Quais parâmetros são obrigatórios no método *cancelar\_LiberadoRoteirizacao*?

**A1:** É necessário fornecer a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Qual o status atual esperado para cancelar a roteirização?

**A2:** Os pedidos devem estar com status LIBERADOS\_PARA\_ROTEIRIZACAO.

**Q3:** Qual o novo status definido pelo cancelamento?

**A3:** O status é revertido para SEPARADOS.

**Q4:** Como é validada a senha da central?

**A4:** Utilizando o cadastro da Empresa; se a senha estiver errada, retorna “Erro: Senha invalida.”

**Q5:** O que acontece se o cliente não for identificado?

**A5:** O método retorna erro indicando que o cliente não está cadastrado.

**Q6:** Qual o papel do GerenciadorPedidosFachada nesta operação?

**A6:** Ele é chamado para efetuar a alteração de status de LIBERADOS\_PARA\_ROTEIRIZACAO para SEPARADOS.

**Q7:** Se algum pedido não estiver no status LIBERADOS\_PARA\_ROTEIRIZACAO, o que ocorre?

**A7:** Apenas os pedidos com o status esperado serão alterados; os demais gerarão erro ou serão ignorados.

**Q8:** Qual é o retorno esperado em caso de sucesso?

**A8:** Um ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e mensagem informando a

reversão do status.

**Q9:** Como são tratadas exceções neste método?

**A9:** Exceções são capturadas e o método retorna “Erro geral : cancelar\_LiberadoRoteirizacao.”

**Q10:** O método realiza a alteração em lote?

**A10:** Sim, a alteração é aplicada a todos os pedidos que atendam às condições de status.

**Q11:** Quais cadastros são envolvidos na alteração?

**A11:** A operação é realizada pela fachada que utiliza internamente os cadastros de pedidos.

**Q12:** É possível reexecutar o cancelamento se ocorrer erro?

**A12:** Sim, contanto que os pedidos estejam novamente no status LIBERADOS\_PARA\_ROTEIRIZACAO e os parâmetros estejam corretos.

**Q13:** Como saber quantos pedidos foram afetados?

**A13:** O objeto de resultado informa a quantidade de pedidos atualizados.

**Q14:** Existe alguma validação adicional para a lista de pedidos?

**A14:** A lista não pode ser nula e deve conter pelo menos um pedido, caso contrário a operação não é realizada.

**Q15:** Para que situação o método *cancelar\_LiberadoRoteirizacao* deve ser usado?

**A15:** Ele é utilizado para reverter a liberação para roteirização, retornando os pedidos para o status SEPARADOS.

---

## 16. Método: cancelar\_Separados

### Validações do método cancelar\_Separados

1. **Acesso e Cliente:** Valida a senha da central e identifica o cliente.
2. **Status Atual:** Verifica que os pedidos estão com status SEPARADOS.
3. **Chamada à Fachada:** Chama o GerenciadorPedidosFachada para alterar o status para EM\_SEPARACAO.

### FAQ – cancelar\_Separados

**Q1:** Quais parâmetros são obrigatórios no método *cancelar\_Separados*?

**A1:** É necessário informar a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Qual o status atual dos pedidos para que o cancelamento seja aplicado?

**A2:** Os pedidos devem estar com status SEPARADOS.

**Q3:** Qual novo status é definido ao cancelar a separação?

**A3:** O status será alterado para EM\_SEPARACAO.

**Q4:** Como é validada a senha da central?

**A4:** Através do cadastro da Empresa; se inválida, retorna “Erro: Senha invalida.”

**Q5:** O que acontece se o cliente não for identificado?

**A5:** O método retorna um erro informando que o cliente não está cadastrado.

**Q6:** Qual é o papel do GerenciadorPedidosFachada neste cancelamento?

**A6:** Ele processa a alteração de status, revertendo de SEPARADOS para EM\_SEPARACAO.

**Q7:** O método altera todos os pedidos da lista?

**A7:** Sim, desde que os pedidos estejam com o status SEPARADOS.

**Q8:** Qual o retorno em caso de sucesso?

**A8:** Um objeto ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e mensagem informando a alteração.

**Q9:** Como são tratadas exceções?

**A9:** Exceções são capturadas e o método retorna “Erro geral : cancelar\_Separados.”

**Q10:** É possível cancelar apenas alguns pedidos?

**A10:** A alteração é aplicada em lote, porém somente os pedidos com status SEPARADOS serão afetados.

**Q11:** Quais cadastros são utilizados nesta operação?

**A11:** A operação utiliza a fachada, que invoca os cadastros de pedidos conforme necessário.

**Q12:** Posso reexecutar o cancelamento se houver erro?

**A12:** Sim, desde que os pedidos estejam no status SEPARADOS e os parâmetros estejam corretos.

**Q13:** Como confirmar a quantidade de pedidos alterados?

**A13:** O objeto de resultado informa quantos pedidos tiveram seu status modificado.

**Q14:** Há alguma validação especial para a lista de pedidos?

**A14:** A lista deve ser não nula e conter pelo menos um pedido.

**Q15:** Qual é o cenário de uso do método *cancelar\_Separados*?

**A15:** Ele é usado para reverter pedidos que já foram marcados como SEPARADOS, retornando-os para o status EM\_SEPARACAO.

---

# 17. Método: cancelar\_EmSeparacao

## Validações do método cancelar\_EmSeparacao

1. **Acesso e Cliente:** Valida a senha da central e identifica o cliente.
2. **Status Atual:** Verifica que os pedidos estão com status EM\_SEPARACAO.
3. **Chamada à Fachada:** Chama o GerenciadorPedidosFachada para alterar o status para LIBERADO\_PARA\_SEPARACAO.

## FAQ – cancelar\_EmSeparacao

**Q1:** Quais parâmetros são obrigatórios no método *cancelar\_EmSeparacao*?

**A1:** É necessário informar a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Qual o status atual exigido para este cancelamento?

**A2:** Os pedidos devem estar com status EM\_SEPARACAO.

**Q3:** Qual é o novo status aplicado?

**A3:** O status é revertido para LIBERADO\_PARA\_SEPARACAO.

**Q4:** Como o método valida a senha da central?

**A4:** Utilizando o cadastro da Empresa; se a senha for inválida, retorna “Erro: Senha invalida.”

**Q5:** O que ocorre se o cliente não for identificado?

**A5:** Retorna erro informando que o cliente não está cadastrado.

**Q6:** Qual a função do GerenciadorPedidosFachada aqui?

**A6:** Ele executa a alteração de status, revertendo de EM\_SEPARACAO para LIBERADO\_PARA\_SEPARACAO.

**Q7:** O método opera em lote?

**A7:** Sim, ele altera o status de todos os pedidos na lista que atendem à condição.

**Q8:** Qual é o retorno em caso de sucesso?

**A8:** Um ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e mensagem confirmando a alteração.

**Q9:** Como são tratadas exceções?

**A9:** Exceções são capturadas e o método retorna “Erro geral : cancelar\_EmSeparacao.”

**Q10:** O que fazer se algum pedido não estiver no status EM\_SEPARACAO?

**A10:** Somente os pedidos com o status correto serão alterados; os demais gerarão erro.

**Q11:** Quais cadastros são utilizados na operação?

**A11:** A operação é feita pela fachada que utiliza os cadastros de pedidos conforme necessário.

**Q12:** É possível reexecutar o cancelamento se ocorrer erro?

**A12:** Sim, contanto que os pedidos estejam novamente no status EM\_SEPARACAO e os parâmetros estejam corretos.

**Q13:** Como confirmar quantos pedidos foram alterados?

**A13:** O objeto de resultado informa a quantidade de pedidos atualizados.

**Q14:** Há alguma verificação extra na lista de pedidos?

**A14:** Sim, a lista não pode ser nula e deve conter pelo menos um pedido.

**Q15:** Qual cenário de uso se aplica ao método *cancelar\_EmSeparacao*?

**A15:** Ele é usado para reverter a operação de separação, retornando os pedidos para LIBERADO\_PARA\_SEPARACAO.

---

## 18. Método:

# cancelar\_LiberadosSeparacao

## Validações do método

# cancelar\_LiberadosSeparacao

1. **Acesso e Cliente:** Verifica a senha da central e identifica o cliente.
2. **Status Atual:** Confirma que os pedidos estão com status LIBERADO\_PARA\_SEPARACAO.
3. **Chamada à Fachada:** Chama o GerenciadorPedidosFachada para alterar o status para VENDA\_FINALIZADA.

## FAQ – cancelar\_LiberadosSeparacao

**Q1:** Quais parâmetros são obrigatórios no método *cancelar\_LiberadosSeparacao*?

**A1:** São necessários a lista de pedidos, CPF/CNPJ, senha do cliente e senha da central.

**Q2:** Qual o status atual exigido para o cancelamento?

**A2:** Os pedidos devem estar com status LIBERADO\_PARA\_SEPARACAO.

**Q3:** Qual o novo status após o cancelamento?

**A3:** O status será alterado para VENDA\_FINALIZADA.

**Q4:** Como o método valida a senha da central?

**A4:** A senha é validada pelo cadastro da Empresa; se estiver errada, retorna “Erro: Senha invalida.”

**Q5:** O que acontece se o cliente não for identificado?

**A5:** O método retorna um erro indicando que o cliente não está cadastrado.

**Q6:** Qual a função do GerenciadorPedidosFachada neste método?

**A6:** Ele executa a alteração do status de LIBERADO\_PARA\_SEPARACAO para VENDA\_FINALIZADA.

**Q7:** O método altera o status de todos os pedidos da lista?

**A7:** Sim, desde que todos os pedidos estejam com o status LIBERADO\_PARA\_SEPARACAO.

**Q8:** Qual é o retorno em caso de sucesso?

**A8:** Um objeto ResultadoAlteracaoStatusPedidosVO com sucesso verdadeiro e mensagem informando a alteração.

**Q9:** Como são tratadas exceções neste método?

**A9:** Exceções são capturadas e o método retorna “Erro geral : cancelar\_LiberadosSeparacao.”

**Q10:** É possível cancelar parcialmente os pedidos?

**A10:** Apenas os pedidos que estiverem com o status correto serão alterados; os demais serão ignorados ou causarão erro.

**Q11:** Quais cadastros são utilizados na operação?

**A11:** A operação utiliza a fachada, que invoca os cadastros de pedidos conforme necessário.

**Q12:** Posso reexecutar o cancelamento se ocorrer erro?

**A12:** Sim, contanto que os pedidos estejam novamente no status LIBERADO\_PARA\_SEPARACAO e os parâmetros estejam corretos.

**Q13:** Como saber quantos pedidos foram atualizados?

**A13:** O objeto de resultado retorna a quantidade de pedidos cujo status foi modificado.

**Q14:** Há alguma verificação específica para a lista de pedidos?

**A14:** A lista deve ser não nula e conter ao menos um pedido, ou a operação não será realizada.

**Q15:** Qual é o cenário de uso do método *cancelar\_LiberadosSeparacao*?

**A15:** Este método é usado para cancelar a liberação para separação, revertendo os pedidos para o status VENDA\_FINALIZADA.

---

# Considerações Finais

Cada método *WebService* da classe *ImportadorPedidos* segue um padrão de validação de acesso (senha da central e identificação do cliente) antes de executar a lógica específica de cada operação – seja conversão de CEP, importação de pedidos, consulta de status, listagens ou alterações de status. As FAQs aqui apresentadas visam responder dúvidas comuns sobre as validações, parâmetros obrigatórios e comportamentos esperados de cada método.

# FAQ - Webservice Integração - Importação de Despesas

## Validações do WebMethod

### consultarDespesas

Este método realiza as seguintes validações:

1. Verificação da senha da central (obrigatória).
2. Verificação da existência do cliente com base em CPF/CNPJ, senha e ID da central.
3. Validação do formato e existência da placa do veículo (se fornecida).
4. Validação da data inicial (formato válido e existência).
5. Validação da data final (formato válido e existência).
6. Caso algum dos dados esteja incorreto, mensagens de erro específicas são retornadas.

## Perguntas Frequentes – WebMethod

### consultarDespesas

#### **1. O que acontece se a senha da central estiver incorreta no método `consultarDespesas`?**

Será retornado um erro indicando "Senha inválida".

#### **2. O que significa o erro "cliente com CPF/CNPJ não cadastrado"?**

Indica que o CPF/CNPJ informado não está registrado na central ou que a senha está incorreta.

#### **3. Posso omitir a placa do veículo na consulta de despesas?**

Sim. A placa é opcional. Se não for informada, as despesas serão listadas para todos os veículos do cliente.

#### **4. O que acontece se eu informar a placa do veículo em um formato inválido?**

Será retornado o erro "Veiculo de placa XXX não é válido. Informe a placa no formato XXX-NNNN".

#### **5. Quais formatos de data são aceitos no `consultarDespesas`?**

Apenas o formato `yyyy-MM-dd` é aceito. Datas em outros formatos gerarão erro.

## 6. O que acontece se a data inicial for inválida?

Será retornada a mensagem "Data inicial inválida".

## 7. E se a data final estiver incorreta ou nula?

Será retornado erro: "Data final inválida".

## 8. O método `consultarDespesas` valida o vínculo entre cliente e veículo?

Sim, ele valida se a placa informada está associada ao cliente autenticado.

## 9. Se a placa estiver correta, mas não pertencer ao cliente, o que ocorre?

Será retornado erro dizendo que o veículo não é válido.

## 10. O método `consultarDespesas` retorna despesas para um intervalo de datas?

Sim, todas as despesas entre a data inicial e final informadas são retornadas.

## 11. E se não houver nenhuma despesa no intervalo?

O método retorna sucesso, porém com a lista de despesas vazia.

## 12. O método pode lançar erro geral?

Sim, em casos de falhas internas, será retornado "Erro Geral do Consulta de Despesas".

## 13. Preciso do CNPJ completo com formatação?

Sim, o método espera o CNPJ no formato `xx.xxx.xxx/xxxx-xx`.

## 14. O método `consultarDespesas` valida a senha do cliente?

Sim, a senha do cliente é obrigatória e validada junto ao CPF/CNPJ.

## 15. Posso usar CPF no lugar do CNPJ?

Sim, o método aceita tanto CPF quanto CNPJ, desde que válidos e cadastrados.

---

# Validações do WebMethod `importarDespesas`

Este método realiza as seguintes validações detalhadas:

1. Verificação se a lista de despesas foi enviada e contém ao menos um item.
2. Validação da senha da central.
3. Verificação da existência do cliente com CPF/CNPJ e senha.
4. Para cada despesa:
  - Condutor preenchido
  - Data preenchida e válida
  - Hora preenchida e válida
  - Nome informado
  - Odômetro > 0
  - Placa do veículo informada
  - Quantidade preenchida

- Tipo ou tipo genérico informado
  - Valor informado
  - Tipo de combustível informado se for abastecimento externo
  - Tipo de combustível compatível com o veículo
  - Quantidade de combustível dentro da capacidade do tanque
  - Odômetro e horímetro coerentes com as despesas anteriores/posteriores
  - Fornecedor com CNPJ/CPF válido, nome (caso for cadastrar), e existente ou a ser cadastrado corretamente
- 

## Perguntas Frequentes – WebMethod

### importarDespesas

#### **1. O que acontece se a lista de despesas estiver vazia no importarDespesas?**

Será retornado erro: "A lista de despesas deve conter pelo menos um elemento."

#### **2. Como é validado o condutor?**

O método exige que o campo de condutor esteja preenchido em cada despesa.

#### **3. Preciso informar a data da despesa?**

Sim. A data é obrigatória e deve estar no formato válido.

#### **4. E se a hora estiver em formato incorreto?**

Será retornado erro: "Hora inválida da despesa X".

#### **5. Qual o erro se o nome da despesa estiver ausente?**

Erro: "Informe o NOME da despesa X".

#### **6. Posso registrar uma despesa com odômetro igual a zero?**

Não. O campo deve ser maior que zero para ser aceito.

#### **7. A placa do veículo é obrigatória na importação?**

Sim. Deve ser informada e válida para cada despesa.

#### **8. O que ocorre se a quantidade for nula?**

Será retornado erro indicando que a quantidade da despesa deve ser informada.

#### **9. E se eu esquecer de preencher o tipo da despesa?**

Erro: "Informe o TIPO da despesa X".

#### **10. Como funciona a validação de tipo de combustível?**

Se for abastecimento externo, o tipo de combustível deve ser informado, compatível com o veículo e dentro da capacidade do tanque.

**11. O que acontece se o tipo de combustível não for compatível?**

Erro: "Combustível não compatível com o veículo da despesa X".

**12. O método `importarDespesas` verifica odômetro com despesas anteriores?**

Sim. O sistema impede valores inferiores aos das despesas anteriores e superiores aos das posteriores.

**13. Posso importar despesas com fornecedor novo?**

Sim, desde que esteja marcado para cadastrar e com CNPJ/CPF e nome preenchidos corretamente.

**14. O que acontece se o CPF ou CNPJ do fornecedor for inválido?**

Erro: "CPF ou CNPJ inválido do Fornecedor da despesa X".

**15. O método pode lançar erro geral?**

Sim, será retornado "Erro Geral do Importador de Despesas" com detalhes do problema.

# FAQ - Webservice Integração - Automatizador de Roteirização

## Validações do WebMethod `roteirizarPedidos`

O método `roteirizarPedidos` realiza as seguintes validações, retornando mensagens de erro específicas em caso de falha:

1. **Senha da central inválida**
2. **Cliente não localizado na central com CPF/CNPJ e senha**
3. **Roteirização anterior sincronizada em dispositivo móvel**
4. **Falha ao excluir roteirização anterior**
5. **Roteirização não encontrada pelo ID informado**
6. **Placa de veículo não cadastrada ou fora do formato válido**
7. **Motorista não cadastrado no sistema**
8. **Data e hora de início da rota nulas**
9. **Código de ponto de interesse de origem inválido ou não cadastrado**
10. **Código de ponto de interesse de destino inválido ou não cadastrado**
11. **Nenhum carregamento informado**
12. **Nenhum pedido encontrado nos carregamentos informados**
13. **Pedidos sem código de ponto de interesse quando agrupamento está ativo**
14. **Falha ao chamar API de roteirização (Google Directions API)**
15. **Erro ao inserir viagem ou itinerários na base de dados (com rollback)**

---

## Perguntas Frequentes – WebMethod

### `roteirizarPedidos`

**1. O que acontece se a senha da central estiver incorreta ao usar o `roteirizarPedidos`?**  
Será retornada a mensagem "Erro: Senha inválida."

**2. Como o sistema valida o cliente na chamada do `roteirizarPedidos`?**

O cliente deve estar vinculado à central e ser identificado pelo CPF/CNPJ e senha informados.

### **3. O que significa o erro “Essa rota já foi sincronizada em um dispositivo móvel”?**

Indica que uma rota existente foi sincronizada e precisa ser removida manualmente para permitir nova roteirização.

### **4. É obrigatório informar um ID de rota no `roteirizarPedidos`?**

Não. Se informado, o sistema tenta excluir a roteirização anterior vinculada a esse ID.

### **5. O que ocorre se o ID de rota não for encontrado?**

Será retornada a mensagem “Roteirização não encontrada.”

### **6. Como deve ser o formato da placa do veículo no método `roteirizarPedidos`?**

A placa deve seguir o padrão `XXX-XXXX` e o veículo deve estar cadastrado.

### **7. E se o nome do motorista estiver incorreto?**

Será retornado erro dizendo que o motorista não está cadastrado.

### **8. A data/hora de início da rota pode ser nula?**

Não. É obrigatória. Caso contrário, será retornado "Data e Hora de Início da Rota Inválida."

### **9. Como o sistema valida os pontos de interesse de origem e destino?**

Verifica se os códigos estão cadastrados e, se houver geometria, utiliza o centróide para coordenadas.

### **10. O que acontece se não houver carregamentos informados?**

Erro: “Nenhum carregamento informado.”

### **11. E se nenhum pedido for localizado nos carregamentos informados?**

Erro: “Nenhum pedido cadastrado nos carregamentos informados.”

### **12. Quando é necessário o código de ponto de interesse no pedido?**

Se a opção `agruparPedidosMesmoCliente` estiver ativada, todos os pedidos devem conter o código Pol.

### **13. O método pode agrupar pedidos por zona ou cliente?**

Sim. Ele agrupa por zona automaticamente e, se configurado, também por cliente com base no código Pol.

### **14. A roteirização pode falhar por limitações externas?**

Sim. A comunicação com a API do Google pode falhar e retornar "Não é possível roteirizar os pedidos."

### **15. O que acontece em caso de falha na gravação da viagem ou itinerários?**

O sistema realiza `rollback` na transação e retorna “Erro na inserção da viagem e/ou itinerários.”

## Validações do WebMethod `listarMotoristas`

1. Validação da senha da central
  2. Validação do cliente (CPF/CNPJ + senha do cliente)
  3. Captura de erro geral caso ocorra falha durante a execução
- 

## Perguntas Frequentes – WebMethod

### listarMotoristas

1. O que faz o método `listarMotoristas`?  
Retorna a lista de motoristas cadastrados para o cliente autenticado.
  2. A senha da central é obrigatória no `listarMotoristas`?  
Sim, e será validada antes da execução.
  3. O que acontece se a senha da central estiver errada?  
Retorna erro informando "Senha inválida".
  4. O cliente precisa estar cadastrado na central?  
Sim. Caso contrário, retorna erro "cliente com CPF/CNPJ não cadastrado".
  5. O método `listarMotoristas` retorna motoristas de outros clientes?  
Não, somente os motoristas vinculados ao cliente autenticado.
  6. Posso usar CPF ou CNPJ no `listarMotoristas`?  
Sim, desde que esteja vinculado ao cliente e central.
  7. O que acontece se houver um erro interno?  
Será retornada mensagem "Erro ao consultar lista de motoristas".
  8. Preciso informar a senha do cliente?  
Sim, a autenticação do cliente é obrigatória.
  9. Quantos motoristas o método pode retornar?  
Todos os cadastrados para o cliente.
  10. O método `listarMotoristas` retorna motoristas inativos?  
A documentação não especifica; depende da lógica interna do DAO.
  11. O método retorna lista vazia se não houver motoristas?  
Sim, com mensagem de sucesso e lista vazia.
  12. O método `listarMotoristas` exige parâmetros adicionais?  
Não, apenas CPF/CNPJ, senha do cliente e senha da central.
  13. Qual o formato da resposta?  
Um objeto contendo sucesso, mensagem e lista de motoristas.
  14. O método `listarMotoristas` pode ser consumido via SOAP?  
Sim, pois está anotado com `@WebMethod`.
  15. O que fazer se o método sempre retornar erro?  
Verificar credenciais, conexão com banco e integridade dos dados.
- 

## Validações do WebMethod `listarVeiculos`

1. Validação da senha da central
  2. Validação do cliente (CPF/CNPJ + senha do cliente)
  3. Tratamento de erro geral em caso de exceções
- 

## Perguntas Frequentes – WebMethod

### listarVeiculos

1. O que retorna o método `listarVeiculos`?  
Uma lista de veículos vinculados ao cliente autenticado.
  2. A senha da central é obrigatória?  
Sim. Sem ela, o método retorna erro.
  3. E se a senha estiver incorreta?  
Erro: "Senha inválida".
  4. O cliente precisa estar vinculado à central?  
Sim. Caso contrário, não será possível listar os veículos.
  5. O método retorna veículos de outros clientes?  
Não. Apenas do cliente autenticado.
  6. O que é necessário para chamar `listarVeiculos`?  
CPF/CNPJ, senha do cliente e senha da central.
  7. O método retorna lista vazia se não houver veículos?  
Sim, com mensagem de sucesso.
  8. O método pode lançar erro interno?  
Sim, com a mensagem "Erro ao consultar lista de veículos".
  9. A resposta inclui todos os dados dos veículos?  
Inclui os principais dados conforme definidos no DAO.
  10. Pode ser usado em sistemas externos?  
Sim, via SOAP.
  11. O método exige permissão especial?  
Apenas autenticação válida.
  12. O método `listarVeiculos` valida os parâmetros?  
Sim. Senha e cliente são obrigatórios.
  13. Pode ser testado por ferramentas como SoapUI?  
Sim.
  14. Os veículos precisam estar ativos para aparecer?  
Depende da implementação no DAO.
  15. O método afeta o banco de dados?  
Não. É apenas consulta.
- 

## Validações do WebMethod `cadastrarZona`

1. Validação da senha da central
  2. Validação do cliente (CPF/CNPJ + senha do cliente)
  3. Nome da zona com no mínimo 3 caracteres
  4. Código da zona obrigatório e não vazio
  5. Verificação se a zona já existe: se sim, atualiza; se não, insere
  6. Tratamento de erro geral
- 

## Perguntas Frequentes – WebMethod

### cadastrarZona

1. O que faz o método `cadastrarZona`?  
Cria ou atualiza zonas de roteirização para o cliente.
  2. O nome da zona pode ter menos de 3 caracteres?  
Não. Gera erro se for menor.
  3. O código da zona pode ser vazio?  
Não. O campo é obrigatório.
  4. E se a zona já existir?  
O sistema tenta atualizá-la.
  5. O que é retornado se a zona for atualizada?  
Mensagem de sucesso indicando atualização.
  6. Como saber se a zona foi realmente modificada?  
A resposta indica se houve alteração nos dados.
  7. Pode-se cadastrar zonas repetidas?  
Não com o mesmo código.
  8. O método `cadastrarZona` exige autenticação?  
Sim, tanto do cliente quanto da central.
  9. É possível cadastrar múltiplas zonas de uma vez?  
Não. Cada chamada cadastra uma zona.
  10. E se ocorrer erro na conexão com o banco?  
Será retornado erro geral.
  11. Como saber se a zona foi inserida com sucesso?  
A mensagem retorna "inserida com sucesso".
  12. É permitido usar apenas o nome sem código?  
Não. Ambos são obrigatórios.
  13. O código da zona pode conter letras e números?  
Sim, desde que seja válido e único.
  14. O método pode ser usado via SOAP?  
Sim.
  15. Qual tipo de retorno esse método fornece?  
Um objeto com sucesso, mensagem e ID, se aplicável.
-

# Validações do WebMethod `cadastrarLoja`

1. Validação da senha da central
  2. Validação do cliente
  3. Nome da loja com no mínimo 3 caracteres
  4. Código da loja obrigatório
  5. Verificação de existência para decidir entre inserir ou atualizar
  6. Tratamento de erro geral
- 

## Perguntas Frequentes – WebMethod `cadastrarLoja`

(Semelhantes ao `cadastrarZona`, adaptadas para "loja")

1. Posso cadastrar uma loja sem nome?  
Não. Nome com pelo menos 3 caracteres é obrigatório.
2. E se o código estiver ausente?  
Erro: "Informe o código da loja".
3. Posso atualizar uma loja existente?  
Sim. Se o código existir, ela será atualizada.
4. Como saberei se a loja foi alterada ou inserida?  
A mensagem informa o tipo de operação.
5. O método `cadastrarLoja` pode duplicar lojas?  
Não com o mesmo código.
6. O nome da loja pode conter espaços?  
Sim, sem problema.
7. O método exige CPF/CNPJ?  
Sim. É necessário autenticar o cliente.
8. O código da loja deve seguir algum padrão?  
Não, mas deve ser único.
9. Há limite de lojas por cliente?  
Não especificado.
10. A chamada afeta outras lojas?  
Não. Apenas a loja da requisição.
11. Posso usar esse método por integração externa?  
Sim, via SOAP.
12. Como lidar com erro na inserção?  
Verifique conexão, dados e permissões.
13. Posso cadastrar lojas com acentos?  
Sim.
14. A loja precisa estar ativa?  
Não há validação para isso.

15. O método retorna a loja criada?

Não diretamente. Retorna apenas mensagem de sucesso ou erro.

---

## Validações do WebMethod

### cadastrarVendedor

1. Nome com no mínimo 3 caracteres
  2. Código do vendedor obrigatório
  3. Cliente e central devem ser válidos
  4. Se existir, atualiza; se não, insere
  5. Tratamento de erros
- 

## Perguntas Frequentes – WebMethod

### cadastrarVendedor

1. O nome do vendedor pode ser curto?  
Não. Deve ter ao menos 3 caracteres.
2. E o código pode estar vazio?  
Não. É obrigatório.
3. O que acontece se o vendedor já existir?  
Ele será atualizado.
4. Como saber se foi inserido ou alterado?  
A mensagem retorna essa informação.
5. E se não ocorrer nenhuma mudança?  
Mensagem indicará que os dados são os mesmos.
6. Posso cadastrar vendedor com mesmo código e nome diferentes?  
Não recomendado. Código deve ser único.
7. Pode-se usar esse método para integração externa?  
Sim.
8. O método retorna ID?  
Não explicitamente. Apenas mensagens.
9. O CPF do cliente é validado?  
Sim, como em todos os métodos.
10. O nome do vendedor pode conter símbolos?  
Sim, se for coerente com o negócio.
11. Há validações para e-mail ou telefone?  
Não nesse método.
12. O que fazer em caso de erro?  
Verificar logs e parâmetros informados.

13. Posso atualizar o código de um vendedor?  
Não diretamente. É melhor excluir e recriar.
  14. É possível cadastrar dois vendedores com mesmo nome?  
Sim, desde que o código seja diferente.
  15. O método pode ser testado em homologação?  
Sim, se o ambiente estiver configurado.
- 

## Validações do WebMethod

### cadastRARMotorista

1. Nome com no mínimo 3 caracteres
  2. Obrigatório CPF com pelo menos 11 dígitos ou CNH com pelo menos 4
  3. Cliente e central devem ser válidos
  4. Se motorista já existe por CPF ou CNH, atualiza
  5. Caso contrário, insere novo
  6. Tratamento de erros e rollback
- 

## Perguntas Frequentes – WebMethod

### cadastRARMotorista

1. É obrigatório CPF ou CNH no `cadastRARMotorista`?  
Sim. Pelo menos um deles.
2. O nome precisa ter mínimo de caracteres?  
Sim. Ao menos 3.
3. O método identifica duplicidade por CPF?  
Sim. Verifica CPF e, se necessário, CNH.
4. O que ocorre se os dados forem os mesmos?  
Mensagem indica que não houve alteração.
5. E se ocorrer erro no banco?  
Retorna erro geral com log.
6. O método retorna o motorista inserido?  
Não. Apenas mensagem de sucesso ou erro.
7. A CNH pode ser usada sem CPF?  
Sim, desde que tenha ao menos 4 caracteres.
8. E se houver erro de autenticação?  
O método falha com mensagem apropriada.
9. A operação é reversível?  
Não diretamente. Requer nova chamada com dados corretos.
10. O código da operação é retornado?  
Não. Apenas a mensagem.

11. O motorista pode ter o mesmo nome de outro?  
Sim. Desde que CPF ou CNH sejam diferentes.
12. A inserção exige e-mail ou telefone?  
Não obrigatoriamente.
13. Há verificação de duplicidade com outros clientes?  
Não. A validação é dentro do cliente.
14. O método altera dados sensíveis?  
Sim. Por isso requer autenticação completa.
15. Pode-se cadastrar via integração?  
Sim. Ideal para automações.