



# Manual del Protocolo XML-RPC Mensajería Negocios

## Índice de contenidos

1	INTRODUCCIÓN.....	3
2	FUNCIONALIDADES DEL API DE COMUNICACIÓN XML-RPC .....	4
2.1	Envío Libre .....	4
2.2	Envío a Grupo de Contactos .....	5
2.3	Agregar nuevo Grupo de Contactos.....	6
2.4	Agregar Contacto a Grupo .....	8
2.5	Obtener números de teléfono de Contactos de un Grupo .....	9
2.6	Obtener número de Contactos que tiene Grupo .....	10
2.7	Obtener mensaje en el Buzón de Entrada .....	10
2.8	Obtener mensajes enviados y su estado .....	11
3	ANEXOS.....	14
3.1	Límite del número de conexiones desde la aplicación cliente .....	14
3.2	Formato de los parámetros para las llamadas XML-RPC.....	14
3.3	Códigos de retorno .....	18
3.4	Ejemplos de implementación .....	20
3.4.1	Ejemplo Java Envío Libre .....	20
3.4.2	Ejemplo PHP Envío Libre .....	22
3.4.3	Ejemplo Java Envío a Grupo de Contactos .....	23
3.4.4	Ejemplo PHP Envío a Grupo de Contactos.....	24
3.4.5	Ejemplo Java Agregar nuevo Grupo de Contactos .....	25
3.4.6	Ejemplo PHP Agregar nuevo Grupo de Contactos .....	26
3.4.7	Ejemplo Java Agregar Contacto a Grupo.....	27
3.4.8	Ejemplo PHP Agregar Contacto a Grupo.....	28
3.4.9	Ejemplo Java Obtener número de Contactos que tiene un Grupo .....	29
3.4.10	Ejemplo PHP Obtener número de Contactos que tiene un Grupo .....	30
3.4.11	Ejemplo Java Obtener números de teléfono de Contactos de un Grupo .....	31
3.4.12	Ejemplo Java Obtener mensaje recibidos en el Buzón de Entrada.....	32
3.4.13	Ejemplo Java Obtener mensaje enviados y su estado .....	34
3.4.14	Ejemplo PHP Obtener números de teléfono de Contactos de un Grupo.....	36

## 1 INTRODUCCIÓN

El protocolo de comunicación XML-RPC proporciona la posibilidad de hacer un uso del servicio Mensajería Negocios sin tener que acceder a través del portal Web.

Mediante este acceso se pueden realizar conexiones directas entre sus máquinas y los servidores de Mensajería Negocios.

A lo largo del documento se especificará cómo acceder a las siguientes funcionalidades a través del API:

- Envío Libre
- Envío a Grupo
- Crear un nuevo Grupo de Contactos
- Añadir Contacto a Grupo
- Obtener los números de teléfono de los Contactos de un Grupo
- Obtener el número de Contactos que tiene un Grupo
- Obtener los mensajes recibidos en el Buzón de Entrada
- Obtener los mensajes enviados y su estado

Además podrá encontrar toda la información necesaria para realizar llamadas XML-RPC a través del API y ejemplos de implementación de clientes, tanto en Java como en PHP.

Para hacer un uso adecuado del API es necesaria la siguiente información:

- **URL de conexión** (URL a la que realizar las peticiones XML-RPC): Debe conectarse a la siguiente dirección utilizando conexión segura (acceso HTTPS):

<https://www.mensajerianegocios.movistar.es/SrvConexion>

- **Usuario de acceso:** Se trata del usuario que establece el usuario gestor a través del menú Usuario Api de la Aplicación <sup>1</sup>. Este usuario puede cambiarse posteriormente a través del mismo menú.
- **Contraseña:** Contraseña asociada al usuario API, se establece de igual forma que el usuario de acceso.

Para cambiar la contraseña posteriormente es necesario conocer la contraseña previa, aunque puede ser consultada pulsando el botón enviar del mismo menú, y se enviará al destinatario de email y sms definido en el contacto .

---

<sup>1</sup> Los usuarios autorizados (no gestores) que se pueden crear a través de la web del servicio no pueden utilizarse como usuarios de acceso al API.

---

## 2 FUNCIONALIDADES DEL API DE COMUNICACIÓN XML-RPC

A continuación se describen los métodos remotos disponibles para conectarse al servidor de envío de mensajes de Mensajería Negocios.

Para mayor información acerca del formato de los parámetros de la llamada consultar *anexo 3.2 Formato de los parámetros para las llamadas XML-RPC*.

### 2.1 Envío Libre

Este método permite el envío de mensajes SMS personalizando el mensaje para cada destinatario, es decir, para cada destinatario se indicará el texto a enviar y el remitente (217812 o Remitente de texto) a utilizar.

Las características más importantes de este tipo de envío son:

- Los destinatarios de los mensajes deben estar almacenados en sus sistemas, es decir, en la aplicación que invoca al API.
- Cada operación de envío permite enviar un máximo de 500 SMS.

Las especificaciones de la llamada son las siguientes:

Nombre del método remoto

**MensajeríaNegocios.enviarSMS**

Parámetros de entrada

El parámetro de entrada es una lista con los siguientes valores:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario proporcionado por Telefónica para acceder al servicio).
- **Contraseña:** Contraseña asociada al usuario API (clave proporcionada al acceder al servicio y que el usuario gestor puede cambiar).
- **Lista de mensajes**, donde cada mensaje es una lista con los siguientes campos:
  - Teléfono, es el destinatario del mensaje.
  - Texto del mensaje a enviar.
  - Remitente del mensaje, el numérico de la aplicación 217812 o uno de los alfanuméricos asignados.

*Ejemplo:* Una lista de SMS podría estar formada por SMS1 y SMS2, de manera que:

SMS1:

- Destinatario: 609000001
- Texto: Texto SMS 1
- Remitente: 217812
- Remitente Alfanumérico Principal: MiRPrincipal

SMS2:

- Destinatario: 609000003
- Texto: Texto SMS 2
- Remitente: MiRemitente

- El SMS 1 se enviaría al destinatario 609000001, que recibiría el texto 'Texto SMS 1 Fdo. MiRPrincipal' por el remitente 217812.
- El SMS2 se enviaría al destinatario 609000003, que recibiría el texto 'Texto SMS 2' por el remitente MiRemitente.

### Códigos de retorno

Esta llamada devuelve un código de error o una lista de valores (ver tabla del *Anexo 3.3*):

- Si la llamada se ejecuta con éxito, devuelve una lista de valores que representa que se ha realizado el envío de cada uno de los SMS indicados como parámetro en la lista de mensajes. Cada uno de los elementos de la lista son códigos numéricos que representan el código de retorno de cada mensaje

*Ejemplo:* Si en la llamada incluimos 3 mensajes, los dos primeros son correctos y el último da error, nuestro retorno sería (0,0,4).

- Caso contrario, devuelve un código de error numérico que representa que se ha producido alguna anomalía que ha imposibilitado el envío de todos y cada uno de los mensajes pasados como parámetro en la lista de mensajes.

*Ejemplo:* Si en la llamada indicamos un usuario de acceso al API no válido, recibiremos el código de error numérico -2, y el envío no se habrá realizado.

## **2.2 Envío a Grupo de Contactos**

Este método permite el envío de un mensaje a un Grupo de Contactos. Todos los contactos del grupo recibirán el mismo mensaje.

Las características más importantes de este tipo de envío son:

- El Grupo de Contactos destinatario del mensaje habrá sido creado previamente en Mensajería Negocios, bien a través del API, bien a través de la página web del servicio. Es

---

decir, independientemente de la forma de creación del grupo, los números de teléfono de dichos contactos se obtienen de la base de datos de Mensajería Negocios.

- El mismo mensaje es enviado a todos los contactos del grupo, con el mismo contenido y el mismo remitente.
- Este mensaje se envía a todos los contactos del grupo y al propio administrador.

Las especificaciones de la llamada son las siguientes:

Nombre del método remoto

### **MensajeríaNegocios.enviarAGrupoContacto**

Parámetros de entrada

El parámetro de entrada es una lista con los siguientes valores y en el siguiente orden:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario configurado en la aplicación para acceder al servicio vía API).
- **Contraseña:** Contraseña asociada al usuario API (clave que el usuario gestor establece en la aplicación y que él mismo puede consultar o cambiar).
- **Nombre del Grupo** al que enviar el mensaje
- **Texto** del mensaje
- **Remitente** del mensaje

Códigos de retorno

Esta llamada devuelve un valor numérico que corresponde a un código de retorno de la tabla del Anexo 3.3. El código de retorno indica si la operación de envío de grupo ha sido aceptada por el servidor de Mensajería Negocios. El resultado del envío a cada destinatario del grupo se puede visualizar a través de la página Web del servicio.

### **2.3 Agregar nuevo Grupo de Contactos**

Este método permite crear un nuevo Grupo de Contactos en Mensajería Negocios.

Las especificaciones de la llamada son las siguientes:

Nombre del método remoto:

### **MensajeríaNegocios.agregarGrupoContactos**

Parámetros de entrada

El parámetro de entrada es una lista con los siguientes valores y en el siguiente orden:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario configurado en la aplicación para acceder al servicio vía API).
- **Contraseña:** Contraseña asociada al usuario API (clave que el usuario gestor establece en la aplicación y que él mismo puede consultar o cambiar).
- **Nombre del Grupo** a crear
- **Mensaje de Bienvenida** que se envía a un nuevo contacto del grupo que se suscribe en el mismo. Este mensaje tiene un máximo de 160 caracteres y se envía con el remitente alfanumérico asignado al cliente.

#### Códigos de retorno

Esta llamada devuelve un valor numérico que corresponde a un código de retorno de la tabla del *Anexo 3.3*.

## 2.4 Agregar Contacto a Grupo

Este método permite incluir un nuevo contacto en un Grupo de Contactos existente en Mensajería Negocios.

Las especificaciones de la llamada son las siguientes:

Nombre del método remoto

**MensajeríaNegocios.agregarContactoaGrupoContactos**

Parámetros de entrada

El parámetro de entrada es una lista con los siguientes valores y en el siguiente orden:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario configurado en la aplicación para acceder al servicio vía API).
- **Contraseña:** Contraseña asociada al usuario API (clave que el usuario gestor establece en la aplicación y que él mismo puede consultar o cambiar).
- **Nombre del Grupo** al que se quiere añadir el Contacto
- **Número de teléfono** del contacto a añadir

Códigos de retorno

Esta llamada devuelve un valor numérico que corresponde a un código de retorno de la tabla del *Anexo 3.3*



## 2.5 Obtener números de teléfono de Contactos de un Grupo

Este método permite consultar los números de teléfono de los Contactos pertenecientes a un Grupo de Contactos en Mensajería Negocios.

Las especificaciones de la llamada son las siguientes:

Nombre del método remoto

**MensajeríaNegocios.obtenerContactosdeGrupoContactos**

Parámetros de entrada

El parámetro de entrada es una lista con los siguientes valores y en el siguiente orden:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario configurado en la aplicación para acceder al servicio vía API).
- **Contraseña:** Contraseña asociada al usuario API (clave que el usuario gestor establece en la aplicación y que él mismo puede consultar o cambiar).
- **Nombre del Grupo** a consultar

Códigos de retorno

Los posibles códigos de retorno son:

- Si la llamada se ejecuta con éxito, devuelve un número positivo que indica el número de contactos que tiene el grupo.

*Ejemplo:* Devuelve 609000001 606000002 si el grupo tiene dos contactos.

- Caso contrario, devuelve un número negativo que corresponde a un código de retorno de la tabla del *Anexo 3.3*.

## 2.6 Obtener número de Contactos que tiene Grupo

Este método permite consultar el número de Contactos que tiene asociados un Grupo dado de alta en Mensajería Negocios.

Las especificaciones de la llamada son las siguientes:

Nombre del método remoto

**MensajeríaNegocios.obtenerNumeroContactosdeGrupoContactos**

Parámetros de entrada

El parámetro de entrada es una lista con los siguientes valores:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario configurado en la aplicación para acceder al servicio vía API).
- **Contraseña:** Contraseña asociada al usuario API (clave que el usuario gestor establece en la aplicación y que él mismo puede consultar o cambiar).
- **Nombre del Grupo** a consultar

Códigos de retorno

Los posibles códigos de retorno son:

- Si la llamada se ejecuta con éxito, devuelve un número positivo que indica el número de contactos que tiene el grupo.  
*Ejemplo:* Devuelve 5 si el grupo tiene 5 contactos.
- Caso contrario, devuelve un número negativo que corresponde a un código de retorno de la tabla del Anexo 3.3.

## 2.7 Obtener mensaje en el Buzón de Entrada

Este método permite leer los mensajes recibidos en el Buzón de Entrada.

Las especificaciones de la llamada son las siguientes:

Nombre del método remoto

**MensajeríaNegocios.obtenerSMSRecibidos**

Parámetros de entrada

El parámetro de entrada es una lista con los siguientes valores y en el siguiente orden:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario configurado en la aplicación para acceder al servicio vía API).

- **Contraseña:** Contraseña asociada al usuario API (clave que el usuario gestor establece en la aplicación y que él mismo puede consultar o cambiar).
- **Fecha Inicial** a consultar, ejemplo 2016-01-01 00:00:00
- **Fecha Final** a consultar, ejemplo 2016-02-01 00:00:00

**Nota:** El rango de fechas no puede ser superior a un día.

#### Códigos de retorno

Los posibles códigos de retorno son:

- Si la llamada se ejecuta con éxito, devuelve una array con la lista de mensajes recibidos en el buzón de entrada con el formato fecha;MSISDN;TEXT0;1/0. El valor final indica si el mensaje ha sido leído o no. 0=> No leído, 1=>Leído.

*Ejemplo respuesta:*

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<methodResponse xmlns:ex="http://ws.apache.org/xmlrpc/namespaces/extensions">
  <params>
    <param>
      <value>
        <array>
          <data>
            <value>01/10/2013 12:35:07;34650424244;test;1</value>
            <value>01/10/2013 12:27:59;34650424244;test;0</value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodResponse>
```

- Caso contrario, devuelve un número negativo que corresponde a un código de retorno de la tabla del **Anexo 3.3**.

## 2.8 Obtener mensajes enviados y su estado

Este método permite obtener los mensajes enviados y su estado.

Las especificaciones de la llamada son las siguientes:

Nombre del método remoto

**MensajeríaNegocios.obtenerSMSEnviados**

Parámetros de entrada

El parámetro de entrada es una lista con los siguientes valores y en el siguiente orden:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario configurado en la aplicación para acceder al servicio vía API).
- **Contraseña:** Contraseña asociada al usuario API (clave que el usuario gestor establece en la aplicación y que él mismo puede consultar o cambiar).
- **Fecha Inicial** a consultar, ejemplo 2016-01-01 00:00:00
- **Fecha Final** a consultar, ejemplo 2016-02-01 00:00:00

**Nota:** El rango de fechas no puede ser superior a un día.

### Códigos de retorno

Los posibles códigos de retorno son:

- Si la llamada se ejecuta con éxito, devuelve un array con la lista de mensajes enviados y su estado.

*Ejemplo respuesta:*

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<methodResponse xmlns:ex="http://ws.apache.org/xmlrpc/namespaces/extensions">
  <params>
    <param>
      <value>
        <array>
          <data>
            <value>
              <array>
                <data>
                  <value>MNALERTAS</value>
                  <value>34699486685</value>
                  <value>2013-01-11 02:00 Testing new probe</value>
                  <value>ALERTAS</value>
                  <value>11/01/13 00:00</value>
                  <value>1</value>
                  <value>DELIVRD</value>
                </data>
              </array>
            </value>
          <value>
            <array>
              <data>
                <value>MNALERTAS</value>
                <value>34699486685</value>
                <value>2013-01-12 01:45 Testing new probe</value>
                <value>ALERTAS</value>
                <value>11/01/13 23:45</value>
                <value>1</value>
                <value>DELIVRD</value>
              </data>
            </array>
          </value>
        </data>
      </array>
    </param>
  </params>
</methodResponse>
```

```
</array>  
</value>  
</param>  
</params>  
</methodResponse>  
</methodResponse>
```

- Caso contrario, devuelve un número negativo que corresponde a un código de retorno de la tabla del **Anexo 3.3**.

### 3 ANEXOS

#### 3.1 Límite del número de conexiones desde la aplicación cliente



**La aplicación únicamente puede establecer una conexión simultánea** con los servidores de SMS de Mensajería Negocios. Además, **la comunicación** entre su aplicación y los servidores de Mensajería Negocios **es siempre síncrona**, es decir, sólo se puede realizar una petición simultánea al API, hasta que el API no devuelve respuesta a una operación la aplicación cliente no puede lanzar la siguiente petición.

*Por ejemplo*, si se quiere realizar un envío de 700 mensajes, considerando que el caudal máximo es de 500 SMS por envío, deberá realizar una primera conexión invocando al método *EnviarSMS* con los 500 primeros destinatarios. Una vez que el API haya devuelto el código de retorno que indique que los 500 envíos han finalizado, podrá realizar una segunda invocación para enviar los 200 mensajes restantes.



**El caudal máximo es de 500 SMS por envío**, es decir, se podrán realizar tantas operaciones de envío como se requiera pero en bloques de 500 SMS, y no se podrá enviar más hasta que finalice la conexión.

#### 3.2 Formato de los parámetros para las llamadas XML-RPC

A continuación se describe la sintaxis que deben cumplir los distintos parámetros del sistema:

- **Nombre de Usuario:** usuario de acceso (se corresponde con el usuario configurado en la aplicación para acceder al servicio vía API).
- **Contraseña:** Contraseña asociada al usuario API (clave que el usuario gestor establece en la aplicación y que él mismo puede consultar o cambiar).

---

- **Nombre de Grupo**

El nombre de un grupo de contactos permite los siguientes caracteres: 0123456789\_ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz.

La longitud máxima es de 8 caracteres.

- **Mensaje de Bienvenida**

El mensaje de bienvenida que recibe un usuario que se suscribe a un grupo de contactos tiene una longitud máxima de 160 caracteres.

Los caracteres permitidos son los mismos que los permitidos para el texto de un mensaje enviado a través de API (descrito en este mismo apartado, en "Texto del mensaje").

- **Teléfonos o Contactos**

Los teléfonos móviles permitidos en la aplicación, son exclusivamente los de **numeración española**, con o sin prefijo de España (34).

Se define como numeración española aquellos teléfonos que cumplen las siguientes características:

- Tienen una longitud de 9 dígitos, sin contar el prefijo
- El primer dígito del número debe de ser 6 ó 7, sin contar el prefijo
- Pueden ir precedidos del prefijo de España: 34

Cualquier otro formato de numeración se toma como incorrecto.

*Ejemplo* de numeración correcta:

606000007 ó 34606000007 ó 706000007 ó 34706000007

- **Remitente del mensaje**

El remitente de los SMS enviados puede ser de dos tipos:

- **Numérico** por defecto: 217812
- **Alfanumérico(s)** asignado(s) por Telefónica en el momento del alta de cliente.

Existen dos tipos:

- Remitente Alfanumérico Principal: Es obligatorio disponer al menos de un remitente de texto. Será utilizado como firma en los mensajes enviados con remitente 217812.
- Remitentes Alfanuméricos Secundarios: Hasta 5 remitentes de texto opcionales.

Para seleccionar el remitente más adecuado para un envío recuerde elegir el 217812 si desea que quien lo reciba pueda contestar a dicho mensaje. Elija un remitente alfanumérico si no desea contestación.

▪ **Texto del mensaje**

El contenido del mensaje de texto no debe exceder los 459 caracteres (3 SMS concatenados). Existen dos casos:

- En caso de realizar el envío de un mensaje con remitente numérico, la longitud máxima del mensaje es de **459 caracteres**.
- Sin embargo, en caso de realizar un envío con el remitente por defecto 217812, se añade automáticamente una firma al final del contenido del mensaje, por lo que se contabiliza la longitud de la firma junto con la longitud del mensaje. Es decir, si elige como remitente el remitente por defecto 217812 la longitud máxima de su texto podrá ser **459 - <longitud de la firma>**.

*Por ejemplo:*

Si el texto del mensajes es “Se adelanta la reunión a las 12 horas.”, y el remitente alfanumérico principal que tiene asignado su empresa es “MiRemitente” y se realiza un envío con remitente numérico (217812), el mensaje que recibirán en sus teléfonos los clientes es:”Se adelanta la reunión a las 12 horas. Fdo: MiRemitente”

Los caracteres permitidos en los contenidos de los mensajes son:

**01234567890  
ABCDEFGHIJKLMNÑOPQRSTUVWXYZ  
abcdefghijklmnñopqrstuvwxyz  
ÇäöüÄÖÜÆÅå@£\$¥Øø!"#  
%&'()\*+,-./:;<=>§¿?  
y Espacio**



Los caracteres no permitidos se convierten automáticamente a otros válidos siguiendo la siguiente tabla de conversiones:

Carácter	Conversión
Á	A
á	a
É	E
é	e
Í	I
í	i
Ó	O
ó	o
Ú	U
ú	u
à	a
è	e
ì	i
ò	o
ù	u
_	espacio
ç	c
^	'
{	(
}	)
	/
[	(
]	)
€	E



**IMPORTANTE:** Se debe utilizar la codificación LATIN-15 o ISO 8859-15, de no ser así, es muy posible que no se pueda enviar el mensaje.

### 3.3 Códigos de retorno

La siguiente tabla detalla el significado de los códigos de retorno de los métodos del API XML-RPC:

SIGNIFICADO	CÓDIGO DE RETORNO
<b>Operación correcta</b> La operación se ha realizado correctamente.	0
<b>Contraseña incorrecta</b> La contraseña introducida en la autenticación no es correcta, no corresponde al usuario de acceso indicado.	-1
<b>Usuario incorrecto</b> El usuario introducido en la autenticación no es un usuario válido o no es un administrador de la aplicación (los usuarios delegados no pueden acceder al API).	-2
<b>Usuario bloqueado</b> El usuario introducido en la autenticación está bloqueado o ha consumido 50 mensajes, por lo que tiene restringido el acceso al servicio.	-3
<b>Error en la aplicación</b> Ha ocurrido un error grave en la aplicación. Póngase en contacto con Atención Telefónica Movistar (1489+200200203).	-4
<b>Teléfono móvil incorrecto</b> El teléfono introducido tiene sintaxis incorrecta, está repetido o pertenece a un gestor o usuario autorizado.	-5
<b>Número destinatarios sobrepasado</b> El número de destinatarios del mensaje excede el máximo número de destinatarios permitidos (máximo 500) en un solo envío.	-6
<b>Lista de mensajes vacía</b> La lista de mensajes SMS pasada como parámetro está vacía.	-7
<b>Texto del mensaje - Longitud errónea</b> La longitud del texto del mensaje es errónea, excede el número de caracteres permitidos o está vacío.	-8
<b>Texto del mensaje - Sintaxis errónea</b> El texto del mensaje contiene caracteres no permitidos.	-9

<b>Fechas fuera de rango</b> Se ha solicitado obtener mensajes del buzón de entrada con un rango de fechas superior a un día.	<b>-10</b>
<b>Remitente erróneo</b> El remitente utilizado es incorrecto o está vacío o no es uno de sus remitentes disponibles en el servicio.	<b>-11</b>
<b>Grupo</b> Al crear el grupo se ha indicado un nombre de grupo vacío, con sintaxis errónea o el grupo ya existe. Al modificar un grupo existente, se devuelve este código si el grupo no existe.	<b>-13</b>
<b>Grupo – No contiene contactos</b> El Grupo no tiene ningún contacto.	<b>-14</b>
<b>Número de parámetros incorrecto</b> En un envío libre, la lista de mensajes no está correctamente definida ya que el número de parámetros de alguno de los SMS es incorrecto. Revise que se ha definido para cada SMS su remitente, texto, destinatario y rango de fechas.	<b>-16</b>
<b>Mensaje de Bienvenida incorrecto</b> El mensaje de bienvenida contiene caracteres no permitidos o está vacío.	<b>-18</b>
<b>Grupo o Mensaje de bienvenida erróneo</b> El nombre de Grupo o el mensaje de bienvenida excede el máximo número de caracteres permitidos (8 y 160 caracteres respectivamente).	<b>-19</b>
<b>Conexiones simultáneas al servidor de Mensajería Negocios</b> No se pueden establecer dos conexiones simultáneas al sistema (ver <i>Anexo - Límite del número de conexiones desde la aplicación cliente</i> ).	<b>-20</b>

### 3.4 Ejemplos de implementación

Para el correcto funcionamiento de los ejemplos indicados a continuación es necesario el uso de una librería XML-RPC tanto para Java como para PHP. Se pueden usar, por ejemplo, las siguientes (incluyendo sus dependencias si las hubiese):

- **Java:** *xmlrpc-3.1.jar* disponible en <http://ftp.udc.es/apache-dist/ws/xmlrpc/>
- **PHP:** *IXR\_Library.inc.php* disponible en <http://scripts.incutio.com/xmlrpc/>

**NOTA: Si se desea utilizar un cliente no JAVA ni PHP consultar:**

<http://directory.xmlrpc.com/implementations>

#### 3.4.1 Ejemplo Java Envío Libre

```
import java.net.URL;
import java.util.Vector;

import org.apache.xmlrpc.client.XmlRpcClient;
import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;

public class EnvioSMS {
    /**
     * @param args
     */
    public static void main(String[] args) {
        String url_conexion= "https://www.mensajerianegocios.movistar.es/SrvConexion";
        String login= "login"; //Nombre de usuario
        String password= "contrasena"; //Contraseña

        try {
            //Conectar con el Servidor
            XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
            config.setServerURL(new URL(url_conexion));
            config.setEncoding("ISO-8859-15");
            //Construir llamada
            Vector params = new Vector();
            //Primer parámetro: Login
            params.addElement(login);
            //Segundo parámetro: Contraseña
            params.addElement(password);

            //Composición de los SMS
            Vector SMS = new Vector();

            //Crear primer SMS
            Vector primer_SMS = new Vector();
            //Introducir telefono
            primer_SMS.addElement("609000001");
            //Introducir texto del mensaje
            primer_SMS.addElement("Texto primer mensaje");
            //Introducir remitente
            primer_SMS.addElement("miRemite");

            //Incluir primer SMS en la lista
```

```
SMS.addElement(primer_SMS);

//Crear segundo SMS
Vector segundo_SMS = new Vector();
//Introducir telefono
segundo_SMS.addElement("609000003");
//Introducir texto del SMS
segundo_SMS.addElement("Texto segundo mensaje");
//Introducir remitente, elegimos remitente por defecto 217812
segundo_SMS.addElement("217812");
//Incluir segundo SMS en la lista
SMS.addElement(segundo_SMS);

//Introducir los SMS para la llamada
params.addElement(SMS);

//Realizar la petición al servidor
XmlRpcClient client = new XmlRpcClient();
client.setConfig(config);
Object result = client.execute("MensajeriaNegocios.enviarSMS",
params);

//Imprimir los resultados
if(result instanceof Integer) {
    System.out.println(result);
}else{
    for(int i=0; i<((Object[])result).length;i++){
        System.out.println(((Object[])result)[i]);
    }
}
}catch (Exception e) {
    e.printStackTrace();
}

}

}
```

### 3.4.2 Ejemplo PHP Envío Libre

```
<?php
include ('IXR_Library.inc.php'); //Libreria para el uso de xml-rpc

$url_conexion='https://www.mensajerianegocios.movistar.es/SrvConexion';
$login='Login'; //Nombre de usuario
$password='Contrasena'; //Contraseña

$client = new IXR_Client($url_conexion);

//Ejecuta el metodo rpc
$client->query ('MensajeriaNegocios.enviarSMS', $login, $password,
array(array('606000001','Mensaje 1', 'miRemite'), array('606000003', 'Mensaje 2 ', '217812')));

echo '<pre>';
print_r($client->getResponse());
echo '</pre>';
?>
```

### 3.4.3 Ejemplo Java Envío a Grupo de Contactos

```
import java.net.URL;
import java.util.Vector;
import org.apache.xmlrpc.client.XmlRpcClient;
import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;

public class EnvioGrupo {

    public static void main(String[] args) {
        String url_conexion= "https://www.mensajerianegocios.movistar.es/SrvConexion";
        String login= "login"; //Nombre de usuario
        String password= "contrasena"; //Contraseña
        String colectivo="Grupo"; //Nombre de Grupo
        String texto="Texto del mensaje"; //Texto del mensaje
        String remite="Remite del mensaje"; //Remite del mensaje, 217812 o remite de
usuario
        try {
            //Conectar con el Servidor
            XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
            config.setServerURL(new URL(url_conexion));
            config.setEncoding("ISO-8859-15");
            //Construir llamada
            Vector params = new Vector();
            //Primer parámetro: Login
            params.addElement(login);
            //Segundo parámetro: Contraseña
            params.addElement(password);
            //Tercer parámetro: Grupo
            params.addElement(colectivo);
            //Cuarto parámetro: Mensaje de texto
            params.addElement(texto);
            //Quinto parámetro: Remite
            params.addElement(remite);
            //Realizar la petición al servidor
            XmlRpcClient client = new XmlRpcClient();
            client.setConfig(config);
            Object result =
client.execute("MensajeriaNegocios.enviarAGrupoContacto", params);
            //Imprimir los resultados
            System.out.println("Resultado:"+result);

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

### 3.4.4 Ejemplo PHP Envío a Grupo de Contactos

```
<?php
include ('IXR_Library.inc.php'); //Librería para el uso de xml-rpc

$url_conexion='https://www.mensajerianegocios.movistar.es/SrvConexion';
$login='Login'; //Nombre de usuario
$password='Contraseña'; //Contraseña
$remite='miRemite'; //Remite 217812 o remite de usuario

$client = new IXR_Client($url_conexion);

//Ejecuta el metodo rpc
$client->query ('MensajeríaNegocios.enviarAGrupoContacto', $login, $password, 'Grupo',
'Mensaje a grupo', $remite);

echo '<pre>';
print_r($client->getResponse());
echo '</pre>';
?>
```



### 3.4.5 Ejemplo Java Agregar nuevo Grupo de Contactos

```
import java.net.URL;
import java.util.Vector;
import org.apache.xmlrpc.client.XmlRpcClient;
import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;

public class AddGrupoContactos {
    /**
     * @param args
     */
    public static void main(String[] args) {
        String url_conexion= "https://www.mensajerianegocios.movistar.es/SrvConexion";
        String login= "login"; //Nombre de usuario
        String password= "contrasena"; //Contraseña
        String colectivo="Grupo"; //Nombre de Grupo
        String bienvenida="Texto de bienvenida"; //Texto bienvenida

        try {
            //Conectar con el Servidor
            XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
            config.setServerURL(new URL(url_conexion));
            config.setEncoding("ISO-8859-15");

            //Construir llamada
            Vector params = new Vector();
            //Primer parámetro: Login
            params.addElement(login);
            //Segundo parámetro: Contraseña
            params.addElement(password);
            //Tercer parámetro: Grupo
            params.addElement(colectivo);
            //Cuarto parámetro: Texto de bienvenida
            params.addElement(bienvenida);

            //Realizar la petición al servidor
            XmlRpcClient client = new XmlRpcClient();
            client.setConfig(config);
            Object result =
client.execute("MensajeriaNegocios.agregarGrupoContactos", params);
            //Imprimir los resultados
            System.out.println("Resultado:"+result);

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

### 3.4.6 Ejemplo PHP Agregar nuevo Grupo de Contactos

```
<?php
include ('IXR_Library.inc.php'); //Libreria para el uso de xml-rpc

$url_conexion='https://www.mensajerianegocios.movistar.es/SrvConexion';
$login='Login'; //Nombre de usuario
$password='Contrasena'; //Contraseña

$client = new IXR_Client($url_conexion);

//Ejecuta el metodo rpc
$client->query ('MensajeriaNegocios.agregarGrupoContactos', $login, $password, , 'Grupo',
'Bienvenido al grupo');

echo '<pre>';
print_r($client->getResponse());
echo '</pre>'; ?>
```

### 3.4.7 Ejemplo Java Agregar Contacto a Grupo

```
import java.net.URL;
import java.util.Vector;
import org.apache.xmlrpc.client.XmlRpcClient;
import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;

public class AddContactoaGrupo {
    /**
     * @param args
     */
    public static void main(String[] args) {
        //URL de acceso al servicio
        String url_conexion= "https://www.mensajerianegocios.movistar.es/SrvConexion";
        String login= "login"; //Nombre de usuario
        String password= "contrasena"; //Contraseña
        String colectivo="Grupo"; //Nombre de Grupo
        String movil="609000001";
        try {
            //Conectar con el Servidor
            XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
            config.setServerURL(new URL(url_conexion));
            config.setEncoding("ISO-8859-15");

            //Construir llamada
            Vector params = new Vector();
            //Primer parámetro: Login
            params.addElement(login);
            //Segundo parámetro: Contraseña
            params.addElement(password);
            //Tercer parámetro: Grupo
            params.addElement(colectivo);
            //Cuarto parámetro: Móvil
            params.addElement(movil);

            //Realizar la petición al servidor
            XmlRpcClient client = new XmlRpcClient();
            client.setConfig(config);
            Object result =
client.execute("MensajeriaNegocios.agregarContactoaGrupoContactos",
                params);

            System.out.println("Resultado:"+(Integer)result);

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

### 3.4.8 Ejemplo PHP Agregar Contacto a Grupo

```
<?php
include ('IXR_Library.inc.php'); //Libreria para el uso de xml-rpc

$url_conexion='https://www.mensajerianegocios.movistar.es/SrvConexion';
$login='Login'; //Nombre de usuario
$password='Contrasena'; //Contraseña

$client = new IXR_Client($url_conexion);

//Ejecuta el metodo rpc
$client->query ('MensajeriaNegocios.agregarContactoaGrupoContactos', $login, $password,
'Grupo', '609090909');

//Respuesta de la ejecución
echo '<pre>';
print_r($client->getResponse());
echo '</pre>';
?>
```

### 3.4.9 Ejemplo Java Obtener número de Contactos que tiene un Grupo

```
import java.net.URL;
import java.util.Vector;
import org.apache.xmlrpc.client.XmlRpcClient;
import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;

public class ObtenerNumeroContactos {

    public static void main(String[] args) {
        String url_conexion= "https://www.mensajerianegocios.movistar.es/SrvConexion";
        String login= "login"; //Nombre de usuario
        String password= "contrasena"; //Contraseña
        String colectivo="Grupo"; //Nombre de Grupo
        try {
            //Conectar con el Servidor
            XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
            config.setServerURL(new URL(url_conexion));
            config.setEncoding("ISO-8859-15");
            //Construir llamada
            Vector params = new Vector();
            //Primer parámetro: Login
            params.addElement(login);
            //Segundo parámetro: Contraseña
            params.addElement(password);
            //Tercer parámetro: Grupo
            params.addElement(colectivo);

            //Realizar la petición al servidor
            XmlRpcClient client = new XmlRpcClient();
            client.setConfig(config);
            Object result =
client.execute("MensajeriaNegocios.obtenerNumeroContactosdeGrupoContactos", params);

            System.out.println("Resultado:"+(Integer)result);

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

### 3.4.10 Ejemplo PHP Obtener número de Contactos que tiene un Grupo

```
<?php
include ('IXR_Library.inc.php'); //Libreria para el uso de xml-rpc

$url_conexion='https://www.mensajerianegocios.movistar.es/SrvConexion';
$login='Login'; //Nombre de usuario
$password='Contrasena'; //Contraseña

$client = new IXR_Client($url_conexion);

//Ejecuta el metodo rpc
$client->query ('MensajeriaNegocios.obtenerNumeroContactosdeGrupoContactos', $login,
$password, 'Grupo');

echo '<pre>';
print_r($client->getResponse());
echo '</pre>'; ?>
```

### 3.4.11 Ejemplo Java Obtener números de teléfono de Contactos de un Grupo

```
import java.net.URL;
import java.util.Vector;
import org.apache.xmlrpc.client.XmlRpcClient;
import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;

public class ObtenerContactosGrupo {

    public static void main(String[] args) {
        String url_conexion= "https://www.mensajerianegocios.movistar.es/SrvConexion";
        String login= "login"; //Nombre de usuario
        String password= "contrasena"; //Contraseña
        String colectivo="Grupo"; //Nombre de Grupo

        try {
            //Conectar con el Servidor
            XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
            config.setServerURL(new URL(url_conexion));
            config.setEncoding("ISO-8859-15");
            //Construir llamada
            Vector params = new Vector();
            //Primer parámetro: Login
            params.addElement(login);
            //Segundo parámetro: Contraseña
            params.addElement(password);
            //Tercer parámetro: Grupo
            params.addElement(colectivo);

            //Realizar la petición al servidor
            XmlRpcClient client = new XmlRpcClient();
            client.setConfig(config);
            Object result =
client.execute("MensajeriaNegocios.obtenerContactosdeGrupoContactos", params);
            //Imprimir los resultados
            if(result instanceof Integer) {
                System.out.println(result);
            }else{
                for(int i=0; i<((Object[])result).length;i++){
                    System.out.println(((Object[])result)[i]);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

### 3.4.12 Ejemplo Java Obtener mensaje recibidos en el Buzón de Entrada

```
import java.net.MalformedURLException;
import java.util.Vector;

import org.apache.xmlrpc.XmlRpcException;
import org.apache.xmlrpc.client.XmlRpcClient;
import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;
import org.apache.xmlrpc.client.XmlRpcCommonsTransportFactory;

public class obtenerSMSRecibidos {

    // public static String url =
    // "https://www.mensajerianegocios.movistar.es/SrvConexion";
    public static String url = "http://localhost:8400/mnegocios/SrvConexion";
    public static String login = "user2588";
    public static String password = "ab12cd34";
    public static String metodo = "MensajeriaNegocios.obtenerSMSRecibidos";

    public static String fechaInicial = "2016-01-01 00:00:00";
    public static String fechaFinal = "2017-02-01 00:00:00";

    public static void main(String[] args) {

        System.setProperty("javax.net.ssl.trustStore", "mn2.jks");

        XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
        try {
            config.setServerURL(new java.net.URL(url));
        } catch (MalformedURLException e) {
            System.out
                .println("La URI introducida no tiene una sintaxis
correcta...");
        }
        config.setEncoding("ISO-8859-1");
        XmlRpcClient client = new XmlRpcClient();
        client.setTransportFactory(new XmlRpcCommonsTransportFactory(client));
        client.setConfig(config);

        Vector params = new Vector();

        params.addElement(login);
        params.addElement(password);
        params.addElement(fechaInicial);
        params.addElement(fechaFinal);

        try {
            Object resultado = client.execute(metodo, params);

            if (resultado.getClass().getName()
                .compareToIgnoreCase("java.lang.Integer") == 0) {
                System.out.println("Error: " + resultado);
            } else {
                Object[] result = (Object[]) resultado;

                for (int i = 0; i < result.length; i++) {
                    System.out.println("Numero: " + result[i]);
                }
            }
        }
    }
}
```



```
        } catch (XmlRpcException e) {  
            System.out.println("Se produjo un excepcion: "  
                + e.getLocalizedMessage());  
        }  
    }  
}
```

### 3.4.13 Ejemplo Java Obtener mensaje enviados y su estado

```
import java.net.MalformedURLException;
import java.util.Vector;

import org.apache.xmlrpc.XmlRpcException;
import org.apache.xmlrpc.client.XmlRpcClient;
import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;
import org.apache.xmlrpc.client.XmlRpcCommonsTransportFactory;

public class obtenerSMSEnviados {

    // public static String url =
    // "https://www.mensajerianegocios.movistar.es/SrvConexion";
    public static String url = "http://localhost:8400/mnegocios/SrvConexion";
    public static String login = "user2588";
    public static String password = "ab12cd34";
    public static String metodo = "MensajeriaNegocios.obtenerSMSEnviados";

    public static String fechaInicial = "2017-01-23 00:00:00";
    public static String fechaFinal = "2017-01-24 00:00:00";

    public static void main(String[] args) {

        System.setProperty("javax.net.ssl.trustStore", "mn2.jks");

        XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
        try {
            config.setServerURL(new java.net.URL(url));
        } catch (MalformedURLException e) {
            System.out
                .println("La URI introducida no tiene una sintaxis
correcta...");
        }
        config.setEncoding("ISO-8859-1");
        XmlRpcClient client = new XmlRpcClient();
        client.setTransportFactory(new XmlRpcCommonsTransportFactory(client));
        client.setConfig(config);

        Vector params = new Vector();

        params.addElement(login);
        params.addElement(password);
        params.addElement(fechaInicial);
        params.addElement(fechaFinal);

        try {
            Object resultado = client.execute(metodo, params);

            if (resultado.getClass().getName()
                .compareToIgnoreCase("java.lang.Integer") == 0) {
                System.out.println("Error: " + resultado);
            } else {
                Object[] result = (Object[]) resultado;

                if (result.length > 0) {
                    for (int i = 0; i < result.length; i++) {
                        Object[] sms = (Object[]) result[i];
                        System.out.println("Login: " + sms[0]);
                    }
                }
            }
        }
    }
}
```

```
sms[6];  
  
        } else {  
            System.out.println("No hay mensajes.");  
        }  
    }  
} catch (XmlRpcException e) {  
    System.out.println("Se produjo un excepcion: "  
        + e.getLocalizedMessage());  
}  
}  
}
```

```
System.out.println("Movil: " + sms[1]);  
System.out.println("Texto: " + sms[2]);  
System.out.println("Remitente: " + sms[3]);  
System.out.println("Fecha: " + sms[4]);  
System.out.println("Estado: " + sms[5]);  
System.out.println("Estado smpp: " +
```

#### 3.4.14 Ejemplo PHP Obtener números de teléfono de Contactos de un Grupo

```
<?php
include ('IXR_Library.inc.php'); //Libreria para el uso de xml-rpc

$url_conexion='https://www.mensajerianegocios.movistar.es/SrvConexion';
$login='Login'; //Nombre de usuario
$password='Contrasena'; //Contraseña

$client = new IXR_Client($url_conexion);

//Ejecuta el metodo rpc
$client->query ('MensajeriaNegocios.obtenerContactosdeGrupoContactos', $login, $password,
'Grupo');

echo '<pre>';
print_r($client->getResponse());
echo '</pre>';
?>
```