

# API Setup

Versão	Data	Descrição
1.0	01-Maio-2017	Criação
1.1	12-Junho-2017	Descrição detalhada dos atributos do pedido
1.2	05-Julho-2017	Inclusão de mais um parâmetro de senha
1.3	10-Outubro-2017	Método para consulta do status do pedido
1.4	26-Outubro-2017	Inclusão do número do carregamento do pedido
1.5	04-Dezembro-17	Inclusão do número GSM para envio de alertas.
1.6	14-Dezembro-17	Inclusão do número da nota fiscal e data/hora de compromisso de entrega.
1.7	03-Janeiro-18	Alteração do tipo do campo data/hora compromisso de entrega. Inclusão do número do lote
1.8	15-Janeiro-18	Inclusão do numero da viagem da qual o pedido está associado.
1.9	04-Fevereiro-18	Método para alteração do número de carregamento de itinerários
2.0	20-Julho-18	Inclusão de método para listar os pedidos de uma rota. Novo endpoint para roteirização automática
2.1	17-Set-18	Inclusão dos atributos codigoZona, codigoVendedor e codigoLoja.
2.2	07-Nov-18	Criação dos métodos para listar o nome dos motoristas e veículos cadastrados.
2.3	10-Dez-18	Inclusão de opção para agrupar pedidos de um mesmo ponto de entrega e desconsiderar o tempo de atendimento.
2.4	04-Jan-19	Inclusão do atributo motorista do veículo nos metodos para listar os veículos cadastrados.
2.5	20-Ago-19	Inclusão do método para retorna a lista de pontos de interesse roteirizados.
2.6	04-Set-19	Método para importação de despesas.

2.7	19-Mai-21	Conversão de Endereços por CEP e importação de pontos de interesse.
-----	-----------	---------------------------------------------------------------------



# Métodos

## Introdução:

Documentação descritiva da API para importação de pedidos no sistema de rastreamento e gestão de frotas.

A tecnologia utilizada é *Webservice – SOAP*. Para ter acesso ao webservice é necessária uma senha fornecida pela Concept Tecnologia e liberação no firewall.

Entrar em contato pelo email [suporte@concept.inf.br](mailto:suporte@concept.inf.br) ou (86) 3301-1878.

## 1. Importador de Pedidos

<b>WSDL</b>
<a href="http://52.6.27.50:8181/importadorPedidos?wsdl">http://52.6.27.50:8181/importadorPedidos?wsdl</a>

### 1.1 Método: Importar Pedidos

<b>Método</b>
importarPedidos

Parâmetros	Tipo
listaPedidos	List<PedidoRotaVO>
cpfCNPJ	String
senhaCliente	String
senhaCentral	String

### ListaPedidos

Lista dos pedidos a serem inseridos. Para cada PedidoRotaVO é necessário informar:

PedidoRotaVO	Tipo	Descrição
numeroPedido	String	Número do pedido. Chave única.
valorPedido	double	Valor do pedido.
codigoPontoInteresse <sup>1</sup>	String	Código do cliente.
latitude <sup>1</sup>	String	Latitude do local de entrega.

CONFIDENCIAL

longitude <sup>1</sup>	String	Longitude do local de entrega
endereco <sup>1</sup>	String	Endereço do local de entrega.
dataPedido	Date	Data do pedido.
horaPedido	String	Hora do Pedido
descricao	String	Descrição adicional ao pedido.
qtdItensPedido	double	Qtd de itens do pedido.
peso <sup>2</sup>	double	Peso em kg total do pedido.
volume <sup>2</sup>	double	Volume em m3 do pedido.
horaEntregaInicial <sup>3</sup>	String	Horário inicial permitido para a entrega.
horaEntregaFinal <sup>3</sup>	String	Horário final permitido para a entrega.
tempoAtendimento	Int	Tempo estimado para fazer a entrega.
zona <sup>4</sup>	ZonaVO	Zona onde será feita a entrega.
loja <sup>4</sup>	LojaVO	Loja que fará a entrega.
vendedor <sup>4</sup>	VendedorVO	Vendedor que fez o pedido.
numeroCarregamento	String	Número do agrupador ou carregamento.
numeroGSM	String	Número GSM/Celular para enviar alertas(SMS)
numeroNotaFiscal	String	Número da Nota Fiscal do pedido
dataCompromissoEntrega	Date	Data de compromisso da entrega
horaCompromissoEntrega	String	Hora de compromisso da entrega
numeroLote	String	Número de lote ( controle cliente )
cadastrarPontoInteresse	Boolean	Flag para cadastramento do ponto de interesse
poi	PontosInteresseVO	Ponto de Interesse a ser cadastrado

<sup>1</sup> A informação do local de entrega pode ser dada através do código do cliente, latitude/longitude ou endereço. Caso o código do cliente seja informado não há necessidade de informar latitude/longitude ou endereço desde que o cliente esteja cadastrado no sistema de rastreamento como ponto de interesse e com o código informado. Atentar que o endereço de entrega pode ser diferente do endereço do cliente. Neste caso não informar o código do cliente (codigoPontoInteresse) e preencher o campo *endereço ou latitude/longitude*.

<sup>2</sup> Os campos peso e volume são utilizados para a roteirização que irá considerar a capacidade do veículo. Não são obrigatórios e podem ter o valor 0 (zero).

<sup>3</sup> Esses campos são utilizados para restringir o horário que a entrega pode ser feita. Serão utilizados na roteirização. Caso não sejam pertinentes informar os valores "00:00" e "23:59"

<sup>4</sup> Informar os nomes das Zonas, Lojas e Vendedores que devem estar cadastrados no sistema da Concept previamente. Deve-se informar o código ou o nome para a importação

Campos em **vermelho** são obrigatórios.

### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

### senhaCliente

Senha fornecida pela Concept Tecnologia.

### senhaCentral

Senha fornecida pela Concept Tecnologia.

## Retorno

O retorno do método é sempre um objeto da classe ResultadoImportacaoVO

Condição	Atributos	
Sucesso	<code>comSucesso = true</code>	mensagem: "Importação de X pedidos realizada com sucesso"
Erro	<code>comSucesso = false</code>	mensagem: {descrição do erro}.

ZonaRotaVO	Tipo	Descrição
codigoZona	String	Código da Zona
nome	String	Nome da Zona

LojaRotaVO	Tipo	Descrição
codigoLoja	String	Código da Loja
nome	String	Nome da Loja

VendedorRotaVO	Tipo	Descrição
codigoVendedor	String	Código do Vendedor
nome	String	Nome do Vendedor

GrupoPontoInteresseVO	Tipo	Descrição
nome	String	Nome do Grupo

Campos em **vermelho** são obrigatórios.

PontosInteresseVO	Tipo	Descrição
<b>codigo</b>	String	Código do Ponto de Interess
<b>nome</b>	String	Nome do Ponto de Interesse
<b>descricao</b>	String	Descrição do Ponto de Interesse

CONFIDENCIAL

latitude	String	Latitude
longitude	String	Longitude
identificar	boolean	Identificar nos avisos e percurso ?
enviarAlerta	boolean	Enviar alerta ao ser visitado
raio	double	Raio em KM
horaInicialRestricao	Time	Horário início permitido de visita
horaFinalRestricao	Time	Horário fim permitido de visita
grupo	GrupoPontoInteresseVO	Grupo do Ponto de Interesse

## 1.2 Método: Consultar o Status da Entrega

Método
consultarStatusEntregaPedido

Parâmetros	Tipo
pedido cpfCNPJ senhaCliente senhaCentral	PedidoRotaVO String String String

### pedido

Pedido a ser consultado. Para o PedidoRotaVO é necessário informar:

PedidoRotaVO	Tipo	Descrição
numeroPedido	String	Número do pedido. Chave única.

Campos em **vermelho** são obrigatórios.

### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

### senhaCliente

Senha fornecida pela Concept Tecnologia.

### senhaCentral

Senha fornecida pela Concept Tecnologia.

## Retorno

O retorno do método é sempre um objeto da classe ResultadoConsultaStatusVO

Condição	Atributos		
Sucesso	comSucesso true	mensagem: "Pedido nunca foi roteirizado para entrega" "Pedido encontrado."	itinerarioViagem ItinerarioViagemVO
Erro	comSucesso false	mensagem: {descrição do erro}.	itinerarioViagem null

ItinerarioViagemVO	Tipo	Descrição
--------------------	------	-----------

CONFIDENCIAL

numeroPedido	String	Número do pedido. Chave única.
dataItinerarioViagem	Date	Data prevista da entrega
status	StatusItinerarioViagem	Status da entrega
horaEntrega	Timestamp	Data e hora que foi realizada a entrega
horaRota	Time	Hora Prevista da Entrega
idViagem	int	Numero da viagem correspondente

StatusItinerarioViagem	Tipo	Descrição
	ENUM	ABERTO ENTREGUE PENDENTE ENDERECO_FECHADO ENDERECO_ERRADO ENDERECO_NAO_LOCALIZADO ADIADO
Descricao	String	Descricao do Status



### 1.3 Método: Atualizar Número do Carregamento

Método
atualizarNumeroCarregamento

Parâmetros	Tipo
numeroCarregamentoInterno	int
numeroCarregamentoNovo	String
cpfCNPJ	String
senhaCliente	String
senhaCentral	String

#### numeroCarregamentoInterno

Número do carregamento antigo obtido através do método *consultarStatusEntregaPedido* e retornado dentro do atributo *idViagem* no objeto *ItinerarioViagemVO*. Esse número não é modificável mesmo que o método *atualizarNumeroCarregamento* seja chamado.

#### numeroCarregamentoNovo

Número do carregamento a ser atualizado nos itinerários viagens.

#### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

#### senhaCliente

Senha fornecida pela Concept Tecnologia.

#### senhaCentral

Senha fornecida pela Concept Tecnologia.

### Retorno

O retorno do método é sempre um objeto da classe *ResultadoAtualizacaoCarregamentoVO*

Condição	Atributos	
Sucesso	<i>comSucesso</i> true	<i>mensagem:</i> "Atualização do carregamento x para y efetuada com sucesso. Z pedidos atualizados."
Erro	<i>comSucesso</i> false	<i>mensagem:</i> {descrição do erro}.

### 1.4 Método: Listar Pedidos Roteirizados

<b>Método</b>
<b>listarItinerariosRoteirizado</b>

Parâmetros	Tipo
codigoViagemRoteirizada	int
cpfCNPJ	String
senhaCliente	String
senhaCentral	String

### numeroCarregamentoInterno

Número da viagem roteirizado obtido após a roteirização invocada no método do automatizador.

### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

### senhaCliente

Senha fornecida pela Concept Tecnologia.

### senhaCentral

Senha fornecida pela Concept Tecnologia.

### Retorno

O retorno do método é sempre um objeto da classe ResultadoConsultaListaltinerariosVO

Condição	Atributos	
Sucesso	comSucesso true	<p><b>mensagem:</b> "Lista de Itinerarios recuperada com sucesso."</p> <p><b>lista:</b> Lista de pedidos na ordem de sequenciamento da roteirização. São objetos da classe ItinerarioViagemVO.</p>
Erro	comSucesso false	<p><b>mensagem:</b> {descrição do erro}.</p> <p><b>lista:</b> Null</p>

ItinerarioViagemVO	Tipo	Descrição
numeroPedido	String	Número do pedido. Chave única.

CONFIDENCIAL

dataItinerarioViagem	Date	Data prevista da entrega
status	StatusItinerarioViagem	Status da entrega
horaEntrega	Timestamp	Data e hora que foi realizada a entrega
horaRota	Time	Hora Prevista da Entrega
idViagem	int	Numero da viagem correspondente

## 1.5 Método: Listar Pontos de Interesse Roteirizados

Método
listarPolRoterizado

Parâmetros	Tipo
codigoViagemRoteirizada	int
cpfCNPJ	String
senhaCliente	String
senhaCentral	String

### numeroCarregamentoInterno

Número da viagem roteirizado obtido após a roteirização invocada no método do automatizador.

### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

### senhaCliente

Senha fornecida pela Concept Tecnologia.

### senhaCentral

Senha fornecida pela Concept Tecnologia.

## Retorno

O retorno do método é sempre um objeto da classe ResultadoConsultaListaltinerariosVO

Condição	Atributos	
Sucesso	comSucesso true	<p>mensagem: "Lista de Pontos de Interesse recuperada com sucesso.."</p> <p>lista: Lista de Pontos de Interesse na ordem de sequenciamento da roteirização. São objetos da classe PontosInteresseVO.</p>
Erro	comSucesso false	<p>mensagem: {descrição do erro}.</p> <p>lista: Null</p>

PontosInteresseVO	Tipo	Descrição
codigo	String	Número do pedido. Chave única.

CONFIDENCIAL

nome	Date	Data prevista da entrega
------	------	--------------------------

## 1.6 Método: Converter endereço usando CEP

Método
converterEnderecoUsandoCEP

Parâmetros	Tipo
listaPedidos cpfCNPJ senhaCliente senhaCentral	List<PedidoRotaVO> String String String

### pedido

Pedido a ser consultado. Para o PedidoRotaVO é necessário informar:

PedidoRotaVO	Tipo	Descrição
numeroPedido	String	Número do pedido. Chave única.
cep	String	Número do CEP somente os números. Ex: 00124500

Campos em **vermelho** são obrigatórios.

### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

### senhaCliente

Senha fornecida pela Concept Tecnologia.

### senhaCentral

Senha fornecida pela Concept Tecnologia.

### Retorno

O retorno do método é sempre um objeto da classe ResultadoConsultaStatusVO

Condição	Atributos		
Sucesso	comSucesso true	mensagem: "{X} Pedidos convertidos com sucesso"	listaPedidos List<PedidoRotaVO>
Erro	comSucesso false	mensagem: {descrição do erro}.	listaPedidos null

PedidoRotaVO	Tipo	Descrição
numeroPedido	String	Número do pedido. Chave única.
endereço	String	Endereço convertido
latitude	String	Latitude do endereço
longitude	String	Longitude do Endereço

## Roteirizador de Pedidos

<b>WSDL</b>
http://52.6.27.50:8181/ automatizador?wsdl

### 1.1 Método: Roteirizar Pedidos

<b>Método</b>
roteirizarPedidos

Parâmetros	Tipo
idViagem	int
listaCarregamentos	List<String>
codigoPontoInteresseOrigem	String
codigoPontoInteresseDestino	String
dataHoraInicioRota	Date
placaVeiculo	String
nomeCondutor	String
isAgruparPedidosMesmoCliente	boolean
cpfCNPJ	String
senhaCliente	String
senhaCentral	String

#### **idViagem**

Para novas viagens passe o valor 0 ( zero ). Para atualizar uma viagem passe o código retornado na roteirização inicial. A viagem será excluída e criada novamente.

#### **listaCarregamentos**

Lista dos carregamentos que contem os pedidos a serem Roterizados.

#### **codigoPontoInteresseOrigem**

Código do ponto de interesse cadastrado na plataforma da Concept referente ao local de partida do veículo que irá transportar os pedidos.

#### **codigoPontoInteresseDestino**

Código do ponto de interesse cadastrado na plataforma da Concept referente ao local de retorno do veículo que irá transportar os pedidos.

#### **dataHoraInicioRota**



Data e Hora planejada para início da viagem.

**placaVeiculo**

Placa do veículo que irá transportar os pedidos. O veículo precisa ser rastreador pela plataforma da Concept.

**nomeCondutor**

Nome do motorista do veículo que fará a viagem.

**isAgruparPedidosMesmoCliente**

Parâmetro opcional. Se o valor for true, fará que na roteirização o tempo de atendimento do primeiro pedido seja considerado como o tempo total de atendimento no ponto de interesse de todos os pedidos atendidos no mesmo local.

**cpfCNPJ**

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

**senhaCliente**

Senha fornecida pela Concept Tecnologia.

**senhaCentral**

Senha fornecida pela Concept Tecnologia.

**Retorno**

O retorno do método é sempre um objeto da classe ResultadoRoteirizacaoVO

Condição	Atributos	
Sucesso	comSucesso true	mensagem: "Pedidos Roteirizados com sucesso." idRoteirizacao: Identificador da roteirização criada.
Erro	comSucesso false	mensagem: {descrição do erro}. idRoteirizacao: 0

**1.2 Método: Listar Motoristas**

Método
listarMotoristas

Parâmetros	Tipo
cpfCNPJ senhaCliente senhaCentral	String String String

### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

### senhaCliente

Senha fornecida pela Concept Tecnologia.

### senhaCentral

Senha fornecida pela Concept Tecnologia.

## Retorno

O retorno do método é sempre um objeto da classe ResultadoConsultaMotoristasVO

Condição	Atributos	
Sucesso	comSucesso true	mensagem: "Pedidos Roteirizados com sucesso." lista: Lista de objetos MotoristaVO
Erro	comSucesso false	mensagem: {descrição do erro}. lista: null

MotoristaVO	Tipo	Descrição
id	Int	Identificador do Motorista. Chave única.
nome	String	Nome do Motorista
cpf	String	CPF
cnh	String	Número da Carteira Nacional de Habilitação

### 1.3 Método: Listar Veiculos

Método
listarVeiculos

Parâmetros	Tipo
cpfCNPJ senhaCliente senhaCentral	String String String

#### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

#### senhaCliente

Senha fornecida pela Concept Tecnologia.

#### senhaCentral

Senha fornecida pela Concept Tecnologia.

### Retorno

O retorno do método é sempre um objeto da classe ResultadoConsultaMotoristasVO

Condição	Atributos	
Sucesso	comSucesso true	mensagem: "Pedidos Roteirizados com sucesso." lista: Lista de objetos VeiculoRoteirizacaoVO
Erro	comSucesso false	mensagem: {descrição do erro}. lista: null

VeiculoRoteirizacaoVO	Tipo	Descrição
id	Int	Identificador do Veículo. Chave única.
placa	String	Placa do Veículo
motorista	MotoristaVO	Motorista atual do veículo

## Importador de Despesas

<b>WSDL</b>
http://52.6.27.50:8181/ importadorDespesas?wsdl

### 1.1 Método: Roteirizar Pedidos

<b>Método</b>
importarDespesas

Parâmetros	Tipo
listaDespesas cpfCNPJ senhaCliente senhaCentral	List< DespesaVO> String String String

#### listaDespesas

Lista das despesas que devem ser importadas. Cada despesa é um objeto da classe DespesaVO.

#### cpfCNPJ

CPF ou CNPJ do cliente no qual serão importados os pedidos. Esse CPF/CNPJ deve ser o mesmo cadastrado no sistema da central de rastreamento.

#### senhaCliente

Senha fornecida pela Concept Tecnologia.

#### senhaCentral

Senha fornecida pela Concept Tecnologia.

#### Retorno

O retorno do método é sempre um objeto da classe ResultadoImportacaoVO

Condição	Atributos	
Sucesso	comSucesso true	mensagem: "Importação de x despesas realizada com sucesso." idRoteirizacao: Identificador da roteirização criada.
Erro	comSucesso	mensagem:

CONFIDENCIAL

	false	{descrição do erro}.
--	-------	----------------------

DespesaVO	Tipo	Descrição
condutor	String	Nome do condutor do veículo referente a despesa.
placa	String	Placa do Veículo
data	Date	Data da despesa
hora	String	Hora da despesa no formato HH:MM
nome	String	Título para despesas.
descricao	String	Texto com a descrição da despesa.
odometro	Int	Odômetro do veículo quando ocorreu a despesa.
quantidade	BigDecimal	Quantidade da despesa
tipo	Enum	Tipo da despesa de acordo com o enum disponível no TipoDespesa.
valor	BigDecimal	Valor unitário da despesa
tipoCombustivel	Enum	Tipo do combustível referente a tipo de despesa Abastecimento, AbastecimentoExternoAVista, AbastecimentoExternoCartao de acordo com o enum TipoCombustivel

## EXEMPLOS

## Exemplo 1 : Código Java para Importação

```

package teste;

import java.util.ArrayList;
import java.util.GregorianCalendar;

import javax.xml.datatype.DatatypeConfigurationException;
import javax.xml.datatype.DatatypeFactory;
import javax.xml.datatype.XMLGregorianCalendar;

import concept.gps.wsclient.ImportadorPedidos;
import concept.gps.wsclient.ImportadorPedidosService;
import concept.gps.wsclient.LojaRotaVO;
import concept.gps.wsclient.PedidoRotaVO;
import concept.gps.wsclient.ResultadoImportacaoVO;
import concept.gps.wsclient.VendedorRotaVO;
import concept.gps.wsclient.ZonaRotaVO;

public class Testador {

    /**
     * @param args
     */
    public static void main(String[] args) {
        try {
            ImportadorPedidos importadorWS = new ImportadorPedidosService().getImportadorPedidosPort();

            LojaRotaVO loja = new LojaRotaVO();
            loja.setNome("Filial 1");

            VendedorRotaVO vendedor = new VendedorRotaVO();
            vendedor.setNome("Edson");

            ZonaRotaVO zona = new ZonaRotaVO();
            zona.setNome("Zona Leste");

            GregorianCalendar c = new GregorianCalendar();
            c.setTime(new java.util.Date());
            XMLGregorianCalendar date2;

            date2 = DatatypeFactory.newInstance().newXMLGregorianCalendar(c);

            PedidoRotaVO pedido = new PedidoRotaVO();
            pedido.setNumeroPedido("12345");
            pedido.setCodigoPontoInteresse("1");
            pedido.setDataPedido(date2);
            pedido.setDescricao("Pedido importado pelo Webservice");
            pedido.setEndereco("Rua Governador Joca Pires, 1526, Teresina - Piauí");
            pedido.setHoraPedido("14:32");
            pedido.setHoraEntregainicial("8:00");
            pedido.setHoraEntregaFinal("18:00");
            pedido.setTempoAtendimento(22);
            pedido.setLatitude("12.3456");
            pedido.setLoja(loja);
            pedido.setZona(zona);
            pedido.setVendedor(vendedor);
            pedido.setNumeroCarregamento("C1");
        }
    }
}

```

## CONFIDENCIAL

```
PedidoRotaVO pedido2 = new PedidoRotaVO();
pedido2.setNumeroPedido("12346");
pedido2.setCodigoPontoInteresse("1");
pedido2.setDataPedido(date2);
pedido2.setDescricao("Pedido importado pelo Webservice");
pedido2.setEndereco("Rua Governador Joca Pires, 1526, Teresina - Piaui");
pedido2.setHoraPedido("14:32");
pedido2.setHoraEntregalnicial("8:00");
pedido2.setHoraEntregaFinal("18:00");
pedido2.setTempoAtendimento(22);
pedido2.setLoja(loja);
pedido2.setZona(zona);
pedido2.setVendedor(vendedor);
pedido2.setNumeroCarregamento("C1");

PedidoRotaVO pedido3 = new PedidoRotaVO();
pedido3.setNumeroPedido("12347");
pedido3.setCodigoPontoInteresse("1");
pedido3.setDataPedido(date2);
pedido3.setDescricao("Pedido importado pelo Webservice");
pedido3.setEndereco("Rua Governador Joca Pires, 1526, Teresina - Piaui");
pedido3.setHoraPedido("14:32");
pedido3.setHoraEntregalnicial("8:00");
pedido3.setHoraEntregaFinal("18:00");
pedido3.setTempoAtendimento(22);
pedido3.setLoja(loja);
pedido3.setZona(zona);
pedido3.setVendedor(vendedor);
pedido3.setNumeroCarregamento("C1");

ArrayList<PedidoRotaVO> lista = new ArrayList<PedidoRotaVO>();
lista.add(pedido);
lista.add(pedido2);
lista.add(pedido3);

ResultadoImportacaoVO resultado = importadorWS.importarPedidos(lista, "<CNPJ>", "<SENHA1>",
"<SENHA2>");

System.out.println("Sucesso ? " + resultado.isComSucesso());
System.out.println("Mensagem = " + resultado.getMensagem());

} catch (DatatypeConfigurationException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

}

}
```



## Exemplo 2 : Código Java para Consulta

```
package teste;

import concept.gps.wsclient.ImportadorPedidos;
import concept.gps.wsclient.ImportadorPedidosService;
import concept.gps.wsclient.PedidoRotaVO;
import concept.gps.wsclient.ResultadoConsultaStatusVO;

public class TestadorConsulta {

    public static void main(String[] args) {
        try {
            ImportadorPedidos importadorWS = new ImportadorPedidosService().getImportadorPedidosPort();

            PedidoRotaVO pedido = new PedidoRotaVO();
            pedido.setNumeroPedido("12345");

            ResultadoConsultaStatusVO resultado = importadorWS.consultarStatusEntregaPedido(pedido,
"<CNPJ>", "<SENHA1>", "<SENHA2>");
            System.out.println("Sucesso ? " + resultado.isComSucesso());
            System.out.println("Mensagem = " + resultado.getMensagem());
            System.out.println("Status = " + resultado.getItinerarioViagem().getStatus());

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

### Exemplo 3 : Código Java para Roteirizar

```
try {
    AutomatizadorFachada automatizador = new
AutomatizadorFachadaService().getAutomatizadorFachadaPort();

    GregorianCalendar horalnicio = new GregorianCalendar();
    horalnicio.set(Calendar.HOUR_OF_DAY, 15);
    horalnicio.set(Calendar.MINUTE, 54);
    horalnicio.set(Calendar.SECOND, 0);

    XMLGregorianCalendar tsHoralnicio;

    tsHoralnicio = DatatypeFactory.newInstance().newXMLGregorianCalendar(horalnicio);

    List<String> listaCarregamento = new ArrayList<String>();
    listaCarregamento.add("C1");
    listaCarregamento.add("C2");

    String codigoPontoInteresseSaida = "2";
    String codigoPontoInteresseRetorno = "2";
    String placaVeiculo = "PIE-5277";
    String nomeCondutor = "TESTE AUTOMATIZADOR";
    boolean isAgruparPedidosMesmoCliente = true;

    automatizador.roteirizarPedidos(0, listaCarregamento, codigoPontoInteresseSaida,
codigoPontoInteresseRetorno, tsHoralnicio, placaVeiculo, nomeCondutor, isAgruparPedidosMesmoCliente , "<CNPJ>",
"<SENHA1>", "<SENHA2>");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

## Exemplo 4 : Código Java para Importar Despesas

```

public static void testarCadastroDespesas() {
    ImportadorDespesas importadorWS = new ImportadorDespesasService().getImportadorDespesasPort();
    List<DespesaVO> listaDespesas = new ArrayList<DespesaVO>();
    try {
        GregorianCalendar c = new GregorianCalendar();
        c.setTime(new java.util.Date());
        XMLGregorianCalendar date2 = DatatypeFactory.newInstance().newXMLGregorianCalendar(c);

        DespesaVO d1 = new DespesaVO();
        d1.setCondutor("Antônio Francisco");
        d1.setData(date2);
        d1.setDescricao("Despesa importada");
        d1.setHora("08:00");
        d1.setNome("Teste de Importação 1");
        d1.setOdometro(4000);
        d1.setPlacaVeiculo("PIE-5277");
        d1.setQuantidade(new BigDecimal("1"));
        d1.setTipo(TipoDespesa.ABASTECIMENTO_EXTERNO);
        d1.setTipoCombustivel(TipoCombustivel.GASOLINA_ADITIVADA);
        d1.setValor(new BigDecimal("123.45"));

        listaDespesas.add(d1);
        concept.gps.wsclient.despesas.ResultadoImportacaoVO resultado =
importadorWS.importarDespesas(listaDespesas, "<CNPJ>", "<SENHA1>", "<SENHA2>");

        System.out.println("Sucesso ? " + resultado.isComSucesso());
        System.out.println("Mensagem = " + resultado.getMensagem());
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

## Código ANT para gerar Stubs

```
<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>
```