



---

TANGRAM - Interface de SMS-MT

1	TANGRAM – INTERFACE DE SMS-MT .....	3
1.1	OBJETIVO .....	3
1.2	REGISTRO DE ALTERAÇÕES .....	3
1.3	INTERFACES DE COMUNICAÇÃO.....	3
1.3.1	Interface HTTP .....	4
1.3.2	Interface FTP.....	4
1.3.3	Documento XML de Pedido .....	5
1.3.3.1	DTD .....	5
1.3.3.2	XML de exemplo.....	6
1.3.3.3	Pedido HTTP/GET de exemplo .....	7
1.3.3.4	Descrição dos elementos e atributos .....	7
1.3.4	Documento XML de Resposta .....	10
1.3.4.1	DTD .....	10
1.3.4.2	XML de exemplo.....	11
1.3.4.3	Descrição dos elementos e atributos .....	12
1.3.4.4	Descrição dos códigos de erro .....	13
1.4	INTERFACE SMS.....	14
1.4.1	SMPP .....	14
1.5	REGRAS PARA TARIFICAÇÃO .....	15
2	ANEXO 1 - MAPEAMENTO COM O PROTOCOLO SMPP .....	16
3	ANEXO 2 - EXEMPLOS EM ASP .....	18

# 1 TANGRAM – Interface de SMS–MT

## 1.1 Objetivo

Definição dos formatos de pedidos para envio de mensagens SMS através da plataforma TANGRAM. Atualmente apenas são suportados pedidos feito via documento XML (utilizando HTTP/POST, COM+, FTP, etc) e pedidos feitos via parâmetros HTTP/GET. Todos os formatos receberão como resposta um documento XML.

## 1.2 Registro de Alterações

Data	Versão	Descrição
25/11/2004	1.01.000	Adicionado o atributo <b>udh</b> ao elemento <b>text</b> , para permitir o envio de múltiplos MTs com conteúdo no campo UserDataHeader do SMS, através de um mesmo pedido ao Tangram.
05/01/2005	1.02.000	Adicionado um apêndice para mapeamento dos campos do XML do Tangram com os parâmetros do protocolo SMPP.
01/03/2005	1.02.001	Movido o tópico de exemplos ASP para um novo anexo ao final do documento.
15/03/2005	1.03.000	Incluída a informação a respeito da interface SMPP.
04/04/2005	1.04.000	Incluído o atributo <b>interface</b> ao pacote, para permitir que a aplicação informe a origem do pedido do item a ser tarifado. Incluída a descrição da integração de cobrança através da interface de envio de MT.
06/04/2005	1.05.000	Alterados os valores do atributo <b>interface</b> do elemento <b>package</b> . Inseridos os atributos <b>copyright</b> e <b>external_id</b> ao elemento <b>package</b> e o elemento <b>app_request_id</b> .
18/04/2005	1.06.000	Inclusão de um novo tipo de notificação, utilizada para eventos de cobrança relacionadas à mensagem ou pacote enviado. Removido o atributo <b>code</b> do elemento <b>package</b> .
12/05/2005	1.07.000	Correção no DTD de pedido para inclusão do campo <b>app_request_id</b> .
07/07/2005	1.08.000	Inclusão do atributo <b>method</b> no XML de pedido
16/02/2007	1.09.000	Detalhamento das interfaces HTTP e FTP.

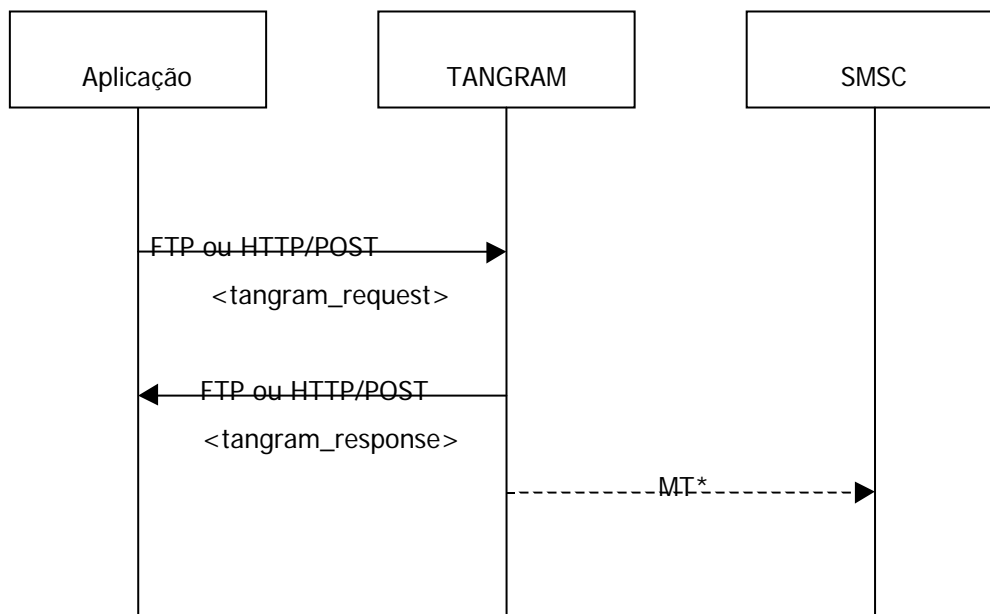
## 1.3 Interfaces de comunicação

Sempre que a Aplicação desejar enviar uma mensagem terminada (MT), ela deverá realizar um pedido HTTP/POST ou HTTP/GET direcionado à URL do TANGRAM ou efetuar um FTP em seu diretório próprio,

contendo um documento XML como corpo da mensagem e identificando-se através de um usuário e senha. O TANGRAM receberá este pedido e responderá à Aplicação através de um documento XML que conterá o resultado da operação (a mensagem pode ser aceita ou rejeitada).

Todos os campos existentes no XML ou no pedido GET apenas poderão ser informados pela Aplicação caso a configuração do Serviço e/ou Canal da Aplicação no TANGRAM permita a sobrescrição de tais campos pelo documento XML recebido. Caso a Aplicação tente sobrescrever um parâmetro para o qual não tem permissões de escrita, a mensagem será recusada.

Fluxograma:



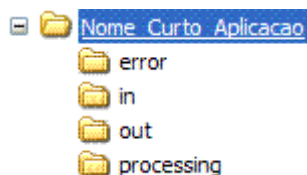
\* Apenas no caso da MT ter sido aceita

### 1.3.1 Interface HTTP

Para o caso da interface HTTP o pedido deve ser encaminhado para o arquivo *tangram.asp* que fica na pasta *scripts*. Assim, a chamada deve ser feita em: <http://IP/scripts/tangram.asp>, onde IP é o site onde está instalado o Tangram.

### 1.3.2 Interface FTP

No caso da interface FTP, na raiz do site do Tangram existe uma pasta *apps* onde são criadas as pastas de cada uma das Aplicações de acordo com seu nome curto. Em cada pasta da operadora existe uma estrutura como a ilustrada a seguir:



Assim, o arquivo deve ser colocado na pasta *in* da aplicação. O Tangram mantém um serviço que coleta cada novo arquivo que é criado nesse diretório. Durante o processamento esse arquivo é salvo temporariamente na pasta *processing*. Após esse processamento, caso haja sucesso, é criado o arquivo de

resposta na pasta *out*. Excepcionalmente, no caso de erros no processamento, o arquivo de entrada é gravado na pasta *error*.

### 1.3.3 Documento XML de Pedido

#### 1.3.3.1 DTD

```
<!ELEMENT tangram_request (send)>
<!ATTLIST tangram_request
    company_id CDATA #REQUIRED
    service_id CDATA #REQUIRED
>

<!ELEMENT send (source?, destination*, channel_id?, text*, user_data_header?,
validity?, schedule?, notification?, package?, retries?, service_type?,
mo_message_id?, app_specific?, app_request_id?, request_datetime)>
<!ATTLIST send
    keep_session CDATA #IMPLIED
>

<!ELEMENT source (#PCDATA)>
<!ELEMENT destination (#PCDATA)>
<!ELEMENT channel_id (#PCDATA)>
<!ELEMENT text (#PCDATA)>
<!ATTLIST text
    binary CDATA #IMPLIED
    udh CDATA #IMPLIED
    method CDATA #IMPLIED
>

<!ELEMENT user_data_header (#PCDATA)>
<!ELEMENT validity (#PCDATA)>
<!ATTLIST validity
    relative CDATA #IMPLIED
>

<!ELEMENT schedule (#PCDATA)>
<!ATTLIST schedule
    relative CDATA #IMPLIED
>

<!ELEMENT notification (#PCDATA)>
<!ATTLIST notification
    type CDATA #IMPLIED
    calltype CDATA #IMPLIED
>
```

```
<!ELEMENT package EMPTY>
<!ATTLIST package
    external_id CDATA #IMPLIED
    name CDATA #IMPLIED
    description CDATA #IMPLIED
    interface CDATA #IMPLIED
    owner_ctn CDATA #IMPLIED
    copyright CDATA #IMPLIED
>
<!ELEMENT retries EMPTY>
<!ATTLIST retries
    max CDATA #REQUIRED
    interval CDATA #REQUIRED
>
<!ELEMENT service_type (#PCDATA)>
<!ELEMENT mo_message_id (#PCDATA)>
<!ELEMENT app_specific (#PCDATA)>
<!ELEMENT app_request_id (#PCDATA)>
<!ELEMENT request_datetime (#PCDATA)>
```

### 1.3.3.2 XML de exemplo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tangram_request company_id="12" service_id="2">
    <send keep_session="false">
        <source>500</source>
        <destination>3191234567</destination>
        <destination>3192345678</destination>
        <destination>3193456789</destination>
        <channel_id>01</channel_id>
        <text binary="false" udh="" method="">Mensagem de teste 1/2</text>
        <text binary="false" udh="01234567" method="truncate">Mensagem de
            teste 2/2</text>
        <user_data_header></user_data_header>
        <validity relative="true">0000000010</validity>
        <schedule relative="true">0000000005</schedule>
        <notification type="5" calltype="0">http://mo/b.cgi</notification>
        <package external_id="" name="" description="" interface=""
            owner_ctn="" copyright=""/>
    <retries max="3" interval="10"/>
</tangram_request>
```

```

<service_type></service_type>
<mo_message_id>A434FD01</mo_message_id>
<app_specific>Mensagem gratuita</app_specific>
<app_request_id>02038834</app_request_id>
<request_datetime></request_datetime>

</send>
</tangram_request>

```

### 1.3.3.3 Pedido HTTP/GET de exemplo

Os campos informados através do pedido HTTP/GET são os mesmos dos campos informados no XML, através do HTTP/POST. Entretanto, os nomes dos campos são definidos de acordo com as seguintes regras:

1. Caso o campo seja um elemento, seu nome será o nome do elemento. Exemplo:  
<text>Texto de exemplo</text> -> teste=Texto+de+exemplo
2. Caso o campo seja um atributo, seu nome será o nome do elemento pai, mais o caracter "@", mais o nome do atributo. Exemplo:  
<retries max="3"/> -> retries@max=3

Veja um exemplo a seguir:

```

http://tangram/scripts/tangram.asp?tangram_request@company_id=12&tangram_request
@service_id=2&source=500&destination=3191234567&destination=3192345678&destinati
on=3193456789&channel_id=01&text=Mensagem+de+teste+1/2&text=Mensagem+de+teste+2/
2&validity@relative=true&validity=0000000010&schedule@relative=true&schedule=000
0000005&notification@type=5&notification@calltype=0&notification=http://mo/b.cgi
&package@name=&package@description=&package@owner_ctn=&retries@max=3&retries@int
erval=10&mo_message_id=A434FD01&app_specific=Mensagem+gratuita&request_datetime=

```

### 1.3.3.4 Descrição dos elementos e atributos

Elemento	Atributo	Tipo/Valores	Obrig.	Descrição
tangram_request	company_id	Inteiro	S	Número inteiro identificador da Empresa
tangram_request	service_id	Inteiro	S	Número inteiro identificador do Serviço
send	keep_session	"false"/"true"/""	N	<p>Determina se a mensagem a ser enviada iniciará uma sessão MO entre a aplicação e o(s) destinatário(s). Os possíveis valores são:</p> <ul style="list-style-type: none"> <li>• "false" – termina uma sessão previamente estabelecida. Caso o destinatário não possua sessão com a aplicação, a mensagem será recusada.</li> <li>• "true" – inicia a sessão, somente se o destinatário não possuir sessão com outra aplicação. Caso contrário, a mensagem será recusada. Caso o destinatário já esteja em sessão com a aplicação, a sessão anterior é terminada e uma nova sessão é criada para a mensagem atual.</li> <li>• não informado – não altera o estado da sessão do destinatário (valor default)</li> </ul>

source		Texto	N	Originador da mensagem.
destination		DDNNNNNNNN	N	Destinatário(s) da mensagem.
channel_id		Inteiro	S, Apenas se serviço possuir mais de um Canal	Número inteiro identificador do Canal do Serviço. Caso o <b>destination</b> não seja informado, todos os clientes provisionados no Canal serão destinatários da mensagem. Se um ou mais <b>destination</b> forem informados, os mesmos deverão estar provisionados no Canal <b>channel_id</b> e serão os únicos destinatários da mensagem.
text		Texto	N	Texto(s) da mensagem a ser enviada
text	binary	"false"/"true"	N	Indica se o conteúdo do texto recebido deve ser enviado em forma textual ou binária. No caso de mensagens binárias, o texto deve estar codificado em caracteres hexadecimais. Por exemplo, para que sejam enviados os bytes 0x38, 0x76, 0x88, o elemento text deve estar na forma: <text binary="true">387688</text>
text	udh	Texto	N	Conteúdo que deve ser enviado como cabeçalho das mensagens. Refere-se ao campo <i>UserDataHeader</i> de um SMS.
text	method	""/"truncate"	N	Indica como o texto deve ser tratado. Caso informado o valor "truncate", o texto será cortado no limite de caracteres aceitos pela plataforma da operadora.
user_data_header		Texto	N	* Obsoleto. Utilizado para manter a compatibilidade com versões anteriores
validity	relative	"false"/"true"	N	Indica se a validade é relativa (=true) ou absoluta (=false – valor default). Validade relativa indica a quantidade de tempo máxima para entrega da mensagem antes que ela seja descartada. Validade absoluta indica a data e hora para expiração da mensagem.
validity		DDMMYYHHNNSSZZ	N	Data e hora de validade da mensagem
schedule	relative	"false"/"true"	N	Indica se o agendamento é relativo (=true) ou absoluto (=false – valor default). Agendamento relativo indica a quantidade de tempo que deverá ser aguardada para que a mensagem seja entregue. Agendamento absoluto indica a data e hora para envio da mensagem
schedule		DDMMYYHHNNSSZZ	N	Tempo a ser aguardado para entrega da mensagem ou data e hora para envio programado da mensagem.



notification	type	Inteiro	N	<p>Tipo da notificação requisitada. Os valores possíveis serão inferiores a 32, sendo calculados baseados nos bits abaixo:</p> <p>bit 0 = Sucesso na entrega ao SMSC</p> <p>bit 1 = Falha de entrega ao SMSC</p> <p>bit 2 = Sucesso na entrega ao Celular</p> <p>bit 3 = Falha na entrega ao Celular</p> <p>bit 4 = Notificação de cobrança</p> <p>bit 5 = Notificação apenas para registro no Tangram</p> <p>Por exemplo:</p> <ul style="list-style-type: none"> <li>• para notificação de todos os eventos de sucesso: 0101<sub>bin</sub> - valor do <b>type</b> = 5.</li> <li>• para notificação de todos os eventos de falha: 1010<sub>bin</sub> - valor do <b>type</b> = 10.</li> <li>• para notificação de todos os eventos do SMSC: 0011<sub>bin</sub> - valor do <b>type</b> = 3.</li> <li>• para notificação de todos os eventos do Celular: 1100<sub>bin</sub> - valor do <b>type</b> = 12.</li> <li>• Para notificação de todos os eventos de cobrança: 10000<sub>bin</sub> - valor do <b>type</b> = 16.</li> <li>• para notificação de todos os eventos: 11111<sub>bin</sub> - valor do <b>type</b> = 31.</li> <li>• caso seja setado o bit 5 (<b>type</b> &gt; 32), a notificação não será entregue à aplicação, sendo apenas registrada no banco do Tangram</li> </ul>
notification	calltype	0..4	N	<p>Informa o tipo de entrega da notificação. Os valores possíveis são:</p> <ol style="list-style-type: none"> <li>0. HTTP GET</li> <li>1. HTTP POST/XML</li> <li>2. SOAP</li> <li>3. COM+</li> </ol>
notification		Texto	N	<p>Informa o local de entrega da notificação. Os valores informados dependem do atributo <b>call_type</b>, e deverão ser:</p> <ol style="list-style-type: none"> <li>0. URL</li> <li>1. URL</li> <li>2. URL</li> <li>3. PID do componente</li> </ol>
package	external_id	Texto	N	<p>Identificador do pacote para a aplicação. Pode ser utilizado para posterior batimento entre os dados registrados no Tangram e os dados registrados na aplicação.</p>
package	name	Texto	N	<p>Nome curto do pacote.</p>
package	description	Texto	N	<p>Descrição (ou nome longo) do pacote.</p>

package	interface	Inteiro	N	Interface utilizada pelo cliente para realização do pedido. Valores possíveis: 0. Desconhecido (default) 1. WEB 2. SMS-MO 3. WAP 4. Portal de Voz ou URA 5. SIM card 6. MMS 7. MExE (J2ME, Synbian, BREW, etc) 8. Call Center
package	owner_ctn	DDNNNNNNNN	N	Dono do pacote. Caso não seja informado, é implícito que o dono do pacote é o destinatário da mensagem.
package	copyright	Texto	N	Texto livre que possibilita a disponibilização de informações de copyright do pacote enviado.
retries	max	inteiro	N	Número máximo de novas tentativas de entrega da mensagem.
retries	interval	Inteiro	N	Número de minutos entre as tentativas de entrega da mensagem
service_type		Texto	N	
mo_message_id		Texto	N	Identificador da mensagem MO que originou a mensagem a ser enviada
app_specific*		Texto	N	Texto livre a ser utilizado pela aplicação.
app_request_id*		Texto	N	Identificador da transação gerado pela Aplicação de forma circular, garantindo uma unicidade em dado intervalo de tempo. Este identificador permite que o Tangram reconheça pedidos repetidos e, neste caso, retorne um erro para a aplicação, caso este pedido seja resultante de uma retentativa de envio indevida.
request_datetime		DDMMYYHHNNSSZZZ	S	Data/hora do pedido

\* O campo **app\_specific** será informado na notificação e, opcionalmente, na geração de arquivos MDR.

## 1.3.4 Documento XML de Resposta

### 1.3.4.1 DTD

```
<!ELEMENT tangram_response (send)>
<!ATTLIST tangram_response
  company_id CDATA #REQUIRED
  service_id CDATA #REQUIRED
```

```
>
<!ELEMENT send (destination*, description, response_datetime)>
<!--ATTLIST send
    code CDATA #REQUIRED
-->
<!--ELEMENT destination (#PCDATA | message_id)>
<!--ATTLIST destination
    code CDATA #IMPLIED
    description CDATA #IMPLIED
    session_id CDATA #IMPLIED
    package_id CDATA #IMPLIED
-->
<!--ELEMENT message_id (#PCDATA)>
<!--ELEMENT description (#PCDATA)>
<!--ATTLIST description
    code CDATA #REQUIRED
-->
<!--ELEMENT response_datetime (#PCDATA)>
```

#### 1.3.4.2 XML de exemplo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tangram_response company_id="2" service_id="21">
    <send code="0">
        <destination code="0" description="Message accepted" session_id="A434FD02"
            package_id="">
            <message_id>A434FD02</message_id>
            <message_id>A434FD03</message_id>
            3191324567</destination>
        <destination code="0" description="Message accepted" session_id="A434FD04"
            package_id="">
            <message_id>A434FD04</message_id>
            <message_id>A434FD05</message_id>
            3192345678</destination>
        <destination code="0" description="Message accepted" session_id="A434FD06"
            package_id="">
            <message_id>A434FD06</message_id>
            <message_id>A434FD07</message_id>
            3193456789</destination>
        <description code="0">Message accepted</description>
        <response_datetime>180902125033020</response_datetime>
```

</send>

</tangram\_response>

### 1.3.4.3 Descrição dos elementos e atributos

Elemento	Atributo	Tipo/Valores	Obrig.	Descrição
tangram_response	company_id	Inteiro	S	Número inteiro identificador da Empresa.
tangram_response	service_id	Inteiro	S	Número inteiro identificador do Serviço.
send	code	0,1	S	Indica o resultado completo da operação. Valores: <ul style="list-style-type: none"> <li>0 = sucesso</li> <li>1 = falha</li> </ul>
destination	code	Inteiro	N	Indica o resultado da operação para este destinatário. Valores: <ul style="list-style-type: none"> <li>0 = sucesso</li> <li>outros = falha</li> </ul>
destination	description	Texto	N	Mensagem descritiva sobre o resultado da operação para este destinatário
destination	session_id	Texto	N	Identificador da sessão do destinatário da mensagens com a aplicação.
destination	package_id	Texto	N	Identificador do pacote de mensagens para o destinatário. Presente apenas se o pedido informar os dados do pacote. Este identificar será informado na notificação de cada item do pacote, caso tenha sido solicitada.
message_id		Texto	S	Identificador gerado pelo TANGRAM para a mensagem enviada para este destinatário. Presente apenas se o atributo <b>[destination] code = 0</b>
destination		Texto	N	Destinatário(s) da mensagem
description	code*	Inteiro	S	Indica o resultado da operação completa. Valores: <ul style="list-style-type: none"> <li>0 = sucesso</li> <li>outros = código de erro da falha encontrada</li> </ul>
description		Texto	S	Mensagem descritiva sobre o resultado da operação completa.
response_datetime		DDMMYYHHNNSSZZZ	S	Data e hora do recebimento do pedido

#### 1.3.4.4 Descrição dos códigos de erro

Código	Descrição
<b>Geral</b>	
0	Operação realizada com sucesso
1	Ocorreu uma falha desconhecida e/ou não documentada
<b>Erros de Autenticação (validação da aplicação)</b>	
101	Acesso negado. Usuário não autorizado para este serviço
102	Acesso negado. IP não autorizado para este serviço
103	Operação não autorizada. O canal informado não existe ou foi informado incorretamente
104	Acesso negado. O canal informado não existe ou foi informado incorretamente
105	Acesso negado. O identificador do serviço é inválido
106	Acesso negado. O serviço foi desabilitado pelo administrador da plataforma
107	Acesso negado. O serviço foi suspenso pelo administrador da plataforma
<b>Erros de Autorização (validação de permissões e provisionamento)</b>	
203	Operação não autorizada. Permissão não existente para esta operação
204	Operação não autorizada. Permissão negada para esta operação
206	Operação não autorizada por completo. Alguns ou todos os clientes não estão provisionados neste canal
207	Operação não autorizada. O cliente não está cadastrado
208	Operação não autorizada. O cliente não está provisionado neste canal
209	Operação não autorizada. O cliente está provisionado para recusar as mensagens deste canal
210	Operação não autorizada. O cliente está desabilitado
211	Operação não autorizada. O cliente está suspenso
212	Operação não autorizada. O cliente está inadimplente
213	Operação não autorizada. O canal informado não existe ou foi informado incorretamente
214	Operação não autorizada. O canal não foi informado
215	Operação não autorizada. O identificador do serviço é inválido

216	Operação não autorizada. Permissão negada para alteração de um dos parâmetros solicitados
217	Operação não autorizada. Permissão não definida para esta operação
218	Operação não autorizada. O cliente não possui saldo para essa operação
<b>Formatação da mensagem (validação do pedido)</b>	
1000	Erro não especificado ao processar o pedido
1001	Identificador da aplicação não informado
1002	Número do cliente (ctn) não informado
1003	Código da operação (operation@code) não informado
1004	Código da operação (operation@code) inválido
<b>Código específico da operação (autenticação ou recuperação de informação)</b>	
4001	Erro ao recuperar informações da plataforma da operadora
4002	Número do cliente (ctn) inválido ou não encontrado na plataforma
4003	Senha (password) incorreto

## 1.4 Interface SMS

Além da interface HTTP, é possível a troca de mensagens entre aplicações e o Tangram através de protocolos SMS como o SMPP (*Short Message Peer to Peer*) ou EMI/UCP (*External Machine Interface / Universal Computer Interface*).

### 1.4.1 SMPP

A interface para envio de mensagens SMPP atualmente suporta os parâmetros listados a seguir (para uma referência completa do protocolo consulte o documento "Short Message Peer to Peer - Protocol Specification v3.4", SMPP Developers Forum):

- source\_addr
- destination\_addr
- data\_coding
- short\_message
- validity\_period
- schedule\_delivery\_time
- registered\_delivery
- service\_type

## 1.5 Regras para tarifação

O Tangram permite que sejam configuradas regras para tarifação dos itens enviados por uma aplicação, através da interface de pedidos MT. Estas configurações determinam como os itens entregues ao Tangram serão tarifados nas plataformas de pré-pago e pós-pago da operadora.

Apesar da regra de cobrança estar definida numa área fora do controle da aplicação, esta poderá auxiliar neste processo, fornecendo informações adicionais, que serão relevantes durante o acerto de contas com a operadora e o detalhamento dos itens tarifados junto aos clientes finais de seus produtos.

As regras de cobrança são definidas no Tangram para cada canal da aplicação. Ou seja, para que a cobrança seja realizada de forma correta, a aplicação deverá informar o canal (elemento **channel\_id**) do item que está sendo enviado, além de outras informações adicionais.

Estas informações adicionais são determinadas nos dados do pacote do item (atributos do elemento **package**), sendo que, quando este elemento é especificado no XML de pedido, todas as mensagens MT geradas no pedido são consideradas como um único item tarifado.

Assim, temos as seguintes possibilidades de tarifação, onde apenas *um registro de cobrança é gerado em cada caso*:

- **Tarifação por uma única mensagem MT para um único destinatário.** Neste caso, é informado apenas um destinatário e um texto, ou seja, apenas uma mensagem SMS é gerada ao fim do processo.
- **Tarifação por várias mensagens MT para um único destinatário.** Neste caso, é informado apenas um destinatário e vários textos, o que causa a geração de vários SMS para este único destinatário.
- **Tarifação por uma única mensagem MT para vários destinatários.** Neste caso, são informados vários destinatários mas apenas um texto, o que causa a geração de vários SMS, um para cada destinatário.
- **Tarifação por várias mensagens MT para vários destinatários.** Neste caso, são informados vários destinatários e várias mensagens de texto, o que resulta em vários SMS enviados para cada destinatário.

A operação de tarifação atrelada a cada um destes casos pode resultar em sucesso ou falha durante a comunicação com as plataformas de cobrança, de acordo com cada cliente. Este resultado pode significar a entrega ou não do item (mensagens SMS), e pode ser interessante para a aplicação que este fato seja de seu conhecimento.

Assim, caso a aplicação precise saber se o item enviado foi ou não entregue para a SMSC, ou seja, se o item foi tarifado corretamente, ela deverá solicitar uma notificação de entrega deste item na plataforma SMS (elemento **notification** do XML de pedido). No caso de falhas na tarifação, será então gerada uma notificação para a aplicação, onde será informada a causa desta falha.

## 2 Anexo 1 – Mapeamento com o protocolo SMPP

Alguns dos campos do XML de pedido do Tangram possuem uma relação direta com os campos do protocolo SMPP, padrão no envio de mensagens SMS. Outros, entretanto, são utilizados apenas internamente.

A seguir é dada como referência uma tabela de correspondência entre os campos do XML e os parâmetros do protocolo SMPP.

XML do Tangram		Protocolo SMPP	Observação
Elemento	Atributo	Parâmetro	
tangram_request	company_id	-	Utilizado internamente no Tangram
tangram_request	service_id	-	Utilizado internamente no Tangram
send	keep_session	-	Utilizado internamente no Tangram
source		source_addr	
destination		destination_addr	
channel_id		-	Utilizado internamente no Tangram
text	binary	data_coding	Caso <b>text@binary</b> ="true", o campo <b>data_coding</b> é setado para 0x04. Caso contrário, é mantido seu valor padrão, 0x00.
text	udh	short_message	O conteúdo deste campo é inserido no campo <b>short_message</b> do SMPP, concatenado ao conteúdo do campo <b>text</b> .
text		short_message	Caso <b>text@binary</b> ="true", o campo <b>short_message</b> recebe o conteúdo deste campo em formato hexadecimal. Caso contrário, o campo <b>short_message</b> recebe o conteúdo deste campo em formato de texto simples. O conteúdo é concatenado após o conteúdo do campo <b>udh</b> , caso exista.
user_data_header		short_message	Mesmo significado do campo <b>text@udh</b>
validity	relative	validity_period	Caso <b>validity@relative</b> ="true", é utilizada a notação de data relativa do protocolo. Caso contrário, é utilizada a notação de data absoluta.
validity		validity_period	A formatação de data obedece o parâmetro <b>validity@relative</b>
schedule	relative	schedule_delivery_time	Caso <b>schedule@relative</b> ="true", é utilizada a notação de data relativa do protocolo. Caso contrário, é utilizada a notação de data absoluta.
schedule		schedule_delivery_time	A formatação de data obedece o parâmetro <b>schedule@relative</b>



notification	type	registered_delivery	Caso seja pedida a notificação de entrega no celular (sucesso ou falha), este campo recebe o valor 0x01. Caso contrário, é enviado com o valor padrão, 0x00.
notification	calltype	-	Utilizado internamente no Tangram
notification		-	Utilizado internamente no Tangram
package	name	-	Utilizado internamente no Tangram
package	code	-	Utilizado internamente no Tangram
package	description	-	Utilizado internamente no Tangram
package	owner_ctn	-	Utilizado internamente no Tangram
retries	max	-	Utilizado internamente no Tangram
retries	interval	-	Utilizado internamente no Tangram
service_type		service_type	Utilizado internamente no Tangram
mo_message_id		-	Utilizado internamente no Tangram
app_specific*		-	Utilizado internamente no Tangram
request_datetime		-	Utilizado internamente no Tangram

## 3 Anexo 2 – Exemplos em ASP

```
<%
' *****
' Exemplo de envio de mensagem para um único destinatário,
' no caso de um serviço que já tenha o originador definido
' internamente no TANGRAM
' *****

' ***** constantes utilizadas para o envio *****
' url do TANGRAM, HTTPS
const strTangramURL = "https://tangram/scripts/tangram.asp"
' código do serviço
const strServiceId = "015"
' usuário e senha do serviço que, juntamente com o IP servidor que vai
implicitamente no pedido Http,
' serão utilizados para validação da aplicação pelo servidor Web
const strTangramUser = "usuario"
const strTangramPwd = "senha"

' ***** monta o XML de envio *****
' recupera o destinatário da mensagem
strDestination = Request("destination")
' recupera o texto da mensagem
strText = Request("text")

' monta a string XML
strXML = _
"<xml version=""1.0"" encoding=""ISO-8859-1"">" & _
"<tangram_request service_id="" & strServiceId & "">" & _
"    <send>" & _
"        <destination>" & strDestination & "</destination>" & _
"        <text>" & strText & "</text>" & _
"    </send>" & _
"</tangram_request>"

' ***** envia ao TANGRAM *****
' carrega a string XML par o componente
Set oDoc = CreateObject("MSXML2.DOMDocument")
oDoc.async = false
oDoc.loadXML strXML

' faz o pedido HTTP
Set oHttp = CreateObject("MSXML2.ServerXMLHttp")
oHttp.SetTimeouts 10000, 10000, 10000, 10000
oHttp.open "POST", strTangramURL, False, strTangramUser, strTangramPwd
oHttp.send oDoc
strXML = oHttp.responseText

' ***** verifica o resultado da operação *****
' recupera o código de retorno da operação
oDoc.loadXML strXML
strCode = oDoc.
selectSingleNode("/tangram_response/send").attributes.getNamedItem("code").text
if strCode = "0" then
' redireciona para a página de sucesso
```

```
Response.Redirect "sucesso.asp"
else
    ' recupera a descrição do erro
    strDescription =
oDoc.selectSingleNode("/tangram_response/send/description").text
    ' redireciona para a página de erro
    Response.Redirect "erro.asp?msg=" & strDescription
end if

Set oDoc = Nothing
Set oHttp = Nothing
%>

<%
' *****
' Exemplo de envio de mensagem para um único destinatário
' no caso de um serviço que tenha o originador definido
' *****

' ***** constantes utilizadas para o envio *****
' url do TANGRAM, HTTPS
const strTangramURL = "https://tangram/scripts/tangram.asp"
' código do serviço
const strServiceId = "015"
' usuário e senha do serviço que, juntamente com o IP servidor que vai
implicitamente no pedido Http,
' serão utilizados para validação da aplicação pelo servidor Web
const strTangramUser = "usuario"
const strTangramPwd = "senha"

' ***** monta o XML de envio *****
' recupera o destinatário da mensagem
strDestination = Request("destination")
' recupera o texto da mensagem
strText = Request("text")

' monta a string XML
strXML = _
"<xml version=""1.0"" encoding=""ISO-8859-1"">" & _
"<tangram_request service_id="" & strServiceId & "">" & _
"    <send>" & _
"        <source>500</source>" & _
"        <destination>" & strDestination & "</destination>" & _
"        <text>" & strText & "</text>" & _
"    </send>" & _
"</tangram_request>"

' ***** envia ao TANGRAM *****
' carrega a string XML par o componente
Set oDoc = CreateObject("MSXML2.DOMDocument")
oDoc.async = false
oDoc.loadXML strXML

' faz o pedido HTTP
Set oHttp = CreateObject("MSXML2.ServerXMLHttp")
oHttp.SetTimeouts 10000, 10000, 10000, 10000
oHttp.open "POST", strTangramURL, False, strTangramUser, strTangramPwd
oHttp.send oDoc
```

```
strXML = oHttp.responseText

' ***** verifica o resultado da operação *****
' recupera o código de retorno da operação
oDoc.loadXML strXML
strCode = oDoc.
selectSingleNode("/tangram_response/send").attributes.getNamedItem("code").text
if strCode = "0" then
' redireciona para a página de sucesso
Response.Redirect "sucesso.asp"
else
' recupera a descrição do erro
strDescription =
oDoc.selectSingleNode("/tangram_response/send/description").text
' redireciona para a página de erro
Response.Redirect "erro.asp?msg=" & strDescription
end if

Set oDoc = Nothing
Set oHttp = Nothing
%>

<%
' *****
' Exemplo de envio de mensagem para um único destinatário
' provisionado em um determinado canal,
' no caso de um serviço ou canal que já tenha o originador
' definido internamente no TANGRAM
' *****

' ***** constantes utilizadas para o envio *****
' url do TANGRAM, HTTPS
const strTangramURL = "https://tangram/scripts/tangram.asp"
' código do serviço/canal
const strServiceId = "015"
const strChannelId = "001"
' usuário e senha do serviço que, juntamente com o IP servidor que vai
implicitamente no pedido Http,
' serão utilizados para validação da aplicação pelo servidor Web
const strTangramUser = "usuario"
const strTangramPwd = "senha"

' ***** monta o XML de envio *****
' recupera o destinatário da mensagem
strDestination = Request("destination")
' recupera o texto da mensagem
strText = Request("text")

' monta a string XML
strXML = _
"<xml version=""1.0"" encoding=""ISO-8859-1"">" & _
"<tangram_request service_id="" & strServiceId & "">" & _
"    <send>" & _
"        <destination>" & strDestination & "</destination>" & _
"        <channel_id>" & strChannelId & "</channel_id>" & _
"        <text>" & strText & "</text>" & _
"    </send>" & _
"</tangram_request>"
```

```

' ***** envia ao TANGRAM *****
' carrega a string XML par o componente
Set oDoc = CreateObject("MSXML2.DOMDocument")
oDoc.async = false
oDoc.loadXML strXML

' faz o pedido HTTP
Set oHttp = CreateObject("MSXML2.ServerXMLHttp")
oHttp.SetTimeouts 10000, 10000, 10000, 10000
oHttp.open "POST", strTangramURL, False, strTangramUser, strTangramPwd
oHttp.send oDoc
strXML = oHttp.responseText

' ***** verifica o resultado da operação *****
' recupera o código de retorno da operação
oDoc.loadXML strXML
strCode = oDoc.
selectSingleNode("/tangram_response/send").attributes.getNamedItem("code").text
if strCode = "0" then
' redireciona para a página de sucesso
Response.Redirect "sucesso.asp"
else
' recupera a descrição do erro
strDescription =
oDoc.selectSingleNode("/tangram_response/send/description").text
' redireciona para a página de erro
Response.Redirect "erro.asp?msg=" & strDescription
end if

Set oDoc = Nothing
Set oHttp = Nothing
%>

<%
' *****
' Exemplo de envio de mensagem para todos os
' assinantes aprovisionados em um canal
' no caso de um serviço ou canal que já tenha o originador
' definido internamente no TANGRAM
' *****

' ***** constantes utilizadas para o envio *****
' url do TANGRAM, HTTPS
const strTangramURL = "https://tangram/scripts/tangram.asp"
' código do serviço e do canal
const strServiceId = "015"
const strChannelId = "001"
' usuário e senha do serviço que, juntamente com o IP servidor que vai
implicitamente no pedido Http,
' serão utilizados para validação da aplicação pelo servidor Web
const strTangramUser = "usuario"
const strTangramPwd = "senha"

' ***** monta o XML de envio *****
' recupera o texto da mensagem
strText = Request("text")

```

```

' monta a string XML
strXML = _
"<xml version=""1.0"" encoding=""ISO-8859-1"">" & _
"<tangram_request service_id="" & strServiceId & "">" & _
"    <send>" & _
"        <channel_id>" & strChannelId & "</channel_id>" & _
"        <text>" & strText & "</text>" & _
"    </send>" & _
"</tangram_request>"

' ***** envia ao TANGRAM *****
' carrega a string XML par o componente
Set oDoc = CreateObject("MSXML2.DOMDocument")
oDoc.async = false
oDoc.loadXML strXML

' faz o pedido HTTP
Set oHttp = CreateObject("MSXML2.ServerXMLHttp")
oHttp.SetTimeouts 10000, 10000, 10000, 10000
oHttp.open "POST", strTangramURL, False, strTangramUser, strTangramPwd
oHttp.send oDoc
strXML = oHttp.responseText

' ***** verifica o resultado da operação *****
' recupera o código de retorno da operação
oDoc.loadXML strXML
strCode = oDoc.
selectSingleNode("/tangram_response/send").attributes.getNamedItem("code").text
if strCode = "0" then
' redireciona para a página de sucesso
    Response.Redirect "sucesso.asp"
else
' recupera a descrição do erro
    strDescription =
oDoc.selectSingleNode("/tangram_response/send/description").text
' redireciona para a página de erro
    Response.Redirect "erro.asp?msg=" & strDescription
end if

Set oDoc = Nothing
Set oHttp = Nothing
%>

<%
' *****
' Exemplo de envio de mensagem para um todos os
' assinantes aprovionados em um serviço sem canais
' ou com um único canal, no caso de um serviço
' que já tenha o originador definido internamente no TANGRAM
' *****

' ***** constantes utilizadas para o envio *****
' url do TANGRAM, HTTPS
const strTangramURL = "https://tangram/scripts/tangram.asp"
' código do serviço e do canal
const strServiceId = "015"

```

```
' usuário e senha do serviço que, juntamente com o IP servidor que vai
implicitamente no pedido Http,
' serão utilizados para validação da aplicação pelo servidor Web
const strTangramUser = "usuario"
const strTangramPwd = "senha"

' ***** monta o XML de envio *****
' recupera o texto da mensagem
strText = Request("text")

' monta a string XML
strXML = _
"<xml version=""1.0"" encoding=""ISO-8859-1"">" & _
"<tangram_request service_id="" & strServiceId & "">" & _
"    <send>" & _
"        <text>" & strText & "</text>" & _
"    </send>" & _
"</tangram_request>"

' ***** envia ao TANGRAM *****
' carrega a string XML par o componente
Set oDoc = CreateObject("MSXML2.DOMDocument")
oDoc.async = false
oDoc.loadXML strXML

' faz o pedido HTTP
Set oHttp = CreateObject("MSXML2.ServerXMLHttp")
oHttp.SetTimeouts 10000, 10000, 10000, 10000
oHttp.open "POST", strTangramURL, False, strTangramUser, strTangramPwd
oHttp.send oDoc
strXML = oHttp.responseText

' ***** verifica o resultado da operação *****
' recupera o código de retorno da operação
oDoc.loadXML strXML
strCode = oDoc.
selectSingleNode("/tangram_response/send").attributes.getNamedItem("code").text
if strCode = "0" then
' redireciona para a página de sucesso
    Response.Redirect "sucesso.asp"
else
' recupera a descrição do erro
    strDescription =
oDoc.selectSingleNode("/tangram_response/send/description").text
' redireciona para a página de erro
    Response.Redirect "erro.asp?msg=" & strDescription
end if

Set oDoc = Nothing
Set oHttp = Nothing
%>
```