



INFORMAÇÕES AO DESENVOLVEDOR (ERP)



© Lúnet App

FORÇA
DE VENDAS

Versão: **1.04**

Ano: **2016**

Aplicativo: **Força de Vendas**

Elaborada por: **A. Linguanoto**

Consumo de WebServices

Finalidade

A Lúnet Soluções em Tecnologia, desenvolveu este manual técnico com a finalidade de proporcionar aos seus clientes/desenvolvedores (ERP) a praticidade para enviar/receber os dados como itens, preços, estoque, formas de pagamento, transportadoras, clientes e pedidos para os representantes nos dispositivos móveis através de WebServices.

Em caso de dúvidas para executar o consumo do webservice contido neste manual, entre em contato com a Central de Suporte ao Cliente Lúnet pelo telefone (19) 3044-0377 ou através do e-mail suporte@lunet.com.br.

1) Consumindo o Webservice Lúnet para Envio de Dados (enviardados())

Os dados a serem enviados deverão ser gerados nos padrões de arquivo CSV conforme contido no manual “Lúnet-Força de Vendas – Manual de Integração - Informações ao Desenvolvedor - Envio de Dados Lúnet - v1.05.pdf” ou superior.

O retorno do webservice sempre será síncrono, ou seja, os dados serão processados durante a chamada da função e no retorno já será devolvido se todos os registros foram inseridos com sucesso, ou por exemplo, quais registros não foram inseridos devido a alguma informação incorreta ou layout incorreto. Os registros que estiverem dentro das condições necessárias serão executados, ou seja, serão incluídos / alterados independente se existir erros em outros registros do arquivo. Resumo dos retornos:

- ERRO: <ERRO> + chr(13) + <DADOS DA LINHA ENVIADA>
- OK

Dados para Consumo:

1) Location:

A) Para consumo sem WSDL

[http://177.66.168.159/funcoes/func_android/webservice_lunetremessa.php?EMP_CGCCPF=\\$cgc&EMP_TOKEN=\\$token](http://177.66.168.159/funcoes/func_android/webservice_lunetremessa.php?EMP_CGCCPF=$cgc&EMP_TOKEN=$token)

Onde:

\$cgc → é o CNPJ ou CPF do cliente registrado em nossos bancos de dados.

\$token → é o token (chave public) do cliente registrado em nossos bancos de dados equivalente ao CNPJ/CPF informado em \$cgc.

B) Para consumo com WSDL

http://177.66.168.159/funcoes/func_android/webservice_lunetremessawsdl.php



2) Uri:

http://177.66.168.159/funcoes/func_android

3) Método

enviardados() → \$webservice->enviardados(\$cgc, \$token, \$dados)

Exemplo da chamada final em PHP:

```
$result = $webservice->enviardados($EMP_CGCPF, $EMP_TOKEN, $DADOS);
```

Onde:

\$cgc e \$token → devem ser enviados conforme mencionado no tópico 1) Location.

\$dados → deverá ser enviado os dados CSV separados por chr(13) e pelo literal "
" no final de cada linha conforme manual de geração dos dados. Poderá ser enviado em \$dados quantos registros precisar como o exemplo abaixo, porém evite arquivos muito grandes para evitar que o processamento do método demore muito. Abaixo tem algumas sugestões para melhorar este processamento.

\$DADOS =

```
1.00;REPRESENTANTE;XX;J;13394372000191;ISENTO;LUNET SOLUÇÕES EM TECNOLOGIA LTDA;LUNET;SP;I;AUTO;TESTE;ADM;ADM;<br>1.00;TRANSPORTADORA;XX;F;28588664810;ISENTO;ALESSANDRO LINGUANOTO;ALINGUANOTO;A;;<br>1.00;FORMA DE PAGAMENTO;XX;30DDL TESTE;2;10.1;A;;<br>1.00;GRUPO DE CLIENTE;XX;CONSUMIDOR FINAL;ADM;A;;<br>1.00;GRUPO DE CLIENTE;X1;DISTRIBUIDORES;ADM;A;;<br>
```

Sugestões:

1) Programe uma tarefa para enviar durante todo o dia (a cada 15 minutos por exemplo) os dados de alterações cadastrais que foram realizados dentro do mesmo dia, como por exemplo dados de itens e cadastros agregados, clientes e cadastros agregados, representantes, transportadoras, condições de pagamento, tipos de frete, tipos de venda e status do pedido.

2) Programe uma segunda tarefa para enviar todo dia durante a madrugada (1 vez ao dia), novamente os mesmos dados acima equivalente as alterações cadastrais que foram realizadas, porém dos últimos dois ou três dias. Com isso, alterações do dia serão enviadas rapidamente e caso exista alguma falha, nesta segunda tarefa, a mesma enviará novamente os dados não enviados.

3) Se quiser ainda, programe uma terceira tarefa para enviar todo final de semana (1 vez na semana, aos domingos por exemplo), novamente os mesmos dados acima equivalente ao cadastro completo, ou seja, independente da data da inclusão e/ou alteração do registro.

4) Para visualizar a estrutura do WSDL utilize o link abaixo

http://177.66.168.159/funcoes/func_android/webservice_lunetremessawSDL.php?wsdl

2) Consumindo o Webservice Lúnet para Recebimento de Dados (receberdados())

Os dados enviados serão gerados nos padrões de arquivo CSV conforme contido no manual “Lúnet-Força de Vendas – Manual de Integração - Informações ao Desenvolvedor - Recebimento Dados Lúnet - v1.02.pdf” ou superior.

O retorno do webservice sempre será síncrono, ou seja, os dados serão processados durante a chamada da função e no retorno já será devolvido os registros solicitados no formato do arquivo. Se nenhum registro for localizado será retornado o conteúdo “EMPTY”. Se ocorrer algum erro na chamada do webservice será retornado “ERRO LUNET: XXX”, onde XXX será o erro ocorrido como por exemplo solicitação de uma versão de layout inexistente. Resumo dos retornos:

- EMPTY
- ERRO LUNET: <ERRO>
- <DADOS>

No final de cada linha sempre terá o delimitador “enter” (chr(10)). Todos os registros solicitados serão enviados em um único retorno, porém evite receber dados de períodos muito extensos para evitar que o processamento do método demore muito. Abaixo tem algumas sugestões para melhorar este processamento.

Dados para Consumo:

1) Location:

A) Para consumo sem WSDL

`http://177.66.168.159/funcoes/func_android/webservice_lunetretorno.php?EMP_CGCCPF=$cgc&EMP_TO KEN=$token`

Onde:

\$cgc → é o CNPJ ou CPF do cliente registrado em nossos bancos de dados.

\$token → é o token (chave public) do cliente registrado em nossos bancos de dados equivalente ao CNPJ/CPF informado em \$cgc.

B) Para consumo com WSDL

`http://177.66.168.159/funcoes/func_android/webservice_lunetretornowsdl.php`

2) Uri:

`http://177.66.168.159/funcoes/func_android`

3) Método



receberdados() → \$webservice->receberdados(\$cgc, \$token, \$versao_layout, \$tipo_retorno, \$periodoInicial, \$periodo_final, \$marcar_comosincronizado)

Exemplo A) da chamada final em PHP dos dados aguardando sincronização com o ERP mantendo o registro pendente para que seja possível recebe-lo novamente em uma próxima chamada caso não foi enviado o método setdadossincronizado()

```
$result = $webservice->receberdados($EMP_CGCCPF, $EMP_TOKEN, "1.00", "AGUARDANDO", "", "", "N");
```

Exemplo B) da chamada final em PHP por PERIODO independente de estar aguardando ou não sincronização com o ERP.

```
$result = $webservice->receberdados($EMP_CGCCPF, $EMP_TOKEN, "1.00", "PERIODO", "01/04/2014", "10/04/2014", "S");
```

Exemplo C) da chamada final em PHP dos dados aguardando sincronização com o ERP já marcando o registro como sincronizado no próprio recebimento (DEPRECATED).

```
$result = $webservice->receberdados($EMP_CGCCPF, $EMP_TOKEN, "1.00", "AGUARDANDO", "", "", "S");
```

Onde:

\$cgc e \$token → devem ser enviados conforme mencionado no tópico 1) Location.

\$versao_layout → parâmetro string que deverá ser enviado conforme layout csv (versão atual "1.02").

\$tipo_retorno → poderá ser uma das opções abaixo:

"CLIENTE,AGUARDANDO" → Nesta situação serão retornados todos os clientes que foram incluídos ou alterados desde a última solicitação do método, ou seja, todos os clientes que estão aguardando para serem sincronizados com o ERP.

"PEDIDO,AGUARDANDO" → Nesta situação serão retornados todos os pedidos de venda/orçamentos que foram incluídos desde a última solicitação do método, ou seja, todos os pedidos de venda/orçamentos que estão aguardando para serem sincronizados com o ERP.

“AGUARDANDO” → Nesta situação serão retornados todos os pedidos de venda/orçamentos que foram incluídos desde a última solicitação do método e também todos os clientes que foram incluídos ou alterados desde a última solicitação do método, ou seja, todos os pedidos de venda/orçamentos/clientes que estão aguardando para serem sincronizados com o ERP.

“CLIENTE,PERIODO” → Nesta situação serão retornados todos os clientes que foram incluídos ou alterados dentro do período mencionado nos dois próximos parâmetros, ou seja, independente se o cliente estava aguardando para ser sincronizado com o ERP.

“PEDIDO,PERIODO” → Nesta situação serão retornados todos os pedidos de vendas/orçamentos que foram incluídos dentro do período mencionado nos dois próximos parâmetros, ou seja, independente se o cliente estava aguardando para ser sincronizado com o ERP.

“PERIODO” → Nesta situação serão retornados todos os pedidos de venda/orçamentos que foram incluídos e também todos os clientes que foram incluídos ou alterados dentro do período mencionado nos dois próximos parâmetros, ou seja, independente se o pedido de venda/orçamento/cliente estava aguardando para ser sincronizado com o ERP.

“CLIENTE,IDLUNET” → Nesta situação será retornado apenas o cliente que possuir o ID (Nosso Número de Pedido de Venda/Orçamento) informado no parâmetro ID, ou seja, independente se o cliente estava aguardando para ser sincronizado com o ERP.

“PEDIDO,IDLUNET” → Nesta situação será retornado apenas o pedido que possuir o ID (Nosso Número de Pedido de Venda/Orçamento) informado no parâmetro ID, ou seja, independente se o pedido estava aguardando para ser sincronizado com o ERP.

“CLIENTE,IDIINTEGRACAO” → Nesta situação será retornado apenas o cliente que possuir o ID (Seu Número de Pedido de Venda/Orçamento) informado no parâmetro ID, ou seja, independente se o cliente estava aguardando para ser sincronizado com o ERP.

“PEDIDO,IDIINTEGRACAO” → Nesta situação será retornado apenas o pedido que possuir o ID (Seu Número de Pedido de Venda/Orçamento) informado no parâmetro ID, ou seja, independente se o pedido estava aguardando para ser sincronizado com o ERP.

\$periodoInicial → data inicial no formato DD/MM/AAAA que será utilizada nos filtros quando utilizada uma dentre as opções de período. Caso você estiver utilizando a opção de aguardando, passe o conteúdo "" (branco) para este parâmetro.

\$periodo_final → data final no formato DD/MM/AAAA que será utilizada nos filtros quando utilizada uma dentre as opções de período. Caso você estiver utilizando a opção de aguardando, passe o conteúdo "" (branco) para este parâmetro.

\$marcar_comosincronizado → informa ao webservice se os registros disponibilizados no recebimento dos dados já deve ou não ser marcados (atualizados) como já sincronizados. Conteúdos Validos: “S” para marcar o registro como sincronizado ou “N” para manter o registro com o status atual. Quando utilizado “N” (recomendado), deve ser chamado no final do processamento o método setdadossincronizado().

\$sid → número do pedido ou do cliente que será utilizado nos filtros quando utilizada uma dentre as opções de pesquisa por ID. Caso você estiver utilizando a opção de pesquisa por ID, passe o conteúdo "" (branco) para este parâmetro.

Sugestões:

1) Programe uma tarefa para receber os dados de todos os registros que estão aguardando sincronização com o ERP “AGUARDANDO” durante todo o dia (a cada 5 minutos por exemplo)

2) Somente se necessário, programe uma tarefa para receber os dados de registros de um determinado período (“CLIENTE,PERIODO”, “PEDIDO,PERIODO” ou “PERIODO”). Esta situação deverá ser utilizada por exemplo uma vez ao dia apenas para verificar se realmente todos os pedidos de venda/orçamentos/clientes foram processados.

3) Para visualizar a estrutura do WSDL utilize o link abaixo

http://177.66.168.159/funcoes/func_android/webservice_lunetretornowsdl.php?wsdl

3) Consumindo o WebService Lúnet para Recebimento de Dados (Processamento OK) (setdadossincronizado())

O objetivo deste método é marcar os registros como sincronizado com ERP. Este webservice/método deverá ser utilizado após inserir/atualizar o registro na base de dados do ERP. Exemplo, a cada uma hora você consome o webservice método receberdados("AGUARDANDO") com o parâmetro marcar como aguardando “N”. Sempre o webservice devolverá TODOS os registros configurados como não sincronizados quando solicitado através do parâmetro “AGUARDANDO, sendo assim para marcar o registro como sincronizado deve ser utilizado o método setdadossincronizado(). Com isto este registro não será mais. Neste novo método é possível marcar clientes, endereços e pedidos como sincronizado.

Os dados a serem enviados deverão ser gerados nos padrões de arquivo CSV conforme contido no manual “Lúnet-Força de Vendas – Manual de Integração - Informações ao Desenvolvedor - Recebimento de Dados Lúnet - v1.02.pdf” ou superior, registros “CLIENTE SINCRONIZADO”, “ENDERECOS ADICIONAIS SINCRONIZADO” e “PEDIDO SINCRONIZADO”.

O retorno do webservice sempre será síncrono, ou seja, os dados serão processados durante a chamada da função e no retorno já será devolvido se todos os registros foram atualizados com sucesso, ou por exemplo, quais registros não foram atualizados devido a alguma informação incorreta ou layout incorreto. Os

registros que estiverem dentro das condições necessárias serão executados, ou seja, serão alterados independente se existir erros em outros registros do arquivo. Resumo dos retornos:

- ERRO: <ERRO> + chr(13) + <DADOS DA LINHA ENVIADA>
- OK

Dados para Consumo:

1) Location:

A) Para consumo sem WSDL

[http://177.66.168.159/funcoes/func_android/webservice_lunetretorno.php?EMP_CGCCPF=\\$cgc&EMP_TOKEN=\\$token](http://177.66.168.159/funcoes/func_android/webservice_lunetretorno.php?EMP_CGCCPF=$cgc&EMP_TOKEN=$token)

Onde:

\$cgc → é o CNPJ ou CPF do cliente registrado em nossos bancos de dados.

\$token → é o token (chave public) do cliente registrado em nossos bancos de dados equivalente ao CNPJ/CPF informado em \$cgc.

B) Para consumo com WSDL

http://177.66.168.159/funcoes/func_android/webservice_lunetretornowsdl.php

2) Uri:

http://177.66.168.159/funcoes/func_android

3) Método

setdadossincronizado() → \$webservice->setdadossincronizado(\$cgc, \$token, \$dados)

Exemplo da chamada final em PHP:

```
$result = $webservice->setdadossincronizado($EMP_CGCCPF, $EMP_TOKEN, $DADOS);
```

Onde:

\$cgc e \$token → devem ser enviados conforme mencionado no tópico 1) Location.

\$dados → deverá ser enviado os dados CSV separados por chr(13) e pelo literal "
" no final de cada linha conforme manual de geração dos dados. Poderá ser enviado em \$dados quantos registros precisar como o exemplo abaixo, porém evite arquivos muito grandes para evitar que o processamento do método demore muito. Abaixo tem algumas sugestões para melhorar este processamento.

SDADOS =

```
1.02;CLIENTE SINCRONIZADO;5_1308_0001;0001108;N;J;13394372000191;<br>
1.02;CLIENTE SINCRONIZADO;5_1308_0002;0001109;N;J;04734186000163;<br>
1.02;ENDERECOS ADICIONAIS SINCRONIZADO;5_1308_0001;0001108;J;13394372000191;COMERCIAL;<br>
1.02;ENDERECOS ADICIONAIS SINCRONIZADO;5_1308_0001;0001109;J;04734186000163;COMERCIAL;<br>
1.02;PEDIDO SINCRONIZADO;04734186000163;5_1308_0010;<br>
1.02;PEDIDO SINCRONIZADO;04734186000163;5_1308_0011;<br>
```

4) Consumindo o Webservice Lúnet para Recebimento de Dados (Tratamento da Exceção) (setdadosaguardando()) - DEPRECATED

O objetivo deste método é marcar os registros como aguardando sincronização com ERP “novamente”. Este webservice/método deverá ser utilizado quando ocorrer algum problema no insert ou update ou for encontrada alguma inconsistência para poder inserir/atualizar o registro na base de dados do ERP. Exemplo, a cada uma hora você consome o webservice método receberdados("AGUARDANDO"). Sempre o webservice devolverá os registros que ainda não foram enviados quando solicitado através do parâmetro “AGUARDANDO” e assim que este registro é disponibilizado no retorno do webservice este é marcado para não ser mais disponibilizado (com exceção se chamar a função por “PERIODO” ou derivadas destes). Mas caso ocorrer algum problema no insert, update ou validação no ERP, o pedido não será inserido e também não será mais enviado, com isso ficará inconsistente a integração e um pedido (ou qualquer outro tipo de registro) ficará "PERDIDO" sem ser integrado. Neste novo método é possível marcar clientes, endereços e pedidos como aguardando sincronização "novamente".

Os dados a serem enviados deverão ser gerados nos padrões de arquivo CSV conforme contido no manual “Lúnet-Força de Vendas – Manual de Integração - Informações ao Desenvolvedor - Recebimento de Dados Lúnet - v1.02.pdf” ou superior, registros “CLIENTE AGUARDANDO”, “ENDERECOS ADICIONAIS AGUARDANDO” e “PEDIDO AGUARDANDO”.

O retorno do webservice sempre será síncrono, ou seja, os dados serão processados durante a chamada da função e no retorno já será devolvido se todos os registros foram atualizados com sucesso, ou por exemplo, quais registros não foram atualizados devido a alguma informação incorreta ou layout incorreto. Os registros que estiverem dentro das condições necessárias serão executados, ou seja, serão alterados independente se existir erros em outros registros do arquivo. Resumo dos retornos:

- ERRO: <ERRO> + chr(13) + <DADOS DA LINHA ENVIADA>
- OK

Dados para Consumo:

1) Location:

A) Para consumo com WSDL

[http://177.66.168.159/funcoes/func_android/webservice_lunetretorno.php?EMP_CGCCPF=\\$cgc&EMP_TO KEN=\\$token](http://177.66.168.159/funcoes/func_android/webservice_lunetretorno.php?EMP_CGCCPF=$cgc&EMP_TO KEN=$token)

Onde:

\$cgc → é o CNPJ ou CPF do cliente registrado em nossos bancos de dados.

\$token → é o token (chave public) do cliente registrado em nossos bancos de dados equivalente ao CNPJ/CPF informado em \$cgc.

B) Para consumo com WSDL

http://177.66.168.159/funcoes/func_android/webservice_lunetretornowsdl.php

2) Uri:

http://177.66.168.159/funcoes/func_android

3) Método

setdadosaguardando() → \$webservice-> **setdadosaguardando** (\$cgc, \$token, \$dados)

Exemplo da chamada final em PHP:

```
$result = $webservice->setdadosaguardando($EMP_CGCCPF, $EMP_TOKEN, $DADOS);
```

Onde:

\$cgc e \$token → devem ser enviados conforme mencionado no tópico 1) Location.

\$dados → deverá ser enviado os dados CSV separados por chr(13) e pelo literal “
” no final de cada linha conforme manual de geração dos dados. Poderá ser enviado em \$dados quantos registros precisar como o exemplo abaixo, porém evite arquivos muito grandes para evitar que o processamento do método demore muito. Abaixo tem algumas sugestões para melhorar este processamento.

\$DADOS =

```
1.02;CLIENTE AGUARDANDO;5_1308_0001;0001108;N;J;13394372000191;<br>
1.02;CLIENTE AGUARDANDO;5_1308_0002;0001109;N;J;04734186000163;<br>
1.02;ENDERECOS ADICIONAIS AGUARDANDO;5_1308_0001;0001108;J;13394372000191;COMERCIAL;<br>
1.02;ENDERECOS ADICIONAIS AGUARDANDO;5_1308_0001;0001109;J;04734186000163;COMERCIAL;<br>
1.02;PEDIDO AGUARDANDO;04734186000163;5_1308_0010;<br>
1.02;PEDIDO AGUARDANDO;04734186000163;5_1308_0011;<br>
```

Sugestões:

1) Programe na confirmação do recebimento dos dados para marcar os registros que foram realmente gravados no banco de dados como “sincronizados” com isso na próxima solicitação ao webservice de recebimento “receberdados()” o registro não será enviado novamente.

DICA:

Você também poderá utilizar, tanto para envio quanto para recebimento de dados, os **Servidores Alternativos**, ou seja, ao invés de consumir os webservices através do endereço **177.66.168.159** consuma os webservices através dos endereços www.forcadevendasandroid.com.br,



www.forcadevendasmovel.com.br ou www.lunet.com.br. Todos os links acessam o mesmo banco de dados em (177.66.168.159), ou seja, não existe a possibilidade de o banco de dados ficar sem integridade ou não sincronizado, pois as informações sempre serão coletadas de um único banco de dados. Estes 3 links, são links alternativos caso o IP 177.66.168.159 venha ser alterado (muito pouco provável) ai você poderá utilizar um dos domínios alternativos sem a necessidade de alteração imediata de sua aplicação, porém ele é mais lento devido estar hospedado em um servidor/provedor diferente do servidor/provedor onde está hospedado os bancos de dados.

INETWEB (SERVIDOR DEDICADO)

177.66.168.159

server1.forcadevendasandroid.com.br

server1.forcadevendasmovel.com.br

server1.lunet.com.br

IPCOMPANY (SERVIDOR COMPARTILHADO)

www.forcadevendasandroid.com.br

UOLHOST (SERVIDOR COMPARTILHADO)

www.forcadevendasmovel.com.br

www.lunet.com.br

Mudanças do layout 1.03 para o Layout 1.04

Registros Incluídos na Versão 1.04

- Nenhum

Registros Alterados

- Implementado alterações equivalente aos links
- Implementado explicação para utilização via WSDL
- Alterado servidor padrão de consumo de www.lunet.com.br (Servidor Compartilhado) para 177.66.168.159 (Novo Servidor Lúnet Dedicado)

Outras Alterações

-

Mudanças do layout 1.02 para o Layout 1.03

Registros Incluídos na Versão 1.03

- Novo método no webservice de recebimento de dados setdadosincronizado () → **Consumindo o Webservice Lúnet para Recebimento de Dados (Processamento OK)**

Registros Alterados

- setdadosaguardando() – DEPRECATED

Outras Alterações

- Nenhum

Mudanças do layout 1.01 para o Layout 1.02

Registros Incluídos na Versão 1.02

- Novo método no webservice de recebimento de dados setdadosaguardando() → **Consumindo o Webservice Lúnet para Recebimento de Dados (Tratamento da Exceção)**

Registros Alterados

- Adicionado novo parâmetro MARCAR_COMOSINCRONIZADO no método receberdados()

Outras Alterações

- Adicionado explicações adicionais para facilitar os analistas/programadores.

Mudanças do layout 1.00 para o Layout 1.01

Registros Incluídos na Versão 1.01

- Consumindo o Webservice Lúnet para Recebimento de Dados

Registros Alterados

- Foi alterado o **link** referente ao consumo de Webservice para Envio de Dados.
Foi alterado o **nome do método** referente ao consumo de Webservice para Envio de Dados.

Outras Alterações

- Nenhum