

Especificaciones de la Interfaz Web Services para envío de SMS

Altiria TIC, S.L.

Versión: 2.0

Copyright © Altiria TIC 2018

Este documento sólo puede ser reproducido por completo o en parte, almacenado, recuperado o transmitido por medios electrónicos, mecánicos, fotocopiado o cualquier otro medio con el consentimiento previo de los autores de acuerdo con los términos que estos indiquen.

Historial de cambios

Versión	Cambios
2.0	<p>Se añade el comando para el envío de múltiples mensajes (sección 2.3.2).</p> <p>Se añaden los códigos de error 018 y 019 (sección 2.7).</p> <p>Se admite la autenticación sin el parámetro “domainId” en caso de que el login sea un email.</p> <p>Se actualizan los ejemplos de programación (sección 2.8).</p>
1.2	<p>Se añade dos nuevos estados de confirmación de entrega: “ERROR_114” y “ERROR_115” (cuadro 2.15).</p> <p>Se actualiza los ejemplos de programación (sección 2.8).</p>
1.1	<p>Se modifica la lista de caracteres válidos para el remitente de los mensajes así como la gestión de los caracteres inválidos en el parámetro “senderId” (sección 2.3.1).</p>
1.0	<p>Primera versión del documento.</p>

Índice general

1. Introducción	3
2. Descripción de la API	4
2.1. Acceso al servicio web	4
2.2. Respuesta del servicio web	4
2.3. Operaciones del servicio	5
2.3.1. Envío de un mensaje de texto	5
2.3.2. Envío de múltiples mensajes de texto	9
2.3.3. Envío de un mensaje multimedia WAP-PUSH	12
2.3.4. Consulta del crédito disponible	15
2.4. Mensajes de texto	18
2.4.1. Codificación por defecto	18
2.4.2. Unicode	18
2.4.3. Longitud del mensaje	19
2.5. Mensajes WAP-PUSH	20
2.5.1. Caracteres permitidos	20
2.5.2. Especificación de la URL	20
2.5.3. Tipos de contenido	21
2.5.4. Formato del contenido	22
2.5.5. Dirección del contenido	22
2.5.6. Envío del contenido	22
2.5.7. Control de acceso al contenido	22
2.6. Confirmación de entrega	24
2.7. Códigos de estado	27
2.8. Ejemplos	28
2.8.1. Envío de un mensaje en PHP	28
2.8.2. Envío de un mensaje en JAVA	29
2.8.3. Envío de un mensaje en .NET	31
2.8.4. Ejemplo en Ruby	34
2.8.5. Ejemplo en Python	35
2.8.6. Ejemplo en Perl	36

Capítulo 1

Introducción

En este documento se presenta la API disponible para el envío de mensajes cortos sobre la interfaz de *Altiria* a través de servicios web.

El servicio de envío de mensajes cortos está disponible en muchos países. Para conocer los países permitidos, las operadoras válidas en cada país y las posibles restricciones geográficas (salvedades al funcionamiento general detallado en este documento que pudieran aplicar en cada caso) se puede enviar un correo electrónico a comercial@altiria.com.

El servicio opcional de confirmación de entrega requiere que el cliente exponga un servicio web para recibir la información de confirmación (ver la sección 2.6).

Capítulo 2

Descripción de la API

2.1. Acceso al servicio web

El servicio web está disponible a través de SOAP 1.1 y de SOAP 1.2

La **URL de acceso** varía según la versión del protocolo:

- **Para SOAP 1.1:** <http://www.altiria.net/api/ws/soap>
- **Para SOAP 1.2:** <http://www.altiria.net/api/ws/soap12>

La **URL del fichero WSDL** es <http://www.altiria.net/api/ws/soap?wsdl>

Se recomienda el uso de la codificación UTF-8 en la comunicación con el servidor.

En cualquier caso los caracteres de la tabla 2.1 deben ser representados de manera especial al comunicarlos al servidor:

Carácter	Representación
Retorno de carro	 o 
Nueva línea	
 o

”	"
’	'
&	&
<	<
>	>

Cuadro 2.1: Caracteres especiales

Como norma general no se deben usar los caracteres prohibidos por la especificación XML para evitar errores de comunicación con el servidor.

2.2. Respuesta del servicio web

Cada operación del servicio web lleva asociada una respuesta desde el servidor de Altiria.

La respuesta a cada operación se remite codificada con el juego de caracteres usado por el cliente al acceder al servicio web.

En los siguientes apartados se detalla la respuesta para cada operación siempre que resulte exitosa. En esos casos el código de estado HTTP será 200.

Si se produjese algún error en el servidor se emitirá una respuesta de tipo “Fault” detallando las causas. En estos casos el código de estado HTTP no será 200 sino 500 y la respuesta no se ajustará a lo especificado en el fichero WSDL.

Un ejemplo de respuesta “Fault” en SOAP 1.1 debido a un error en la petición del cliente (falta el parámetro obligatorio “login”) es el siguiente:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>
        Unmarshalling Error: cvc-complex-type.2.4.a:
        Invalid content was found starting with element 'login'.
        One of '{login}' is expected.
      </faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

El mismo ejemplo de respuesta “Fault” en SOAP 1.2 es el siguiente:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <soap:Fault>
      <soap:Code>
        <soap:Value>soap:Sender</soap:Value>
      </soap:Code>
      <soap:Reason>
        <soap:Text xml:lang="en">
          Unmarshalling Error: cvc-complex-type.2.4.a: Invalid content
          was found starting with element 'login'.
          One of '{login}' is expected.
        </soap:Text>
      </soap:Reason>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

2.3. Operaciones del servicio

A continuación se detallan las operaciones disponibles en el servicio web y los parámetros que las componen según la definición del fichero WSDL. Cada parámetro puede ser obligatorio u opcional y en algunos casos puede aparecer múltiples veces.

2.3.1. Envío de un mensaje de texto

Permite enviar un mensaje corto de texto a uno o a varios teléfonos destinatarios.

Se trata de la operación **sendSms** definida en el fichero WSDL del servicio. Esta compuesta por los **parámetros obligatorios de entrada** detallados en el cuadro 2.2.

Nombre	Valor	Tipo de datos
credentials	Datos de identificación del usuario suministrados por <i>Altiria</i>	Credentials
destination	Número de teléfono móvil del destinatario del mensaje.	String
message	Datos propios del mensaje a enviar	TextMessage

Cuadro 2.2: Parámetros de entrada de la operación sendSms

Para enviar el **mensaje a varios destinatarios** basta repetir el parámetro “destination” tantas veces como sea preciso sin sobrepasar el límite máximo permitido (consultar al soporte técnico de *Altiria* en soporte@altiria.com), asignándole cada vez el valor de un número de teléfono distinto (los teléfonos repetidos son descartados). Se recomienda en cualquier caso no exceder de 100 destinatarios por operación.

En el cuadro 2.3 se detallan los parámetros constituyentes de los tipos de datos complejos Credentials y TextMessage así como el parámetro simple “destination”.

Nombre	Valor	Obligatorio
domainId	Identificador suministrado por <i>Altiria</i> al cliente. Se puede omitir si el login es un email.	no
login	Identificador de usuario suministrado por <i>Altiria</i> al cliente.	sí
passwd	Clave del usuario suministrada por <i>Altiria</i> al cliente.	sí
destination	Número de teléfono móvil del destinatario del mensaje. Se especificará en formato de numeración internacional sin prefijo '00' ni el signo '+'. Ej: 34645852126. Es fundamental incluir el prefijo del país (34 para España) para que el mensaje llegue al destino esperado. No debe superar los 16 dígitos.	sí
msg	Mensaje a enviar. La lista de caracteres válidos y la longitud máxima permitida se detalla en la sección 2.4. No puede estar vacío (cadena vacía).	sí
senderId	Remitente del mensaje a enviar, autorizado por <i>Altiria</i> . La posibilidad de personalizar el remitente depende del país destinatario del mensaje. Puede tomar dos posibles valores: 1) valor alfanumérico de hasta 11 caracteres (números y letras de la “a” a la “z” tanto mayúsculas como minúsculas excluyendo la “Ñ” y la “ñ”); 2) valor numérico de hasta 15 dígitos decimales comenzando por el carácter “+”. Los caracteres inválidos serán suprimidos automáticamente. Si se pretende que el receptor pueda responder al mensaje corto recibido se debería usar un remitente numérico (opción 2) incluyendo el prefijo de país. Si no se incluye, el mensaje se enviará con el remitente por defecto seleccionado por <i>Altiria</i> .	no
ack	Solicitud de confirmación de entrega de los mensajes enviados (ver sección 2.6). Si vale ”true” solicita confirmación de entrega de los SMS enviados. En su ausencia o si tiene otro valor no se solicita confirmación de entrega.	no
idAck	Código identificativo para la confirmación de entrega (ver sección 2.6). Valor alfanumérico de hasta 20 caracteres (números y letras de la “a” a la “z” tanto mayúsculas como minúsculas sin incluir ni “Ñ” ni “ñ”). De rebasar la longitud máxima permitida será truncado. Los caracteres no permitidos serán eliminados. Solo será considerado si el parámetro ack se envía con valor ”true”. Si se incluye explícitamente este parámetro y toma como valor cadena vacía, anula la solicitud de confirmación de entrega.	no

dPort	Puerto destino del SMS a enviar. Valor numérico entre 1 y 65535 (solo dígitos decimales). Solo necesario si se desea cambiar el valor habitual de esta propiedad. Si se define, la longitud máxima del mensaje a enviar (parámetro “msg”) se verá reducida (ver la sección 2.4.3) y se invalidará la posibilidad de enviar mensajes concatenados (ver parámetro “concat”). Si solo se define “sPort”, este tomará el valor 0.	no
sPort	Puerto origen del SMS a enviar. Valor numérico entre 1 y 65535 (solo dígitos decimales). Solo necesario si se desea cambiar el valor habitual de esta propiedad. Si se define, la longitud máxima del mensaje a enviar (parámetro “msg”) se verá reducida (ver la sección 2.4.3) y se invalidará la posibilidad de enviar mensajes concatenados (ver parámetro “concat”). Si solo se define “dPort”, este tomará el valor 0.	no
encoding	El único valor permitido es “unicode” para cambiar la codificación del SMS a Unicode (ver la sección 2.4.2). En su ausencia o si tiene otro valor el SMS tomará la codificación por defecto.	no
concat	Si vale “true” permite concatenar mensajes para enviar un mensaje corto de longitud mayor que la habitual (ver la sección 2.4.3). En su ausencia, si tiene otro valor o si se define el parámetro “sPort” o “dPort” se deshabilita la concatenación de mensajes.	no

Cuadro 2.3: Lista de parámetros para sendsms

De acuerdo al **fichero WSDL**, un **ejemplo de mensaje SOAP** correspondiente a la operación **sendSms**, solicitando el envío de un mensaje concatenado a dos destinatarios sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <TextMessageRequest>
      <credentials>
        <domainId>XXXXX</domainId>
        <login>YYYYY</login>
        <passwd>ZZZZZ</passwd>
      </credentials>
      <destination>346XXXXXXXX</destination>
      <destination>346YYYYYYYY</destination>
      <message>
        <msg>
          Ejemplo de mensaje concatenado enviado a más de un destinatario con la
          codificación UNICODE para admitir las vocales acentuadas y solicitud de
          confirmación de entrega.
        </msg>
        <senderId>remitente</senderId>
        <ack>>true</ack>
        <idAck>123456789</idAck>
        <encoding>unicode</encoding>
        <concat>>true</concat>
      </message>
    </TextMessageRequest>
  </soap:Body>
</soap:Envelope>
```

Según el **fichero WSDL**, la **respuesta de la operación sendSms** está compuesta por un código de estado general de los descritos en apartado 2.7

Si la operación ha resultado exitosa (código de estado “000”) la respuesta contendrá adicionalmente para cada destinatario del envío (parámetros “destination” de entrada) los siguientes datos:

- destination: se corresponde con el número de teléfono del destinatario. Si se hubiese enviado un mensaje concatenado (ver el parámetro “concat”) a un único destinatario le corresponderán varios mensajes, tantos como fragmentos compongan el mensaje concatenado. En ese caso aparecerán datos independientes para cada fragmento siendo cualificado el valor de “destination” con un sufijo que diferencie cada fragmento con un índice numérico comenzando por 0. Por ejemplo para un mensaje concatenado de tres fragmentos enviado al número “xxxxxxxxxxx” se recibirán datos para destination=xxxxxxxxxx(0), destination=xxxxxxxxxx(1) y destination=xxxxxxxxxx(2).
- status: se corresponde con uno de los códigos de estado del apartado 2.7.
- idAck: se corresponde con el código de identificación asociado a la solicitud de confirmación de entrega (ver sección 2.6). Solo aparecerá si se solicita la confirmación de entrega y es aceptada.

Un ejemplo de respuesta exitosa correspondiente a la operación **sendSms** del ejemplo anterior sería el mostrado a continuación. Al tratarse de un mensaje concatenado aparecen datos para cada uno de los tres fragmentos que lo componen para cada destinatario del envío:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <TextMessageResponse>
      <status>000</status>
      <details>
        <destination>346XXXXXXXX(0)</destination>
        <status>000</status>
        <idAck>123456789</idAck>
      </details>
      <details>
        <destination>346XXXXXXXX(1)</destination>
        <status>000</status>
        <idAck>123456789</idAck>
      </details>
      <details>
        <destination>346XXXXXXXX(2)</destination>
        <status>000</status>
        <idAck>123456789</idAck>
      </details>
      <details>
        <destination>346YYYYYYYY(0)</destination>
        <status>000</status>
        <idAck>123456789</idAck>
      </details>
      <details>
        <destination>346YYYYYYYY(1)</destination>
        <status>000</status>
        <idAck>123456789</idAck>
      </details>
      <details>
        <destination>346YYYYYYYY(2)</destination>
        <status>000</status>
        <idAck>123456789</idAck>
      </details>
    </TextMessageResponse>
  </soap:Body>
</soap:Envelope>
```

```
</TextMessageResponse>
</soap:Body>
</soap:Envelope>
```

Un ejemplo de respuesta notificando un error en la autenticación sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <TextMessageResponse>
      <status>020</status>
    </TextMessageResponse>
  </soap:Body>
</soap:Envelope>
```

La infomación de éxito para un destinatario concreto implica que el mensaje ha sido aceptado por la pasarela, no que haya sido enviado y recibido por el destinatario. Un mensaje puede ser aceptado aún cuando no se disponga de crédito suficiente para su envío (ver sección 2.3.4).

Para asegurar el adecuado funcionamiento de esta operación se recomienda probar la correcta recepción de todos los caracteres permitidos en un teléfono móvil antes de poner el sistema en producción.

2.3.2. Envío de múltiples mensajes de texto

Permite enviar una lista de mensajes cortos de texto, cada cual al destinatario indicado.

Se trata de la operación **sendSmsMulti** definida en el fichero WSDL del servicio. Esta compuesta por los **parámetros obligatorios de entrada** detallados en el cuadro 2.4.

Nombre	Valor	Tipo de datos
credentials	Datos de identificación del usuario suministrados por <i>Altiria</i>	Credentials
messages	Datos propios del mensaje a enviar	TextMessageMulti

Cuadro 2.4: Parámetros de entrada de la operación sendSmsMulti

Para enviar más de un mensaje basta repetir el parámetro “messages” tantas veces como sea preciso sin sobrepasar el límite máximo permitido (consultar al soporte técnico de *Altiria* en soporte@altiria.com), asignando cada vez los valores propios de cada mensaje.

En el cuadro 2.5 se detallan los parámetros constituyentes de los tipos de datos complejos Credentials y TextMessageMulti.

Nombre	Valor	Obligatorio
domainId	Identificador suministrado por <i>Altiria</i> al cliente. Se puede omitir si el login es un email.	no
login	Identificador de usuario suministrado por <i>Altiria</i> al cliente.	sí
passwd	Clave del usuario suministrada por <i>Altiria</i> al cliente.	sí
destination	Número de teléfono móvil del destinatario del mensaje. Se especificará en formato de numeración internacional sin prefijo '00' ni el signo '+'. Ej: 34645852126. Es fundamental incluir el prefijo del país (34 para España) para que el mensaje llegue al destino esperado. No debe superar los 16 dígitos.	sí
msg	Mensaje a enviar. La lista de caracteres válidos y la longitud máxima permitida se detalla en la sección 2.4. No puede estar vacío (cadena vacía).	sí

senderId	Remitente del mensaje a enviar, autorizado por <i>Altiria</i> . La posibilidad de personalizar el remitente depende del país destinatario del mensaje. Puede tomar dos posibles valores: 1) valor alfanumérico de hasta 11 caracteres (números y letras de la “a” a la “z” tanto mayúsculas como minúsculas excluyendo la “Ñ” y la “ñ”); 2) valor numérico de hasta 15 dígitos decimales comenzando por el carácter “+”. Los caracteres inválidos serán suprimidos automáticamente. Si se pretende que el receptor pueda responder al mensaje corto recibido se debería usar un remitente numérico (opción 2) incluyendo el prefijo de país. Si no se incluye, el mensaje se enviará con el remitente por defecto seleccionado por Altiria.	no
idMsg	Referencia del mensaje para facilitar su identificación al procesar la respuesta a la petición de envío múltiple.	no
ack	Solicitud de confirmación de entrega de los mensajes enviados (ver sección 2.6). Si vale ”true” solicita confirmación de entrega de los SMS enviados. En su ausencia o si tiene otro valor no se solicita confirmación de entrega.	no
idAck	Código identificativo para la confirmación de entrega (ver sección 2.6). Valor alfanumérico de hasta 20 caracteres (números y letras de la “a” a la “z” tanto mayúsculas como minúsculas sin incluir ni “Ñ” ni “ñ”). De rebasar la longitud máxima permitida será truncado. Los caracteres no permitidos serán eliminados. Solo será considerado si el parámetro ack se envía con valor ”true”. Si se incluye explícitamente este parámetro y toma como valor cadena vacía, anula la solicitud de confirmación de entrega.	no
dPort	Puerto destino del SMS a enviar. Valor numérico entre 1 y 65535 (solo dígitos decimales). Solo necesario si se desea cambiar el valor habitual de esta propiedad. Si se define, la longitud máxima del mensaje a enviar (parámetro “msg”) se verá reducida (ver la sección 2.4.3) y se invalidará la posibilidad de enviar mensajes concatenados (ver parámetro “concat”). Si solo se define “sPort”, este tomará el valor 0.	no
sPort	Puerto origen del SMS a enviar. Valor numérico entre 1 y 65535 (solo dígitos decimales). Solo necesario si se desea cambiar el valor habitual de esta propiedad. Si se define, la longitud máxima del mensaje a enviar (parámetro “msg”) se verá reducida (ver la sección 2.4.3) y se invalidará la posibilidad de enviar mensajes concatenados (ver parámetro “concat”). Si solo se define “dPort”, este tomará el valor 0.	no
encoding	El único valor permitido es “unicode” para cambiar la codificación del SMS a Unicode (ver la sección 2.4.2). En su ausencia o si tiene otro valor el SMS tomará la codificación por defecto.	no
concat	Si vale “true” permite concatenar mensajes para enviar un mensaje corto de longitud mayor que la habitual (ver la sección 2.4.3). En su ausencia, si tiene otro valor o si se define el parámetro “sPort” o “dPort” se deshabilita la concatenación de mensajes.	no

Cuadro 2.5: Lista de parámetros para sendsmsmulti

De acuerdo al fichero WSDL, un ejemplo de mensaje SOAP correspondiente a la operación `sendSmsMulti`, solicitando el envío de tres mensajes sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <TextMessagesRequest xmlns="http://www.altiria.net/api/soap">
    <credentials>
```

```

    <domainId>XXXXX</domainId>
    <login>YYYYY</login>
    <passwd>ZZZZZ</passwd>
  </credentials>
  <messages>
    <msg>Mensaje de prueba 1</msg>
    <destination>346XXXXXXXX</destination>
  </messages>
  <messages>
    <msg>
      Lorem Ipsum es simplemente el texto de relleno de las imprentas
      y archivos de texto. Lorem Ipsum ha sido el texto de relleno
      estandar de las industrias desde el año 1500
    </msg>
    <destination>346YYYYYYYY</destination>
    <concat>true</concat>
    <idMsg>id2</idMsg>
  </messages>
  <messages>
    <msg>Mensaje de prueba 3</msg>
    <destination>346ZZZZZZZZ</destination>
    <senderId>remitente</senderId>
    <ack>true</ack>
    <idAck>123456789</idAck>
    <dPort>5000</dPort>
    <sPort>4000</sPort>
    <idMsg>id3</idMsg>
  </messages>
</TextMessagesRequest>
</soap:Body>
</soap:Envelope>

```

Según el **fichero WSDL**, la **respuesta de la operación sendSmsMulti** está compuesta por un código de estado general de los descritos en apartado 2.7

Si la operación ha resultado exitosa (código de estado “000”) la respuesta contendrá adicionalmente para cada mensaje del envío (parámetros “messages” de entrada) los siguientes datos:

- destination: se corresponde con el número de teléfono del destinatario. Si se hubiese enviado un mensaje concatenado (ver el parámetro “concat”) al destinatario le corresponderán varios mensajes, tantos como fragmentos compongan el mensaje concatenado. En ese caso aparecerán datos independientes para cada fragmento siendo cualificado el valor de “destination” con un sufijo que diferencie cada fragmento con un índice numérico comenzando por 0. Por ejemplo para un mensaje concatenado de tres fragmentos enviado al número “xxxxxxxxxxx” se recibirán datos para destination=xxxxxxxxxxx(0), destination=xxxxxxxxxxx(1) y destination=xxxxxxxxxxx(2).
- status: se corresponde con uno de los códigos de estado del apartado 2.7.
- idAck: se corresponde con el código de identificación asociado a la solicitud de confirmación de entrega (ver sección 2.6). Solo aparecerá si se solicita la confirmación de entrega y es aceptada.
- idMsg: se corresponde con la referencia asociada al mensaje remitida en la petición de envío. Solo aparecerá si se ha incluido en la petición al servidor.

Un ejemplo de respuesta exitosa correspondiente a la operación **sendSmsMulti** del ejemplo anterior sería el mostrado a continuación. Al ser el segundo mensaje concatenado aparecen datos para cada uno de los dos fragmentos que lo componen:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<TextMessageResponse xmlns="http://www.altiria.net/api/soap">
  <status>000</status>
  <details>
    <destination>346XXXXXXXX</destination>
    <status>000</status>
  </details>
  <details>
    <destination>346YYYYYYYY(0)</destination>
    <status>000</status>
    <idMsg>id2</idMsg>
  </details>
  <details>
    <destination>346YYYYYYYY(1)</destination>
    <status>000</status>
    <idMsg>id2</idMsg>
  </details>
  <details>
    <destination>346ZZZZZZZZ</destination>
    <status>000</status>
    <idAck>123456789</idAck>
    <idMsg>id3</idMsg>
  </details>
</TextMessageResponse>
</soap:Body>
</soap:Envelope>
```

Un ejemplo de respuesta notificando un error en la autenticación sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<TextMessageResponse xmlns="http://www.altiria.net/api/soap">
  <status>020</status>
</TextMessageResponse>
</soap:Body>
</soap:Envelope>
```

La información de éxito para un mensaje concreto implica que el mensaje ha sido aceptado por la pasarela, no que haya sido enviado y recibido por el destinatario. Un mensaje puede ser aceptado aún cuando no se disponga de crédito suficiente para su envío (ver sección 2.3.4).

Para asegurar el adecuado funcionamiento de esta operación se recomienda probar la correcta recepción de todos los caracteres permitidos en un teléfono móvil antes de poner el sistema en producción.

2.3.3. Envío de un mensaje multimedia WAP-PUSH

Permite enviar un mensaje multimedia a través de WAP-PUSH a uno o a varios teléfonos destinatarios.

Para conocer en qué consisten los mensajes de este tipo se aconseja leer la sección 2.5.

Para saber más sobre el proceso de envío de mensajes WAP-PUSH a través de la pasarela se aconseja leer la sección 2.5.7.

Se trata de la operación **sendWapPush** definida en el fichero WSDL del servicio. Está compuesta por los **parámetros obligatorios de entrada** detallados en el cuadro 2.6.

Nombre	Valor	Tipo de datos
credentials	Datos de identificación del usuario suministrados por <i>Altiria</i>	Credentials
destination	Número de teléfono móvil del destinatario del mensaje.	String
message	Datos propios del mensaje a enviar	WapPushMessage

Cuadro 2.6: Parámetros de entrada de la operación sendWapPush

Para enviar el **mensaje a varios destinatarios** basta repetir el parámetro “destination” tantas veces como sea preciso sin sobrepasar el límite máximo permitido (consultar al soporte técnico de *Altiria* en soporte@altiria.com), asignándole cada vez el valor de un número de teléfono distinto (los teléfonos repetidos son descartados). Se recomienda en cualquier caso no exceder de 100 destinatarios por operación.

En el cuadro 2.7 se detallan los parámetros constituyentes de los tipos de datos complejos Credentials y WapPushMessage así como el parámetro simple “destination”.

Nombre	Valor	Obligatorio
domainId	Identificador suministrado por <i>Altiria</i> al cliente. Se puede omitir si el login es un email.	no
login	Identificador de usuario suministrado por <i>Altiria</i> al cliente.	sí
passwd	Clave del usuario suministrada por <i>Altiria</i> al cliente.	sí
destination	Número de teléfono móvil del destinatario del mensaje. Se especificará en formato de numeración internacional sin prefijo '00' ni el signo '+'. Ej: 34645852126. Es fundamental incluir el prefijo del país (34 para España) para que el mensaje llegue al destino esperado. No debe superar los 16 dígitos.	sí
msg	Texto a enviar adjunto al mensaje WAP-PUSH. Representa una breve descripción del contenido multimedia. Debería contener exclusivamente caracteres incluidos en la lista del apartado 2.5.1. No debe sobrepasar los 115 caracteres junto a la longitud del parámetro “url” o bien los 88 caracteres si se opta por el envío de clave en el parametro “url” (ver sección 2.5.7). No puede estar vacío.	sí
url	Dirección de Internet desde donde el teléfono móvil se descargará el contenido. Debe contener caracteres validos en una URL (ver sección 2.5.2). No debe sobrepasar los 115 caracteres junto a la longitud del parámetro “msg” o bien los 88 caracteres si se opta por el envío de clave en este parametro (ver sección 2.5.7). No puede estar vacío. No es posible seleccionar un puerto de conexión diferente al 80, el habitual en la navegación WEB.	sí
ack	Solicitud de confirmación de entrega de los mensajes enviados (ver sección 2.6). Si vale ”true” solicita confirmación de entrega de los SMS enviados. En su ausencia o si tiene otro valor no se solicita confirmación de entrega.	no
idAck	Código identificativo para la confirmación de entrega (ver sección 2.6). Valor alfanumérico de hasta 20 caracteres (números y letras de la “a” a la “z” tanto mayúsculas como minúsculas sin incluir ni “Ñ” ni “ñ”). De rebasar la longitud máxima permitida será truncado. Los caracteres no permitidos serán eliminados. Solo será considerado si el parámetro ack se envía con valor ”true”. Si se incluye explícitamente este parámetro y toma como valor cadena vacía, anula la solicitud de confirmación de entrega.	no

Cuadro 2.7: Lista de parámetros para sendwappush

De acuerdo al **fichero WSDL**, un **ejemplo de mensaje SOAP** correspondiente a la operación **sendWapPush** sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <WapPushMessageRequest>
      <credentials>
        <domainId>XXXXX</domainId>
        <login>YYYYYY</login>
        <passwd>ZZZZZZ</passwd>
      </credentials>
      <destination>346XXXXXXXX</destination>
      <destination>346YYYYYYYY</destination>
      <message>
        <msg>Texto de alerta</msg>
        <url>http://www.altiria.com?k=</url>
        <ack>true</ack>
        <idAck>123456789</idAck>
      </message>
    </WapPushMessageRequest>
  </soap:Body>
</soap:Envelope>
```

Según el **fichero WSDL**, la **respuesta de la operación sendWapPush** está compuesta por un código de estado general de los descritos en apartado 2.7

Si la operación ha resultado exitosa (código de estado “000”) la respuesta contendrá adicionalmente para cada destinatario del envío (parámetros “destination” de entrada) los siguientes datos:

- destination: se corresponde con el número de teléfono del destinatario.
- status: se corresponde con uno de los códigos de estado del apartado 2.7.
- idAck: se corresponde con el código de identificación asociado a la solicitud de confirmación de entrega (ver sección 2.6). Solo aparecerá si se solicita la confirmación de entrega y es aceptada.
- key: se corresponde con la clave única asociada al destinatario (ver sección 2.5.7). Solo aparecerá si se solicita la clave para cada destinatario

Un ejemplo de respuesta exitosa correspondiente a la operación **sendWapPush** del ejemplo anterior sería el mostrado a continuación:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <WapPushMessageResponse>
      <status>000</status>
      <details>
        <destination>346XXXXXXXX</destination>
        <status>000</status>
        <idAck>123456789</idAck>
        <key>346XXXXXXXX-22879196</key>
      </details>
      <details>
        <destination>346YYYYYYYY</destination>
        <status>000</status>
        <idAck>123456789</idAck>
```

```
<key>346YYYYYYYY-22879196</key>
</details>
</WapPushMessageResponse>
</soap:Body>
</soap:Envelope>
```

Un ejemplo de respuesta notificando un error en la autenticación sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<TextMessageResponse>
<status>020</status>
</TextMessageResponse>
</soap:Body>
</soap:Envelope>
```

La infomación de éxito para un destinatario concreto implica que el mensaje ha sido aceptado por la pasarela, no que haya sido enviado y recibido por el destinatario. Un mensaje puede ser aceptado aún cuando no se disponga de crédito suficiente para su envío (ver sección 2.3.4).

Para asegurar el adecuado funcionamiento de esta operación se recomienda probar un ciclo completo de servicio, desde el envío del mensaje WAP-PUSH hasta la descarga del contenido en el teléfono móvil, como paso previo a poner el sistema en producción.

2.3.4. Consulta del crédito disponible

Permite conocer el crédito instantáneo disponible para enviar mensajes.

Se trata de la operación **getCredit** definida en el fichero WSDL del servicio. Está compuesta por los **parámetros obligatorios de entrada** detallados en el cuadro 2.8.

Nombre	Valor	Tipo de datos
credentials	Datos de identificación del usuario suministrados por <i>Altiria</i>	Credentials

Cuadro 2.8: Parámetros de entrada de la operación getCredit

En el cuadro 2.9 se detallan los parámetros constituyentes del tipo de datos complejo Credentials.

La única forma de averiguar si se tiene crédito suficiente para enviar los mensajes, aparte de llevar un contador propio de saldo disponible, es mediante una consulta previa a través de esta operación.

La operación ofrece información del crédito disponible en un momento dado. Puesto que el sistema decrementa el crédito justo al enviar el mensaje al destinatario, es necesario seguir el siguiente esquema para utilizar adecuadamente la operación de consulta de crédito disponible:

- Antes de comenzar con los envíos, se calcula cuantos mensajes en total se desean enviar, por ejemplo 5000.
- Se consulta el valor del crédito disponible una única vez.
- A partir del coste en créditos de cada mensaje a enviar y del saldo disponible se estima si se podrán enviar o no todos los mensajes.
- En caso positivo, se usan las operaciones de envío de mensajes. En caso negativo se debe adquirir más crédito

Cuando el sistema haya finalmente enviado todos los mensajes, una nueva consulta del crédito disponible ofrecerá el valor actualizado.

En cualquier caso la comprobación efectiva del saldo disponible para efectuar un envío se realiza en un proceso interno justo antes de efectuar el envío. En caso de que no se disponga de crédito suficiente, el mensaje no será enviado y el cliente será informado a través de correo electrónico. Si posteriormente se adquiere más crédito disponible, se podrá avisar a *Altiria* para reintentar los envíos pendientes.

Nombre	Valor	Obligatorio
domainId	Identificador suministrado por <i>Altiria</i> al cliente. Se puede omitir si el login es un email.	no
login	Identificador de usuario suministrado por <i>Altiria</i> al cliente.	sí
passwd	Clave del usuario suministrada por <i>Altiria</i> al cliente.	sí

Cuadro 2.9: Lista de parámetros para getcredit

De acuerdo al **fichero WSDL**, un ejemplo de mensaje **SOAP** correspondiente a la operación **getCredit** sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreditRequest>
      <credentials>
        <domainId>XXXXXX</domainId>
        <login>YYYYYY</login>
        <passwd>ZZZZZZ</passwd>
      </credentials>
    </CreditRequest>
  </soap:Body>
</soap:Envelope>
```

Según el **fichero WSDL**, la respuesta de la operación **getCredit** está compuesta por un código de estado general de los descritos en apartado 2.7.

Si la operación ha resultado exitosa (código de estado “000”) la respuesta contendrá adicionalmente el valor del crédito disponible como un número con dos decimales.

Un ejemplo de respuesta exitosa correspondiente a la operación **getCredit** del ejemplo anterior sería el mostrado a continuación:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreditResponse>
      <status>000</status>
      <credit>1000000.70</credit>
    </CreditResponse>
  </soap:Body>
</soap:Envelope>
```

Un ejemplo de respuesta notificando un error en la autenticación sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <TextMessageResponse>  
      <status>020</status>  
    </TextMessageResponse>  
  </soap:Body>  
</soap:Envelope>
```

2.4. Mensajes de texto

Los **caracteres permitidos** para el texto del mensaje corto y la **longitud máxima** dependerán de la codificación de caracteres seleccionada: codificación por defecto (ver sección 2.4.1) o la codificación UNICODE (ver sección 2.4.2).

2.4.1. Codificación por defecto

La **codificación por defecto** permite los caracteres de la tabla 2.10.

La **longitud máxima permitida** se detalla en la sección 2.4.3.

Las vocales con tilde o acento agudo (á) son aceptadas pero se enviarán al teléfono móvil sin acentuar.

Adicionalmente se admiten los **caracteres extendidos** de la tabla 2.11. Cada carácter extendido **ocupa el doble espacio que un carácter normal**, esto debe considerarse para el cómputo de la longitud máxima del mensaje.

En caso de que el mensaje a enviar contenga **caracteres fuera de las listas** presentadas, estos serán **reemplazados por el carácter “?”** antes de enviar el mensaje.

@	(4	-	L	W	h	s	Ú	ù
cr ¹)	5	A	M	X	i	t	á	
lf ²	*	6	B	N	Y	j	u	é	
Ç	+	7	C	Ñ	Z	k	v	í	
sp ³	,	8	D	O	¿	l	w	ó	
!	-	9	E	P	a	m	x	ú	
”	.	:	F	Q	b	n	y	Û	
#	/	;	G	R	c	ñ	z	ü	
\$	0	<	H	S	d	o	Á	à	
%	1	=	I	T	e	p	É	è	
&	2	>	J	U	f	q	Í	ì	
'	3	?	K	V	g	r	Ó	ò	

Cuadro 2.10: Lista de caracteres permitidos para mensajes de texto en la codificación por defecto

[]	\	^	{	}		~	€
---	---	---	---	---	---	--	---	---

Cuadro 2.11: Lista de caracteres extendidos permitidos para mensajes de texto

2.4.2. Unicode

La **codificación UNICODE**, forzada mediante el parámetro “unicode” (ver el cuadro 2.3), permite todo el juego de caracteres UNICODE de 16bits.

La **longitud máxima permitida** se detalla en la sección 2.4.3, siendo siempre menor que usando la codificación por defecto (ver la sección 2.4.1).

Con esta codificación sería posible por ejemplo el envío de vocales con tilde.

¹Retorno de carro

²Nueva línea

³Espacio blanco

2.4.3. Longitud del mensaje

La longitud máxima de un mensaje de texto es un valor variable que depende de la codificación de caracteres usada y de la posibilidad de concatenación. Los mensajes que **excedan la longitud máxima aplicable serán rechazados** (no enviados).

- La longitud máxima de un mensaje corto con la **codificación por defecto es de 160 caracteres** (ver sección 2.4.1). Los caracteres extendidos (ver la tabla 2.11) ocupan el doble, por tanto la longitud máxima se reduce. Por ejemplo si el texto del SMS contuviera el símbolo del euro “€” y los corchetes “[]”, la longitud máxima del mensaje corto se reduciría a 157 caracteres.
- La longitud máxima de un mensaje corto con la **codificación UNICODE es de 70 caracteres** (ver sección 2.4.2).

En caso de **definir puertos origen o destino** del SMS (ver los parámetros sPort y dPort en el cuadro 2.3) la **longitud máxima se reduce** de la siguiente forma:

- **152 caracteres para la codificación por defecto** (ver sección 2.4.1). Igualmente hay que considerar que los caracteres extendidos (ver la tabla 2.11) ocupan el doble.
- **66 caracteres para la codificación UNICODE** (ver sección 2.4.2).

Mediante el uso de **mensajes concatenados es posible ampliar esos límites**. Un mensaje concatenado consiste en varios mensajes en secuencia recibidos como un único mensaje en el teléfono del destinatario.

Los mensajes concatenados se posibilitan mediante el parámetro “concat” (ver el cuadro 2.3) siempre que no se estén definiendo ni el puerto origen ni el puerto destino del SMS (ver los parámetros sPort y dPort en el cuadro 2.3).

La plataforma de *Altiria* permite **concatenar hasta 10 mensajes**, aplicando en ese caso los límites siguientes a la longitud del mensaje:

- **1530 caracteres para la codificación por defecto** (ver sección 2.4.1). Igualmente hay que considerar que los caracteres extendidos (ver la tabla 2.11) ocupan el doble.
- **670 caracteres para la codificación UNICODE** (ver sección 2.4.2).

2.5. Mensajes WAP-PUSH

La interfaz HTTP de *Altiria* permite el envío de mensajes multimedia (imágenes, sonidos, juegos, etc) mediante la tecnología de los mensajes WAP-PUSH.

Los mensajes WAP-PUSH incluyen información sobre la ubicación de un determinado contenido multimedia, una dirección de Internet. En este sentido son completamente diferentes a los mensajes de texto normales, puesto que en estos el contenido relevante es el propio texto.

Básicamente un mensaje de este tipo se compone de un pequeño texto a modo de presentación del contenido que se ofrece y la dirección de Internet donde se ubica dicho contenido.

Cuando un teléfono móvil recibe un mensaje WAP-PUSH, le presenta al usuario la breve descripción mencionada junto con la posibilidad de descargarse el contenido multimedia referenciado. Si el usuario acepta, el teléfono de manera automática accede al contenido a través de HTTP, se lo descarga como si de un navegador WEB se tratara y lo almacena, mostrando además otras opciones en función del tipo de contenido (ver una imagen, reproducir un sonido...).

2.5.1. Caracteres permitidos

El cuadro 2.12 detalla los caracteres admisibles para los mensajes WAP-PUSH (parámetro “msg” del cuadro 2.7).

cr ¹	If ²	sp ³	!	”	#	&
,	()	*	+	,	-
.	/	0	1	2	3	4
5	6	7	8	9	:	;
<	=	>	?	A	B	C
D	E	F	G	H	I	J
K	L	M	N	O	P	Q
R	S	T	U	V	W	X
Y	Z	a	b	c	d	e
f	g	h	i	j	k	l
m	n	o	p	q	r	s
t	u	v	w	x	y	z
Á	É	Í	Ó	Ú	á	é
í	ó	ú				

Cuadro 2.12: Lista de caracteres permitidos para los mensajes WAP-PUSH

Las vocales con tilde o acento agudo (á) son aceptadas pero se enviarán al teléfono móvil sin acentuar.

En caso de que el mensaje a enviar contenga caracteres fuera de la lista presentada, estos serán reemplazados por el carácter “?” y el mensaje será enviado.

2.5.2. Especificación de la URL

La URL de descarga de los contenidos multimedia suministrados a través de mensajes WAP-PUSH (parámetro “url” del cuadro 2.7) debe seguir las siguientes normas de composición:

- Los caracteres del cuadro 2.13 son seguros y se pueden incluir sin codificar.

¹Retorno de carro

²Nueva línea

³Espacio blanco

- Los caracteres del cuadro 2.14 son reservados y se pueden incluir sin codificar si se emplean dentro de la URL de acuerdo a su uso reservado. Por ejemplo el carácter “&” se usa para separar los parametros de un formulario. Si estos caracteres se emplean de otro modo se deben codificar.
- Otros caracteres se pueden incluir previa codificación. De todos modos pueden no ser seguros y es posible que algunos presenten problemas en algunos teléfonos. Se recomienda prescindir de ellos siempre que sea posible.

a	b	c	d	e	f	g	h	i	j	k	l	m	n
o	p	q	r	s	t	u	v	w	x	y	z	A	B
C	D	E	F	G	H	I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3
4	5	6	7	8	9	-	-	.	!	*	'	()

Cuadro 2.13: Lista de caracteres seguros

\$	&	+	,	/	:	;	=	?	@
----	---	---	---	---	---	---	---	---	---

Cuadro 2.14: Lista de caracteres reservados

La codificación de un carácter se logra a partir de su representación en hexadecimal en un determinado juego de caracteres, insertando el simbolo “%” por cada par de dígitos hexadecimales. Por ejemplo la “ñ” en UTF-8 se codificaría como “%C3%B1”.

El juego de caracteres a escoger debería ser el del servidor que albergue el contenido a descargar mediante el mensaje WAP-PUSH.

Según lo visto si se desea permitir la descarga de un contenido de la URL:

`http://www.miempresa.com/contenidos/imagen[1].jpg`

se debe enviar como (usando ISO8859-1):

`http://www.miempresa.com/contenidos/imagen%5B1%5D.jpg`

Es importante reseñar que cada carácter codificado ocupa un número mayor de caracteres en el cómputo de la longitud completa de la URL.

En cualquier caso se recomienda probar la correcta descarga de los contenidos desde la URL seleccionada para comprobar que todo el proceso se efectúa correctamente.

2.5.3. Tipos de contenido

El contenido más general que se puede enviar es una página “wml”. Las páginas “wml” son similares a las conocidas páginas web, adaptadas a los requisitos de un teléfono móvil.

De este modo se podrá enviar un contenido formado por texto y otros tipos de archivos como imágenes (ej: jpg y gif) o sonidos (ej: midi), incluidos en la propia página.

También es posible enviar directamente el archivo multimedia al teléfono, evitando incluirlo en una página “wml”.

Actualmente hay mucha diversidad de teléfonos móviles, cada uno con sus propias capacidades multimedia. Es posible que determinados teléfonos no sean capaces de manejar algunos tipos de archivos. Para esas situaciones, la posibilidad de incluir texto en una página “wml”, un recurso manejado por

todos los terminales WAP, permite al menos que el teléfono acceda a parte de la información. A este respecto una buena práctica es incluir en el texto información para el destinatario sobre la opción de acceder al fichero multimedia a través de un navegador web convencional, adjuntando la información relativa a la dirección de Internet.

En caso de optar por la inclusión de texto en una página “wml” se recomienda emplear un juego de caracteres sencillo, reconocible por la mayoría de los teléfonos. Como referencia se puede usar el detallado en el cuadro 2.12, sin incluir las vocales acentuadas.

2.5.4. Formato del contenido

Independientemente del tipo de contenido escogido, siempre se debería considerar que el medio habitual de acceso al mismo será un teléfono móvil.

Esto tiene importantes incidencias en cuanto al tamaño máximo de la información suministrada. Se recomienda no enviar contenidos que ocupen más de 10kB, sobre todo si se suministran embebidos en páginas “wml”.

Para optimizar el tamaño, se sugiere adaptar los contenidos a los requerimientos de un teléfono móvil. Por ejemplo si se trata de una imagen es conveniente ajustar su tamaño al habitual de la pantalla, guardando además una relación de aspecto adecuada para que al recibirla ocupe el máximo en todas las direcciones. Una buena medida como referencia pueden ser 240 x 240 “pixels”.

2.5.5. Dirección del contenido

El teléfono móvil conoce la ubicación del contenido multimedia mediante la información de dirección que le llega en el mensaje WAP-PUSH.

Es obvio que para que el teléfono pueda descargarse la información la dirección debe respresentar la ubicación de un recurso accesible públicamente a través de HTTP, mediante navegación WEB.

Un detalle importante asociado a la dirección del contenido es que muchos teléfonos la emplean como identificador de los mensajes WAP-PUSH recibidos. Esto supone que si se recibe un mensaje WAP-PUSH con la misma dirección del contenido asociado que un mensaje ya recibido y almacenado en el teléfono, el nuevo mensaje reemplazará al antiguo.

Existen sin embargo teléfonos que no siguen este patrón y almacenan los dos mensajes con idéntica dirección del contenido de forma independiente.

2.5.6. Envío del contenido

Cuando el teléfono móvil solicita el contenido referenciado en el mensaje WAP-PUSH, envía una petición HTTP GET (en algunos casos se envía un HTTP HEAD previamente) a la dirección apropiada.

Es necesario entonces un servidor HTTP que atienda la petición y entregue el contenido apropiadamente.

2.5.7. Control de acceso al contenido

La URL indicada en el campo “url” de la operación “sendWapPush” será enviada en el mensaje WAP-PUSH a cada destinatario para que los interesados se descarguen el contenido ahí alojado. Para poder distinguir qué destinatarios han accedido realmente al contenido *Altiria* ofrece la posibilidad de enviar un mensaje diferente a cada uno de ellos. A la URL a suministrar al teléfono se le añadirá un parámetro identificador, una clave. Este parámetro estará unívocamente asociado al teléfono del destinatario mediante la respuesta generada por la pasarela a la operación “sendWapPush”. Cuando se reciba una petición de descarga se podrá extraer la clave y de esta manera obtener información del destinatario.

Si se desea solicitar el envío de clave es preciso que la URL suministrada en el parámetro “url” termine con la subcadena “k=”. *Altiria* agregará en ese caso a la URL enviada en cada mensaje la clave única con el formato: “xxxxxxxxxxx-zzzzzzzzzz”; ‘xxxxxxxxxxx’ representa el número de teléfono del destinatario en formato internacional y “zzzzzzzzzz” representa un número asociado a cada operación “sendWapPush”, como máximo de diez cifras.

Es necesario resaltar que si se opta por este método la longitud total para los parametros “msg” y “url” pasará de 115 caracteres a 88.

Finalmente para clarificar todos los elementos del servicio de envío de mensajes multimedia a través de la operación “sendWapPush” se esquematizan los procesos involucrados en el envío de un mensaje a dos destinatarios solicitando claves independientes:

1. El cliente accede al servicio web mediante la operación “sendWapPush” de la pasarela de *Altiria* incluyendo los siguientes parámetros:

- destination=346xxxxxxxx.
- destination=346yyyyyyyy.
- url=www.miempresa.com/contenidos/descarga.php?k=

Se observa que se desea enviar el contenido multimedia a dos destinatarios y que además la URL acaba con la subcadena “k=”, solicitando entonces la generación de claves identificadoras.

2. La pasarela de *Altiria* recibe la petición y remite dos mensajes WAP-PUSH individuales a los destinatarios seleccionados. Como URL de descarga, los respectivos teléfonos móviles recibirán (la segunda parte de la clave es simplemente un ejemplo) :

- El móvil “346xxxxxxxx”: www.miempresa.com/contenidos/descarga.php?k=346xxxxxxxx-12365
- El móvil “346yyyyyyyy”: www.miempresa.com/contenidos/descarga.php?k=346yyyyyyyy-12365

Además la pasarela de *Altiria* responde al cliente un status general “000” y los siguientes datos para cada destinatario:

```
status=000; destination=346xxxxxxxx; key=346xxxxxxxx-12365  
status=000; destination=346yyyyyyyy; key=346yyyyyyyy-12365
```

3. Ambos destinatarios reciben en su teléfono la solicitud de aceptación de descarga del contenido multimedia. Suponemos que el destinatario con el número de teléfono “346xxxxxxxx” acepta la descarga.
4. El servidor WEB del cliente, habilitado para ofrecer los contenidos multimedia, recibe una petición de descarga. La petición estará dirigida a la URL

```
http://www.miempresa.com/contenidos/descarga.php?k=346xxxxxxxx-12365
```

por lo que podrá asociarla directamente al destinatario con número de teléfono “346xxxxxxxx”.

5. El servidor WEB entrega el contenido al teléfono del destinatario.
6. El teléfono del destinatario muestra el contenido.

2.6. Confirmación de entrega

El servicio de confirmación de entrega, solicitado a través del parámetro “ack” de las operaciones de envío, permite recibir notificaciones con información sobre el estado de entrega de los mensajes cortos enviados mediante la pasarela.

Para tener acceso a este servicio es preciso que el cliente haya notificado a *Altiria* la dirección de Internet a donde se enviarán las informaciones de confirmación de entrega. En caso contrario, las solicitudes de confirmación de entrega serán ignoradas aunque los mensajes sí serán enviados.

Para que el cliente pueda asociar la información recibida sobre el estado de entrega de un mensaje con el propio mensaje enviado previamente a través de la pasarela, se usará el identificador devuelto en la respuesta a las operaciones de envío en la parte “idAck”.

Este identificador puede albergar dos tipos de valores:

- Si en la propia operación de envío se incluye el parámetro “idAck” (es opcional), contendrá ese valor truncado a veinte caracteres y formado solo por caracteres válidos. Es importante constatar que el identificador devuelto en este caso solo coincidirá con el suministrado en la correspondiente operación de envío si cumple los criterios de composición explicados en las tablas 2.3 y 2.7.
- Si en la propia operación de envío no se incluye el parámetro “idAck”, contendrá un valor numérico de diez dígitos como máximo generado por la pasarela automáticamente.

Las **notificaciones del estado de entrega** serán enviadas mediante la **operación sendNotification** del servicio web SOAP que debe exponer el cliente de acuerdo al **fichero WSDL proporcionado por Altiria** para desarrollar este servicio.

El cliente debe suministrar la URL donde se accederá al servicio web, especificando si se trata de SOAP 1.1 o SOAP 1.2

La codificación de caracteres en la comunicación con el servicio web del cliente será UTF-8.

La operación sobre el servicio web consta de un único parámetro de entrada (tipo Notification).

En el cuadro 2.15 se detallan los parámetros constituyentes del tipo de datos complejo Notification definido en el fichero WSDL del servicio web.

Dato	Valor
destination	Número de teléfono móvil al que se refiere la información de estado de entrega. Si esa información es relativa a un mensaje concatenado, el teléfono figurará cualificado con un sufijo que haga referencia al fragmento particular del que se trate, por ejemplo xxxxxxxxxxx(2) (ver la respuesta al enviar un mensaje concatenado en la sección 2.3.1).
idAck	Identificador que coincidirá con el suministrado por <i>Altiria</i> al cliente en la parte “idAck” de la respuesta a la operación de envío. El contenido de este campo se ha explicado en el inicio de esta sección.
status	Estado relativo a la entrega del mensaje. Podrá tomar los valores: “ENTREGADO”, “NO ENTREGADO”, “ERROR_100”, “ERROR_101”, “ERROR_114” o “ERROR_115”.

Cuadro 2.15: Lista de datos de la confirmación de entrega

El estado “NO ENTREGADO” aparece cuando el mensaje no puede ser entregado al teléfono móvil. Es un estado definitivo, por lo que ese mensaje particular nunca será entregado. Las causas pueden ser múltiples, desde que el teléfono haya estado apagado durante un tiempo superior al periodo de validez, hasta que el número no exista. En general no se puede conocer la causa.

El estado “ERROR_100” indica que el mensaje por el momento no ha podido ser entregado al destinatario debido a algún problema en su teléfono móvil. Las causas más comunes son: mala cobertura, buzón de mensajes cortos lleno o teléfono apagado. El mensaje se intentará enviar varias veces con posterioridad durante un tiempo limitado. Si el problema en el teléfono se subsana a tiempo, el mensaje será finalmente entregado, recibándose la correspondiente confirmación.

El estado “ERROR_101” indica que el mensaje por el momento no ha podido ser entregado al destinatario debido a algún problema en la red de telefonía móvil del operador. Habitualmente, cuando el operador solventa los problemas, el mensaje será entregado, recibándose la correspondiente confirmación.

El estado “ERROR_114” indica que el mensaje ha sido enviado pero no ha podido ser entregado porque el número de teléfono destinatario no existe.

El estado “ERROR_115” indica que el mensaje ha sido enviado pero no ha podido ser entregado porque el destinatario no acepta mensajes.

Un ejemplo de notificación de entrega correspondiente a uno de los fragmentos del envío del SMS concatenado del ejemplo de la operación sendSms (ver sección 2.3.1) sería este:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <NotificationRequest>
      <notificationRequest>
        <destination>346YYYYYYYY(1)</destination>
        <idAck>123456789</idAck>
        <status>ENTREGADO</status>
      </notificationRequest>
    </NotificationRequest>
  </soap:Body>
</soap:Envelope>
```

Opcionalmente se podría configurar el envío de las notificaciones de entrega mediante la llamada a un recurso REST del cliente según la especificación relativa a la confirmación de entrega de la pasarela REST de *Altiria* para el envío de SMS.

Otra alternativa sería configurar el envío de las notificaciones de entrega mediante peticiones HTTP POST como también se detalla en la especificación relativa a la confirmación de entrega de la pasarela HTTP de *Altiria* para el envío de SMS.

Consultar al soporte técnico de *Altiria* (soporte@altiria.com) para conocer más detalles sobre estas posibilidades.

El **servicio web del cliente deberá responder** un único parámetro de tipo “String” con un contenido simple como por ejemplo la cadena “OK”.

Finalmente para clarificar todos los elementos de la funcionalidad de confirmación de entrega, se esquematizan los procesos involucrados en el envío de un mensaje a dos destinatarios:

1. El cliente efectúa un envío de SMS a través de la pasarela de *Altiria* incluyendo entre otros los siguientes parámetros:
 - destination=346xxxxxxx.
 - destination=346yyyyyyy.
 - ack=true.
 - idAck=zzzz.
2. La pasarela de *Altiria* recibe la petición y remite el mensaje a los destinatarios seleccionados. Además la pasarela responde al cliente un status general “000” y los siguientes datos para cada destinatario:

```
status=000; destination=346xxxxxxx; idAck=zzzz  
status=000; destination=346yyyyyyy; idAck=zzzz
```

Si el cliente no hubiera incluido el parámetro “idAck” en la operación “sendSms” (punto 1 del ejemplo), la pasarela de *Altiria* habría autogenerado el identificador para añadirlo a la respuesta.

3. Ambos destinatarios reciben en su teléfono el mensaje corto enviado.
4. El servicio web del cliente, habilitado para recibir las notificaciones de estado de entrega, es accedido desde la pasarela de *Altiria* en sendas peticiones, con el parámetro “notification” conteniendo:

```
destination=346xxxxxxx; idAck=zzzz; status=ENTREGADO  
destination=346yyyyyyy; idAck=zzzz; status=ENTREGADO
```

5. El servicio web del cliente responde la cadena de texto “OK”.
6. Empleando el identificador *zzzz*, el cliente podrá asociar la confirmación de entrega con el mensaje previamente enviado a cada uno de los destinatarios.

2.7. Códigos de estado

El cuadro 2.16 presenta la lista de los posibles códigos de estado que podrán aparecer en la respuesta a cada operación.

CÓDIGO	DETALLE
000	Éxito
001	Error interno. Contactar con el soporte técnico
002	Error de acceso al puerto seguro 443. Contactar con el soporte técnico
010	Error en el formato del número de teléfono
011	Error en el envío de los parámetros de la operación o codificación incorrecta.
013	El mensaje excede la longitud máxima permitida
014	La petición HTTP usa una codificación de caracteres inválida
015	No hay destinatarios válidos para enviar el mensaje
016	Destinatario duplicado
017	Mensaje vacío
018	Se ha excedido el máximo número de destinatarios autorizado
019	Se ha excedido el máximo número de mensajes autorizado
020	Error en la autenticación
022	El remitente seleccionado para el envío no es válido
030	La url y el mensaje superan la longitud máxima permitida
031	La longitud de la url es incorrecta
032	La url contiene caracteres no permitidos
033	El puerto destino del SMS es incorrecto
034	El puerto origen del SMS es incorrecto

Cuadro 2.16: Lista de los códigos de estado

2.8. Ejemplos

Se presentan extractos de programación en varios lenguajes.

El **código fuente** todos los ejemplos se puede **descargar** de esta URL [CODIGOFUENTE]

Altiria no se responsabiliza del funcionamiento de los ejemplos presentados. Se deben considerar como fragmentos de código ilustrativos del acceso a algunas funcionalidades de la pasarela documentada.

Con objeto de **facilitar la lectura algunas líneas del código han sido partidas** con saltos de línea que podrían afectar al correcto funcionamiento del programa en su ejecución.

En los ejemplos se cita la versión de SOAP utilizada (1.1/1.2). Cada versión del protocolo utiliza su propia URL de acceso según se especifica en el apartado 2.1.

2.8.1. Envío de un mensaje en PHP

Ejemplo usando NuSOAP (SOAP 1.1) como cliente de los servicios web (ver [NUSOAP]).

```
<?php
require_once('nusoap-0.9.5/lib/nusoap.php');

//Se fija la URL de los servicios web para SOAP 1.1
$client = new nusoap_client('http://www.altiria.net/api/ws/soap?wsdl', true);
//conexion timeout = 5 segundos
$client->timeout = 5;
//timeout de espera de respuesta = 60 segundos
$client->response_timeout = 60;
//Codificacion UTF-8
$client->soap_defencoding = 'UTF-8';
$client->decode_utf8 = true;

$error = $client->getError();
if($error){
    echo 'Constructor error'. $error;
}

//Se preparan los datos del servicio web
//XX, YY y ZZ se corresponden con los valores de identificacion del usuario en el sistema
$operation = 'sendSms';
$credentials = array('domainId'=>'XX', 'login'=>'YY', 'passwd'=>'ZZ');
//No es posible utilizar el remitente en América pero sí en España y Europa
$message = array('msg'=>'Mensaje de prueba');
//Utilizar esta lista de parámetros solo si se cuenta con un remitente autorizado por Altiria
//$message = array('msg'=>'Mensaje de prueba', 'senderId'=>'remitente');
$destination = array('346xxxxxxxx', '34yyyyyyyy');
$params = array(array('credentials'=>$credentials, 'destination'=>$destination, 'message'=>$message));
//llamada al web service
$result = $client->call($operation, $params);

print_r($result);
echo "<br/>";

if($client->fault){
    echo ('Fault: ');
    print_r($result);
}
```

```

}else{
    $err = $client->getError();
    if($err){
        if (strpos($err,"timed out"))
            //error response timeout
            echo ('ERROR TimeOut');
        else
            //error en la conexion o de connetion timeout
            echo ('Error: '.$err);
    }else{
        //Evaluamos la respuesta de Altiria
        if($result['status']!= '000'){
            echo 'ERROR.Codigo de Altiria: '.$result['status'];
        }else{
            foreach ($result as $clave => $val){
                if($clave=='details') {
                    //Array ([status]=>000 [details]=>Array ([destination]=>34xxxxxxx [status]=>000))
                    if(sizeof($destination)==1) {
                        if ($val['status']=='000')
                            echo $val['destination']." OK <br>";
                        else
                            echo $val['destination']." ERROR. Codigo de error de Altiria: ".$val['status']."<br>";
                    }
                    //Array ( [status] => 000
                    //      [details] => Array (
                    //          [0] => Array ( [destination] => 346xxxxxxx [status] => 000 )
                    //          [1] => Array ( [destination] => 346yyyyyyyy [status] => 000 ) )
                    //      )
                    else {
                        foreach ($val as $clave2 => $val2){
                            if ($val2['status']=='000')
                                echo $val2['destination']." OK <br>";
                            else
                                echo $val2['destination']." ERROR. Codigo de error de Altiria: ".$val2['status']."<br>";
                        }
                    }
                }
            }
        }
    }
}
?>

```

2.8.2. Envío de un mensaje en JAVA

Ejemplo en Java para SOAP 1.2 creando automáticamente las clases del servicio web mediante WsImport (ver [WSIMPORT]), herramienta integrada dentro de la propia JDK de Java.

Un ejemplo de uso sería:

```
wsimport -d src -keep http://www.altiria.net/api/ws/soap12?wsdl
```

Se suministra la URL del fichero WSDL para SOAP 1.2

```
try {
    //Se suministra la URL del fichero WSDL para SOAP 1.2
    URL wsdlURL = new URL("http://www.altiria.net/api/ws/soap12?wsdl");

    net.altiria.api.soap.SmsGatewayService service =
        new net.altiria.api.soap.SmsGatewayService(wsdlURL);
    net.altiria.api.soap.SmsGatewayPort soapApi = service.getSmsGatewayApi();

    BindingProvider bp = (BindingProvider)soapApi;
    //Tiempo maximo de respuesta.
    bp.getRequestContext().put("com.sun.xml.internal.ws.request.timeout", 60000);

    //Se preparan los datos del servicio web
    net.altiria.api.soap.TextMessageRequest textMessageRequest =
        new net.altiria.api.soap.TextMessageRequest();

    //XX, YY y ZZ se corresponden con los valores de identificación del usuario en el sistema
    net.altiria.api.soap.Credentials credentials = new net.altiria.api.soap.Credentials();
    credentials.setDomainId("XX");
    credentials.setLogin("YY");
    credentials.setPasswd("ZZ");

    textMessageRequest.setCredentials(credentials);

    net.altiria.api.soap.TextMessage message = new net.altiria.api.soap.TextMessage();
    message.setMsg("Mensaje de prueba");

    //No es posible utilizar el remitente en América pero sí en España y Europa
    //Descomentar la línea solo si se cuenta con un remitente autorizado por Altiria
    //message.setSenderId("remitente");

    textMessageRequest.setMessage(message);

    textMessageRequest.getDestination().add("346xxxxxxxx");
    textMessageRequest.getDestination().add("346yyyyyyyy");

    net.altiria.api.soap.TextMessageResponse textMessageResponse = null;

    try {
        textMessageResponse = soapApi.sendSms(textMessageRequest);

        String status = textMessageResponse.getStatus();

        if(!status.equals("000"))
            System.out.println("ERROR.Codigo de Altiria: " + status);
        else {
            List<net.altiria.api.soap.TextDestination> destinations =
                textMessageResponse.getDetails();

            for(int i=0;i<destinations.size();i++) {
                System.out.println("sendSms.destination("+i+").status="
                    + destinations.get(i).getStatus());

                System.out.println("sendSms.destination("+i+").msisdn="
```

```
        + destinations.get(i).getDestination());

        if(destinations.get(i).getIdAck()!=null)
            System.out.println("sendSms.destination("+i+").idAck="
                + destinations.get(i).getIdAck());
    }
}
}
catch (WebServiceException e){
    System.out.println("Excepcion:"+e.toString());
    return;
}
}
catch (Exception e) {
    System.out.println("Excepcion:"+e.toString());
}
}
```

2.8.3. Envío de un mensaje en .NET

Todos los ejemplos de .NET se han desarrollado con Visual Studio 2015

Tanto para Visual Basic como para C# hay que generar las clases a partir del WSDL del servicio.

Esto se logra con la herramienta de consola wsdl.exe

```
wsdl.exe /l:CS /nologo /o:C:\stub.cs /namespace:SoapC http://www.altiria.net/api/ws/soap12?wsdl
wsdl.exe /l:VB /nologo /o:C:\stub.vb /namespace:SoapVB http://www.altiria.net/api/ws/soap12?wsdl
```

Se suministra la URL del fichero WSDL para SOAP 1.2.

Otros parámetros de la herramienta wsdl.exe son:

- l: lenguaje (CS, VB)
- o: ruta y nombre del fichero de salida.

En el fichero generado (stub.cs y stub.vb en los ejemplos) se debe definir correctamente la URL de acceso a los servicios web:

```
public SmsGatewayService() {
    this.Url = "http://www.altiria.net/api/ws/soap12";
}

Public Sub New()
    MyBase.New
    Me.Url = "http://www.altiria.net/api/ws/soap12"
End Sub
```

Ejemplo en Visual Basic

```
Module SoapVBAltiria

Sub Main()
```



```

Dim dest(1) As String
Dim sms As SmsGatewayService = New SmsGatewayService()
Dim request As TextMessageRequest = New TextMessageRequest()
Dim cr As Credentials = New Credentials()
Dim tm As TextMessage = New TextMessage()
Dim response As TextMessageResponse = New TextMessageResponse()

' Tiempo maximo de respuesta en milisegundos
sms.Timeout = 60000 'milisec
' XX, YY y ZZ se corresponden con los valores de identificación del usuario en el sistema
cr.domainId = "XX"
cr.login = "YY"
cr.passwd = "ZZ"

tm.encoding = "UTF-8"
tm.msg = "Mensaje de prueba"

' No es posible utilizar el remitente en América pero sí en España y Europa
' Descomentar la línea solo si se cuenta con un remitente autorizado por Altiria
' tm.senderId = "remitente"

dest(0) = "346xxxxxxx"
dest(1) = "346yyyyyyyy"

request.credentials = cr
request.destination = dest
request.message = tm

Try
    response = sms.sendSms(request)
    If (response.status <> "000") Then
        Console.WriteLine("Código de error de Altiria: " + response.status)
    Else
        Console.WriteLine("Código de Altiria: " + response.status)
        For Each rd As TextDestination In response.details
            Console.WriteLine(" " + rd.destination + ": Código de respuesta " + rd.status)
        Next rd

    End If
Catch e As WebException
    If (e.Status = WebExceptionStatus.Timeout) Then
        Console.WriteLine("Error TimeOut")
    Else
        Console.WriteLine("ERROR " + e.Message)
    End If
Catch e As Exception
    Console.WriteLine("ERROR " + e.Message)
End Try

End Sub

End Module

```

Ejemplo en Visual C#

```
string[] dest = new string[2];
SmsGatewayService sms = new SmsGatewayService();
TextMessageRequest request = new TextMessageRequest();
Credentials cr = new Credentials();
TextMessage tm = new TextMessage();
TextMessageResponse response = new TextMessageResponse();

// Tiempo maximo de respuesta en milisegundos
sms.Timeout = 60000; //milisec

// XX, YY y ZZ se corresponden con los valores de identificación del usuario en el sistema
cr.domainId = "XX";
cr.login = "YY";
cr.passwd = "ZZ";

tm.encoding = "UTF-8";
tm.msg = "Mensaje de prueba";

//No es posible utilizar el remitente en América pero sí en España y Europa
//Descomentar la línea solo si se cuenta con un remitente autorizado por Altiria
//tm.senderId = "remitente";

dest[0] = "346xxxxxxxx";
dest[1] = "346yyyyyyyy";

request.credentials = cr;
request.destination = dest;
request.message = tm;

try {
    response = sms.sendSms(request);

    if (response.status != "000")
        Console.WriteLine("Código de error de Altiria: " + response.status);
    else {
        Console.WriteLine("Código de Altiria: " + response.status);
        foreach (TextDestination rd in response.details)
            Console.WriteLine(" " + rd.destination + ": Código de respuesta " + rd.status);
    }
}
catch (WebException e) {
    if (e.Status == WebExceptionStatus.Timeout)
        Console.WriteLine("Error TimeOut");
    else
        Console.WriteLine("ERROR " + e.Message);
}
catch (Exception e) {
    Console.WriteLine("ERROR: " + e.Message);
}
```

2.8.4. Ejemplo en Ruby

Ejemplo en Ruby utilizando la librería Savon (SOAP 1.2) como cliente de los servicios web (ver [SAVON]).

```
# encoding: UTF-8

require 'net/http'
require 'json'
require 'uri'
require 'rubygems'
require 'savon'

def altiriaSms(destinations, message, senderId, debug)
  if debug
    puts "Enter altiriaSms: destinations=#{destinations}, message=#{message}, senderId=#{senderId}"
  end

  begin

    #Se suministra la URL del fichero WSDL para SOAP 1.2
    client = Savon.client(wsdl: 'http://www.altiria.net/api/ws/soap12?wsdl',
      open_timeout: 5, #Tiempo maximo de conexión.
      read_timeout: 60, #Tiempo maximo de respuesta.
      encoding: "UTF-8", #Se fija la codificación de la petición POST
      soap_version: 2) #Se fija la versión del cliente SOAP

    #Se envía la petición
    response = client.call(:send_sms, message: { #Se preparan los datos del servicio web.
      #XX, YY y ZZ se corresponden con los valores de identificación del usuario en el sistema
      credentials: { domainId: 'XX', login: 'YY', passwd: 'ZZ'},
      destination: destinations.split(","),
      message: { msg: message, senderId: senderId }
    })
    if debug
      #Se serializa la respuesta
      filter = response.to_hash
      filter = JSON.parse filter.to_json

      #Error en la respuesta del servidor
      unless response.http.code == 200
        puts("ERROR GENERAL: #{response.http.code}")
        puts("#{response.body}")
      else #Se procesa la respuesta
        puts("Código de estado HTTP: #{response.http.code}")
        puts("Código de estado Altiria: #{filter['text_message_response']['status']}")
        if "#{filter['text_message_response']['status']} == "000"
          puts "text_message_response.details[0].destination ="
            +"#{filter['text_message_response']['details'][0]['destination']}"
          puts "text_message_response.details[0].status ="
            +"#{filter['text_message_response']['details'][0]['status']}"
          puts "text_message_response.details[1].destination ="
            +"#{filter['text_message_response']['details'][1]['destination']}"
          puts "text_message_response.details[1].status ="
            +"#{filter['text_message_response']['details'][1]['status']}"
        end
      end
    end
  end
end
```

```

        else
            puts("Error: #{response.body}")
        end
    end
end

return Nokogiri::XML(response.to_xml)

rescue Net::OpenTimeout
    puts "Tiempo de conexión agotado"
rescue Net::ReadTimeout
    puts "Tiempo de respuesta agotado"
rescue Exception => e
    puts "Error interno: #{e.class}, #{e.message}"
end

end

puts "The function altiriaSms returns:\n "
+ "#{altiriaSms('346xxxxxxxx,346yyyyyyyy','Mensaje de prueba', '', true)}"
#No es posible utilizar el remitente en América pero sí en España y Europa
#Utilizar esta llamada solo si se cuenta con un remitente autorizado por Altiria
#puts "The function altiriaSms returns:\n "
# + "#{altiriaSms('346xxxxxxxx,346yyyyyyyy','Mensaje de prueba', 'remitente', true)}"

```

2.8.5. Ejemplo en Python

Ejemplo en Python utilizando la librería Suds (SOAP 1.2) como cliente de los servicios web (ver [SUDS]).

```

# -*- coding: utf-8 -*-

from suds.client import Client

def altiriaSms(destinations, message, senderId, debug):
    if debug:
        print 'Enter altiriaSms: '+destinations+', message: '+message+', senderId: '+senderId

    try:

        #Se suministra la URL del fichero WSDL para SOAP 1.2
        client = Client('http://www.altiria.net/api/ws/soap12?wsdl')

        client.set_options(timeout=60, #Tiempo maximo de respuesta.
        #Se fija el tipo de contenido de la peticion POST
        headers={'Content-Type': 'application/soap+xml', 'charset': 'UTF-8'},
        faults=False) #Imprescindible para poder recuperar el estado HTTP en la respuesta

        #Se preparan los datos del servicio web
        #XX, YY y ZZ se corresponden con los valores de identificación del usuario en el sistema
        credentialsData = client.factory.create('Credentials')
        credentialsData.domainId = 'XX'
        credentialsData.login = 'YY'

```

```

credentialsData.passwd = 'ZZ'

messageData = client.factory.create('TextMessage')
messageData.msg = message.decode('utf8')
messageData.senderId = senderId

response = client.service.sendSms(credentialsData, destinations.split(','), messageData)

if debug:
    httpStatus = response[0]
    if httpStatus != 200: #Error en la respuesta del servidor
        print 'ERROR GENERAL: '+str(httpStatus)
        print str(response[1])
    else: #Se procesa la respuesta
        print 'Código de estado HTTP: '+str(httpStatus)
        print 'Codigo de estado Altiria: '+response[1].status
        if response[1].status != '000':
            print 'Error: '+str(response[1])
        else:
            print 'Cuerpo de la respuesta:'
            print 'destination[0].status= '+response[1].details[0].status
            print 'destination[0].destination= '+response[1].details[0].destination
            print 'destination[1].status= '+response[1].details[1].status
            print 'destination[1].destination= '+response[1].details[1].destination
    return str(response)

except Exception as ex:
    if str(ex).index('timed out')!=-1:
        print 'Error TimeOut'
    else:
        print 'Error interno: '+str(ex)

print 'The function altiriaSms returns: \n'
    +altiriaSms('346xxxxxxxx,346yyyyyyyy','Mensaje de prueba', '', True)
#No es posible utilizar el remitente en América pero sí en España y Europa
#Utilizar esta llamada solo si se cuenta con un remitente autorizado por Altiria
#print 'The function altiriaSms returns: \n'
#    +altiriaSms('346xxxxxxxx,346yyyyyyyy','Mensaje de prueba', 'remitente', True)

```

2.8.6. Ejemplo en Perl

Ejemplo en Perl utilizando la librería Soap::Lite (SOAP 1.2) como cliente de los servicios web (ver [SOAPLite]).

```

#!/usr/bin/perl

use SOAP::Lite;
use strict;
use warnings;

binmode(STDOUT, ":utf8");

sub altiriaSms{

    #Si debug

```

```

if ($_[3] eq 'true'){
    print 'Enter altiriaSms: destinations='.$_[0].', message='.$_[1].', senderId='.$_[2]."\n";
}

try {
    #Se fija la URL para SOAP 1.2
    my $url = 'http://www.altiria.net/api/ws/soap12';

    #Se crea el objeto de la petición y se establece el tiempo máximo de respuesta
    my $soap = SOAP::Lite->proxy($url, timeout => 60);

    #Se especifica la operación
    $soap->on_action(sub { sprintf '"sendSms"' });

    #Se asocia el namespace del objeto
    $soap->ns("http://www.altiria.net/api/soap","tns");

    #Error en la respuesta del servidor
    $soap->on_fault( sub{ my $soap = shift;
        my $err = $soap->transport->status;
        if(index($err, 'timeout') != -1){
            print "Timeout error \n";
        }else{
            print "Error: ".$err;
        }
    });

    #Se preparan los datos del servicio web.
    #XX, YY y ZZ se corresponden con los valores de identificación del usuario en el sistema
    my @params = ( SOAP::Data->name('tns:credentials' => \SOAP::Data->value(
        SOAP::Data->new(name=>'tns:domainId',value=>'XX')->type('xsd:string'),
        SOAP::Data->new(name=>'tns:login',value=>'YY')->type('xsd:string'),
        SOAP::Data->new(name=>'tns:passwd',value=>'ZZ')->type('xsd:string'))->type('tns:Credentials'));

    #Se añaden los destinatarios al cuerpo de la petición
    my @destinations = split(',', $_[0]);
    foreach my $destination (@destinations){
        push (@params, SOAP::Data->name("tns:destination" => $destination)->type('xsd:string'));
    }

    #Se añade el mensaje al cuerpo de la petición
    push (@params, SOAP::Data->name("tns:message" => \SOAP::Data->value(
        SOAP::Data->new(name => 'tns:msg', value => $_[1])->type('xsd:string'),
        SOAP::Data->new(name=>'tns:senderId',value=>$_[2])->type('xsd:string')
    ))->type('tns:TextMessage'));

    #Se envía la petición
    my $response = $soap->call('TextMessageRequest', @params);

    #Se recupera de la respuesta el nodo del objeto TextMessageResponse
    my $nodes = $response->valueof('//Envelope/Body/TextMessageResponse');

    #Si debug
    if ($_[3] eq 'true'){
        my $altiria_status = $nodes->{status};
    }
}

```

```
print "Código de estado de Altiria: " . $altiria_status. "\n";
if($altiria_status eq '000'){
    print "Cuerpo de la respuesta: \n";
    print "details[0]destination: ".$nodes->{'details'}[0]{'destination'}."\n";
    print "details[0]status: ".$nodes->{'details'}[0]{'status'}."\n";
    print "details[1]destination: ".$nodes->{'details'}[1]{'destination'}."\n";
    print "details[1]status: ".$nodes->{'details'}[1]{'status'}."\n";
}else{
    print "Error de Altiria: ".$response."\n";
}
}

#Se serializa la respuesta
my $serializer = SOAP::Serializer->new;
my $xml = $serializer->serialize($response->dataof('//'));
return $xml;
};
catch { #Se capturan posibles errores en el lado cliente
    print "Error: ".$@->what."\n";
};
}

print "The function altiriaSms returns: "
    .altiriaSms('346xxxxxxxx,346yyyyyyyy','Mensaje de prueba', '', 'true')."\n";
#No es posible utilizar el remitente en América pero sí en España y Europa
#Utilizar esta llamada solo si se cuenta con un remitente autorizado por Altiria
#print "The function altiriaSms returns: "
#    .altiriaSms('346xxxxxxxx,346yyyyyyyy','Mensaje de prueba', 'remitente', 'true')."\n";
```

Referencias

- [FAQ] *Preguntas frecuentes de la pasarela de envío de SMS de Altiria:*
<http://www.altiria.com/preguntas-frecuentes-faq-pasarela-sms-api>
- [HTTPCLIENT] *El proyecto HTTPCLIENT de Apache:*
<http://hc.apache.org/httpcomponents-client-ga/>
- [NUSOAP] *Librería NuSOAP en Sourceforge:*
<http://sourceforge.net/projects/nussoap/>
- [WSIMPORT] *Herramienta para generación de WS en Java:*
<http://docs.oracle.com/javase/8/docs/technotes/tools/unix/wsimport.html>
- [PERL] *Página oficial de descargas de perl:*
<https://www.perl.org/get.html>
- [CODIGOFUENTE] *Enlace a la descarga del código fuente de los ejemplos:*
<https://static.altiria.com/especificaciones/altiria-push-ejemplos-codigo-fuente.zip>
- [REQUESTS] *Página oficial de Requests:*
<http://docs.python-requests.org>
- [SAVON] *Página oficial de Savon:*
<http://savourb.com>
- [SUDS] *API de Suds:*
<https://pypi.python.org/pypi/suds>
- [SOAPLite] *API de SOAP::Lite:*
<http://search.cpan.org/~phred/SOAP-Lite-1.20/lib/SOAP/Lite.pm>