

# Especificaciones de la Pasarela SMS PREMIUM

Altiria TIC, S.L.L.

Versión: 3.0

Copyright © Altiria TIC 2009

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
3.0	Reescritura del documento para añadir la opción de modo asíncrono. Cambios en los parámetros del POST inicial de <i>Altiria</i> y en la respuesta del cliente.

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Modo síncrono o de respuesta inmediata . . . . .	3
1.2. Modo asíncrono o de respuesta dirigida . . . . .	3
<b>2. Petición inicial de <i>Altiria</i></b>	<b>4</b>
2.1. Política de reintentos . . . . .	5
<b>3. Respuesta del cliente a la petición inicial</b>	<b>6</b>
3.1. Modo síncrono . . . . .	6
3.1.1. Mensaje de texto . . . . .	6
3.1.2. Mensaje WAP-PUSH . . . . .	6
3.2. Modo asíncrono . . . . .	7
<b>4. Envío de la respuesta al usuario en modo asíncrono</b>	<b>7</b>
4.1. Parámetros de la petición . . . . .	8
4.2. Tipos de respuesta . . . . .	8
4.2.1. Mensaje de texto . . . . .	8
4.2.2. Mensaje WAP-PUSH . . . . .	8
4.3. Respuesta de <i>Altiria</i> a la petición del cliente . . . . .	9
<b>5. Caracteres de un mensaje de texto</b>	<b>10</b>
<b>6. Caracteres de un mensaje WAP-PUSH</b>	<b>10</b>
<b>7. Caracteres de una URL de un mensaje WAP-PUSH</b>	<b>10</b>
<b>8. Seguridad</b>	<b>11</b>
<b>9. Mensajes multimedia WAP-PUSH</b>	<b>12</b>
9.1. Tipos de contenido . . . . .	12
9.2. Formato del contenido . . . . .	12
9.3. Dirección del contenido . . . . .	12
9.4. Envío del contenido . . . . .	13
<b>10. Ejemplos de uso en modo síncrono</b>	<b>13</b>
10.1. Ejemplo en PHP . . . . .	13
10.2. Ejemplo en JAVA . . . . .	13
<b>11. Ejemplos de uso en modo asíncrono</b>	<b>14</b>
11.1. Ejemplo en PHP . . . . .	14
11.1.1. Recepción de la petición inicial de <i>Altiria</i> . . . . .	14
11.1.2. Envío de la respuesta usuario . . . . .	14
11.2. Ejemplo en JAVA . . . . .	15

11.2.1. Recepción de la petición inicial de <i>Altiria</i> . . . . .	15
11.2.2. Envío de la respuesta al usuario . . . . .	15

## 1. Introducción

En este documento se presentan las especificaciones técnicas para el uso de la pasarela HTTP de *Altiria* para recepción y envío de información a través de mensajes cortos SMS PREMIUM

*Altiria* cuenta con una pasarela que permite la interconexión entre las operadoras de telefonía móvil, que reciben los mensajes cortos enviados por los usuarios, y nuestro cliente.

Son posibles dos modos de funcionamiento:

### 1.1. Modo síncrono o de respuesta inmediata

El modelo de funcionamiento es muy simple y sigue el siguiente esquema:

1. El usuario interesado en recibir cierta información envía un mensaje corto a un número prefijado, incluyendo habitualmente como parte del mensaje una palabra clave asociada al servicio de información.
2. *Altiria* recibe el mensaje enviado por el usuario desde la operadora correspondiente.
3. *Altiria* formatea la información recibida y la envía al cliente a través de una petición HTTP POST, enviando en cada parámetro de la petición una variable de interés relacionada con el mensaje (ver sección 2).
4. El cliente recibe los parámetros de la petición. De acuerdo a su propia lógica, procesa la información recibida y genera un mensaje de respuesta para el usuario.
5. El cliente envía esa contestación **de forma inmediata** en respuesta a la petición HTTP POST inicial de *Altiria* (ver sección 3).
6. *Altiria* recibe esa información y se la remite a la operadora para que la envíe en un SMS al teléfono móvil del usuario.
7. El usuario recibe en su móvil la información solicitada.

Como se observa el proceso es muy sencillo, tan solo requiere por parte del cliente la puesta en marcha de un pequeño servidor HTTP en una URL determinada, que reconozca la petición enviada por *Altiria*, la procese y genere la respuesta adecuada para el usuario.

### 1.2. Modo asíncrono o de respuesta diferida

El modelo de funcionamiento es algo más complicado de acuerdo al siguiente esquema:

1. El usuario interesado en recibir cierta información envía un mensaje corto a un número prefijado, incluyendo como parte del mensaje una palabra clave asociada al servicio de información.
2. *Altiria* recibe el mensaje enviado por el usuario desde la operadora correspondiente.
3. *Altiria* formatea la información recibida y la envía al cliente a través de una petición HTTP POST, enviando en cada parámetro de la petición una variable de interés relacionada con el mensaje (ver sección 2).
4. El cliente recibe los parámetros de la petición e inmediatamente contesta un OK al servidor de *Altiria* (ver sección 3).
5. **En diferido** (hasta varias horas), el cliente procesa la información de acuerdo a su propia lógica, genera el mensaje de respuesta para el usuario y lo envía en una nueva petición HTTP POST sobre el servidor de *Altiria* (ver sección 4).

6. *Altiria* recibe esa información y se la remite a la operadora para que la envíe en un SMS al teléfono móvil del usuario.
7. El usuario recibe en su móvil la información solicitada.

Como se observa el proceso es un poco más complejo que en el caso de respuesta síncrona porque además de la puesta en marcha de un pequeño servidor HTTP en una URL determinada, que reconozca la petición enviada por *Altiria*, se precisa un cliente HTTP que envíe posteriormente la respuesta generada para el usuario de nuevo hacia el servidor de *Altiria*.

## 2. Petición inicial de *Altiria*

La comunicación entre *Altiria* y el cliente comienza con la petición HTTP POST inicial de *Altiria* sobre el servidor del cliente en una URL prefijada.

El cuerpo de esta petición HTTP POST será del tipo:

*“Content-type: application/x-www-form-urlencoded”*

Estará compuesto por una lista de pares [nombre,valor], codificados con el juego de caracteres (charset) *“UTF-8”*.

La lista de parámetros enviados se detalla en el cuadro 1:

Nombre	Valor	Obligatorio
alias	Identificador único del mensaje enviado. Necesario para enviar la posterior respuesta al usuario en caso de operar en modo asíncrono (ver sección 4). Útil también para distinguir peticiones correspondientes a reintentos pues será similar en cada reenvío (ver sección 2.1). Formato numérico	sí
telnum	Número de teléfono del usuario que mandó el mensaje corto solicitando una información. Se enviará en formato de numeración internacional, con el prefijo de país (34 para España). Ej: 34610123456	sí
keyword	Palabra clave asociada al servicio de información. Será la primera palabra que contendrá el mensaje enviado por el usuario.	sí
text	Texto correspondiente al mensaje enviado por el usuario. Por defecto no incluye la palabra clave aunque se puede añadir si el cliente así lo precisa.	sí
provider	Código numérico de la operadora de la que es abonado el usuario que mandó el mensaje. La relación completa de códigos se detalla en el cuadro 2. Consultar las operadoras disponibles en cada servicio.	sí
date	Fecha en la que se recibió el mensaje corto del usuario. Sigue el formato aaaammddhhmmss (AñoMesDíaHoraMinutoSegundo)	sí
shortnum	Número corto asociado al servicio. Será el número al que envíe el mensaje corto el usuario	si

Cuadro 1: Lista de parámetros de la petición inicial de *Altiria*

Operadora	Código
Movistar	1
Vodafone	2
Orange	3
Euskaltel	4
Yoigo	5
R	6
Telecable	11
ONO	12
Best	13
BT	14
Barablu	15
Jazztel	16
Simyo	17
Vizzavi	18
FonYou	19
Lleida Networks	20
Lyca Mobile	21
Test	99

Cuadro 2: Códigos asociados a las operadoras

## 2.1. Política de reintentos

Existen varias ocasiones en las que *Altiria* reintentará enviar la petición HTTP POST inicial, todas ellas dependientes del resultado de la respuesta del cliente:

- Si el código de status de la respuesta no es 200, es decir se ha producido algún error en el servidor del cliente.
- Si la respuesta a la petición no comienza con el texto adecuado en cada caso (ver sección 3 para conocer los detalles).
- Si no es posible conectarse con el servidor del cliente para enviar la petición HTTP POST inicial.
- Si no se obtiene respuesta del cliente en un tiempo máximo de 5 segundos después del envío de la petición HTTP POST inicial.

En estos casos, *Altiria* considerará que ha ocurrido algún error y volverá a enviar la petición inicial con los mismos datos.

El cliente podrá identificar que una petición es un reintento mediante el parámetro "alias" pues será similar en cada reenvío tal y como se detalla en el cuadro 1. El cliente debería procesar la petición y responder al servidor de *Altiria* (ver sección 3) a fin de evitar más reintentos.

Cada petición se reintentará un número determinado de veces antes de considerar que el cliente no la puede procesar. El valor y espacio de tiempo entre reintentos no están determinados a priori y *Altiria* se reserva el derecho de modificar esta política sin previo aviso para ofrecer un mejor servicio a sus clientes. Cuando se efectúe el último intento sin conseguir comunicar el mensaje al cliente, *Altiria* seguirá la política establecida de acuerdo a la legislación vigente en materia de servicios SMS PREMIUM.

### 3. Respuesta del cliente a la petición inicial

La respuesta será del tipo “*Content-type: text/plain*” y contendrá una cadena de texto con caracteres codificados en el juego (charset) “*UTF-8*”. Es preciso indicar el juego de caracteres usado en la cabecera de la respuesta HTTP para que la pasarela reciba la cadena de texto correctamente.

El cliente responderá en un plazo máximo de 5 segundos a la petición HTTP POST inicial de *Altiria*. De lo contrario se considerará que ha sucedido un error y se actuará de acuerdo a la política de reintentos (ver sección 2.1).

El contenido de la respuesta dependerá del modo de operación escogido:

#### 3.1. Modo síncrono

La respuesta del cliente será enviada por *Altiria* al usuario, quedando completado el servicio.

Puede ser de dos tipos:

##### 3.1.1. Mensaje de texto

La respuesta será recibida por el usuario como un mensaje corto de texto en su teléfono móvil.

El texto de respuesta comenzará por “**TEXT**” si el cliente ha recibido correctamente los parámetros de la petición y ha sido capaz de generar la respuesta para el usuario. Si ha sucedido algún problema podrá responder cualquier otra cosa o generar un código de status distinto de 200 para que actúe la política de reintentos (ver sección 2.1).

La respuesta completa tendrá este formato:

TEXT TEXTO\_MENSAJE

Como separador entre las dos partes se usará un espacio.

**TEXTO\_MENSAJE** será una cadena de texto libre de hasta 160 caracteres. Si se excede la longitud máxima permitida será truncada antes de remitirla al usuario. No puede estar vacía; en ese caso *Altiria* seguirá la política establecida de acuerdo a la legislación vigente en materia de servicios SMS PREMIUM. Deberá contener preferentemente los caracteres incluidos en el apartado 5.

##### 3.1.2. Mensaje WAP-PUSH

La respuesta será recibida por el usuario como un mensaje WAP-PUSH en su teléfono móvil (ver sección 9).

El texto de respuesta comenzará por “**WAP**” si el cliente ha recibido correctamente los parámetros de la petición y ha sido capaz de generar la respuesta para el usuario. Si ha sucedido algún problema podrá responder cualquier otra cosa o generar un código de status distinto de 200 para que actúe la política de reintentos (ver sección 2.1).

La respuesta completa tendrá este formato:

WAP URL TEXTO\_MENSAJE

Como separador entre cada una de las partes se usará un espacio.

**URL** será una cadena de texto correspondiente a la dirección de Internet desde donde el teléfono móvil se descargará el contenido. No se debe especificar el puerto de conexión (se usará el puerto por defecto, el 80) y debe comenzar por “http://” (en caso contrario será añadido automáticamente). Debe contener caracteres validos en una URL (ver sección 7). No puede estar vacía; en ese caso *Altiria* seguirá la política establecida de acuerdo a la legislación vigente en materia de servicios SMS PREMIUM.

**TEXTO\_MENSAJE** será una cadena de texto libre. Representa una breve descripción del contenido multimedia. Debería contener exclusivamente caracteres incluidos en el apartado 6. Si no se incluye se enviará el número corto en que se opera el servicio.

TEXTO\_MENSAJE y URL (incluyendo el prefijo "http://") no pueden rebasar los 115 caracteres. Si sucede así, TEXTO\_MENSAJE será reemplazado por el número corto en que se opera el servicio. Si aún se excediese de los 115 caracteres *Altiria* seguirá la política establecida de acuerdo a la legislación vigente en materia de servicios SMS PREMIUM.

### 3.2. Modo asíncrono

El texto de respuesta consistirá únicamente en la cadena "OK". Si ha sucedido algún problema se podrá responder cualquier otra cosa o generar un código de status distinto de 200 para que actúe la política de reintentos (ver sección 2.1).

A partir de este momento, de manera diferida (hasta varias horas), el cliente podrá procesar la petición para generar la respuesta y emitirla al usuario para poder dar por completado el servicio (ver sección 4 a continuación).

## 4. Envío de la respuesta al usuario en modo asíncrono

El cliente procesará la información de la petición inicial de *Altiria* de acuerdo a su propia lógica. Una vez que tenga la respuesta preparada para entregar al usuario, debe enviarla en una nueva petición HTTP POST.

El cuerpo de esta nueva petición HTTP POST será del tipo:

*"Content-type: application/x-www-form-urlencoded"*

Estará compuesto por una lista de pares [nombre,valor] (ver sección 4.1), codificados con el juego de caracteres (charset) "UTF-8".

La **URL** sobre la que enviar las peticiones HTTP debe ser suministrada por *Altiria*.

También es posible enviar la lista de pares [nombre,valor] como parte de la cadena de caracteres que conforma la URL de la petición POST, siguiendo el siguiente esquema:

http://url\_del\_servidor?nombre1=valor1&nombre2=valor2&nombre3=valor3

Debido a las limitaciones en el número máximo de caracteres de la URL, dependientes de numerosos factores, se desaconseja este método de envío de los parámetros.

En ningún caso se permitirán peticiones HTTP GET.

En caso de enviar los parámetros como parte de la URL, será preciso codificar su valor previamente a su inclusión en la misma. Esto consiste en obtener la representación en hexadecimal en UTF-8 de cada carácter y añadirlo a la URL con un "%" para cada par de dígitos hexadecimales.

Como se ha visto la URL se compondrá a partir de varios parámetros del estilo

nombre1=valor1&nombre2=valor2&nombre3=valor3

De acuerdo a lo indicado cada uno de los valores de los parámetros deberá codificarse en UTF-8 antes de incluirlo en la URL. Por ejemplo una "ñ" se codificaría como "%C3%B1".

Esto es especialmente necesario en los siguientes casos (se indica el carácter y su codificación en la URL en UTF-8):

+ => %2B  
% => %25  
& => %26

De cualquier modo es fundamental codificar los datos de la petición según se ha detallado, de lo contrario la interfaz HTTP podría recibir caracteres incorrectos, siendo esto especialmente grave pues serán enviados en el mensaje de respuesta al usuario.

#### 4.1. Parámetros de la petición

La lista de parámetros enviados en la petición HTTP POST del cliente se detalla en el cuadro 3. Se debe prestar atención a los parámetros obligatorios:

Nombre	Valor	Obligatorio
alias	Identificador único del mensaje enviado por <i>Altiria</i> en la petición inicial. Formato numérico	sí
telnum	Número de teléfono del usuario que mandó el mensaje corto solicitando una información. Se enviará en formato de numeración internacional, con el prefijo de país (34 para España). Ej: 34610123456	sí
keyword	Palabra clave asociada al servicio de información.	sí
shortnum	Número corto asociado al servicio.	sí
type	Tipo de mensaje de respuesta: 0 para texto (ver sección 4.2.1), 1 para WAP-PUSH (ver sección 4.2.2).	sí
text	Texto con la respuesta del cliente, según sea un mensaje de texto (ver sección 4.2.1) o un WAP-PUSH (ver sección 4.2.2).	sí para mensajes de texto
url	Url en caso de respuesta WAP-PUSH (ver sección 4.2.2).	sí para mensajes WAP-PUSH

Cuadro 3: Lista de parámetros de la petición del cliente con la respuesta

#### 4.2. Tipos de respuesta

La respuesta del cliente será enviada por *Altiria* al usuario siempre que no exista ningún error (ver sección 4.3), quedando completado el servicio.

Puede ser de dos tipos:

##### 4.2.1. Mensaje de texto

La respuesta será recibida por el usuario como un mensaje corto de texto en su teléfono móvil.

El parámetro **”text”** (obligatorio) de la petición POST del cliente (ver cuadro 3) será una cadena de texto libre de hasta 160 caracteres, conteniendo preferentemente los incluidos en el apartado 5. Si se excede la longitud máxima permitida se generará un código de error específico (ver sección 4.3).

##### 4.2.2. Mensaje WAP-PUSH

La respuesta será recibida por el usuario como un mensaje WAP-PUSH en su teléfono móvil (ver sección 9).

El parámetro **”url”** (obligatorio) de la petición POST del cliente (ver cuadro 3) será una cadena de texto correspondiente a la dirección de Internet desde donde el teléfono móvil se descargará el contenido. No se debe especificar el puerto de conexión (se usará el puerto por defecto, el 80) y debe comenzar por **”http://”** (en caso contrario será añadido automáticamente). Debe contener caracteres validos en una URL (ver sección 7).

El parámetro "text" de la petición POST del cliente (ver cuadro 3) será una cadena de texto libre. Representa una breve descripción del contenido multimedia. Debería contener exclusivamente caracteres incluidos en el apartado 6. Si no se incluye o está vacío se enviará el número corto en que se opera el servicio.

Los parámetros "text" y "url" (incluyendo el prefijo "http://") no pueden rebasar los 115 caracteres. Si se excede la longitud máxima permitida se generará un código de error específico (ver sección 4.3).

### 4.3. Respuesta de *Altiria* a la petición del cliente

La respuesta de *Altiria* a la petición del cliente pueden ser:

- En caso de éxito: **OK**
- En caso de error: **ERROR xxx**

En caso de éxito el usuario recibirá en su teléfono móvil la respuesta del cliente.

En caso de error, el valor de "xxx" se corresponde con cada uno de los posibles códigos de error, recogidos en el cuadro 4. En esta situación si el error es subsanable (prestar atención a la información asociada a cada error) el cliente debería corregirlo para remitir de nuevo la respuesta al usuario siguiendo lo detallado en la sección 4.

No hay un límite en el número máximo de intentos que puede efectuar el cliente para remitir una respuesta correcta al usuario. Sin embargo sí que se establece un tiempo límite (de varias horas) para responder correctamente desde que el usuario envía el mensaje que desencadena todo el proceso. *Altiria* se reserva el derecho de modificar sin previo aviso esta política para adecuarse a los requerimientos de las operadoras de telefonía.

No está permitido intentar responder más de una vez el mismo mensaje de usuario, identificado unívocamente por el parámetro "alias" (ver cuadro 3), después de haberlo contestado con éxito una primera vez.

CÓDIGO	DETALLE
001	Error interno. Contactar con el soporte técnico antes de reintentar el envío.
002	El parámetro "alias" no es válido o no se corresponde con los parámetros "telnum", "keyword" y "shortnum" de la petición inicial de <i>Altiria</i> .
003	Error en el envío de los parámetros. No se ha encontrado algún parámetro obligatorio.
004	Parámetro "text" vacío en mensaje de texto o parámetro "url" vacío en mensaje WAP-PUSH.
005	El parámetro "text" excede la longitud máxima en un mensaje de texto.
006	Los parámetros "text" y "url" exceden la longitud máxima en un mensaje WAP-PUSH.
007	Ya se ha enviado una respuesta al usuario para el parámetro "alias" suministrado. No se debe reintentar el envío.
008	Ha vencido el tiempo límite para responder a este mensaje. No se debe reintentar el envío.

Cuadro 4: Lista de los códigos de error

## 5. Caracteres de un mensaje de texto

Los caracteres válidos como respuesta de un mensaje de texto al usuario deberían limitarse a los contenidos en la norma ETSI GSM 03.38. Se trata de un estándar público, accesible a través del buscador de la ETSI (ver [GSM.03.38]).

Si se envían caracteres ajenos a la norma, es posible que sean mostrados incorrectamente o con alguna transformación en el teléfono móvil receptor (por ejemplo las vocales acentuadas en general se muestran sin acentuar).

Por otra parte, en función de la operadora, también es posible que determinados caracteres incluidos en la norma sean recibidos por el teléfono móvil con alguna transformación.

El cuadro 5 refleja los caracteres recomendados de respuesta al usuario. Estos en principio serán recibidos en el móvil sin ninguna transformación.

cr <sup>1</sup>	lf <sup>2</sup>	sp <sup>3</sup>	!	”	#	&
,	(	)	*	+	,	-
.	/	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
@	\$	Ç	%	ñ	Ñ	ü
Ü	¿	¡	-			

Cuadro 5: Lista de caracteres recomendados

## 6. Caracteres de un mensaje WAP-PUSH

Los caracteres válidos y recomendados para el texto de un mensaje WAP-PUSH son los mismos que para un mensaje de texto con las siguientes salvedades:

- Si se emplean vocales con tilde o acento agudo (á) se enviarán al teléfono móvil sin acentuar.
- Los caracteres Ññ se sustituyen por los caracteres Nn.

Estos reemplazos son necesarios ya que los caracteres sustituidos no se permiten en las alertas WAP-PUSH.

## 7. Caracteres de una URL de un mensaje WAP-PUSH

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

- Los caracteres del cuadro 6 son seguros y se pueden incluir sin codificar.

<sup>1</sup>Retorno de carro

<sup>2</sup>Nueva línea

<sup>3</sup>Espacio blanco

- Los caracteres del cuadro 7 son reservados y se pueden incluir sin codificar si se emplean dentro de la URL de acuerdo a su uso reservado. Por ejemplo el caracter “&” 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.

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 como juego de caracteres de la codificación):

`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.

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 6: Lista de caracteres seguros

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

Cuadro 7: Lista de caracteres reservados

## 8. Seguridad

Para evitar que el cliente pueda recibir peticiones HTTP de una fuente fraudulenta, ajena a *Altiria*, se puede restringir en el servidor del cliente la dirección IP origen de la petición, permitiendo solo conexiones de los servidores de *Altiria*.

Si el cliente está interesado en esta opción debe solicitar al soporte técnico de *Altiria* las IPs válidas de sus servidores.

## 9. Mensajes multimedia WAP-PUSH

La pasarela SMS Premium 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 descripción del contenido que se ofrece (el texto del mensaje WAP-PUSH) y la dirección de Internet donde se ubica dicho contenido (la url del mensaje WAP-PUSH).

Habitualmente cuando un teléfono móvil recibe un mensaje WAP-PUSH, le presenta al usuario la breve descripción mencionada junto con la posibilidad de acceder al contenido referenciado. Si el usuario acepta, el teléfono de manera automática se conecta a través de HTTP y en función del contenido podrá navegar o solicitar una descarga.

### 9.1. Tipos de contenido

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

De este modo se podrá acceder a 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 acceder directamente a un archivo de un tipo en concreto, como una imagen o un sonido.

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 “xhtml”, un recurso manejado por la gran mayoría de los terminales, 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.

### 9.2. 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 “xhtml”.

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 120 x 120 “pixels”.

### 9.3. Dirección del contenido

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

Es obvio que para que el teléfono pueda descargarse la información la dirección debe representar 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 puede suponer que si se recibe un

mensaje WAP-PUSH con la misma dirección del contenido asociado que un mensaje ya recibido y almacenado, el teléfono lo desheche sin avisar al usuario.

## 9.4. 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.

## 10. Ejemplos de uso en modo síncrono

Se presentan extractos de programación en varios lenguajes de la recepción por parte del cliente de la petición inicial de *Altiria* y de la respuesta síncrona del mensaje para el usuario:

### 10.1. Ejemplo en PHP

Ejemplo en PHP recuperando los parámetros del POST a través de la variable global `_POST` (ver [PHP\_POST]):

```
<?
// Se extraen los parámetros enviados por Altiria en la petición HTTP POST.
// Se recuperan de la variable global _POST y se guardan en variables
$alias .= $_POST['alias'];
$mensaje .= $_POST['text'];
$telefono .= $_POST['telnum'];
$keyword .= $_POST['keyword'];
$shortnum .= $_POST['shortnum'];

// Se procesan los datos y se compone la respuesta de texto para el usuario
$respuesta="TEXT Su petición ha sido procesada con éxito";

// Se compone la cabecera de la respuesta HTTP
header("Content-Type: text/plain; charset=UTF-8");
echo $respuesta;
?>
```

### 10.2. Ejemplo en JAVA

```
public void replyRequest(HttpServletResponse response) throws java.io.IOException {

    response.setContentType("text/plain; charset=UTF-8");
    java.io.PrintWriter out = response.getWriter();

    //Se procesan los datos y se compone la respuesta de texto para el usuario
    String responseStr = "TEXT Su petición ha sido procesada con éxito";

    out.print(responseStr);
    response.flushBuffer();
}
```

```

        out.close();
    }

```

## 11. Ejemplos de uso en modo asíncrono

Se presentan extractos de programación en varios lenguajes de la recepción por parte del cliente de la petición inicial de *Altiria* y de la respuesta asíncrona del mensaje para el usuario:

### 11.1. Ejemplo en PHP

#### 11.1.1. Recepción de la petición inicial de *Altiria*

Ejemplo en PHP recuperando los parámetros del POST a través de la variable global `_POST` (ver [PHP\_POST]):

```

<?
// Se extraen los parámetros enviados por Altiria en la petición HTTP POST.
// Se recuperan de la variable global _POST y se guardan en variables
$alias .= $_POST['alias'];
$mensaje .= $_POST['text'];
$telefono .= $_POST['telnum'];
$keyword .= $_POST['keyword'];
$shortnum .= $_POST['shortnum'];

// Se responde al servidor de Altiria
$respuesta="OK";

// Se compone la cabecera de la respuesta HTTP
header("Content-Type: text/plain; charset=UTF-8");
echo $respuesta;
?>

```

#### 11.1.2. Envío de la respuesta usuario

```

<?
// Como ejemplo la petición se envía a www.altiria.net/sustituirPOSTmt
// Se debe reemplazar la cadena '/sustituirPOSTmt' por la parte correspondiente
// de la URL suministrada por Altiria al dar de alta el servicio

function AltiriaSMS($alias,$telnum,$keyword,$shortnum,$type,$text) {
    $sData = "alias=".$alias."&telnum=".$telnum."&keyword=".$keyword."&shortnum="
        ."$shortnum."&type=".$type."&text=".urlencode(utf8_encode(substr($text,0,160)));
    $fp = fsockopen("www.altiria.net", 80);

    // Reemplazar la cadena '/sustituirPOSTmt' por la parte correspondiente
    // de la URL suministrada por Altiria al dar de alta el servicio
    $buf = "POST /sustituirPOSTmt HTTP/1.0\r\n";
    $buf .= "Host: www.altiria.net\r\n";
    $buf .= "Content-type: application/x-www-form-urlencoded; charset=UTF-8\r\n";
    $buf .= "Content-length: ".strlen($sData)."\r\n";
    $buf .= "\r\n";

```

```
$buf .= $sData;
fputs($fp, $buf);
$buf = "";
while (!feof($fp))
    $buf .= fgets($fp,128);
fclose($fp);
if (strstr($buf,"ERROR"))
    print "Error al enviar mensaje de respuesta\n";
else
    print "Mensaje de respuesta enviado correctamente\n";
}
```

## 11.2. Ejemplo en JAVA

### 11.2.1. Recepción de la petición inicial de *Altiria*

```
public void replyRequest(HttpServletResponse response) throws java.io.IOException {

    response.setContentType("text/plain; charset=UTF-8");
    java.io.PrintWriter out = response.getWriter();

    //Se responde al servidor de Altiria
    String responseStr = "OK";

    out.print(responseStr);
    response.flushBuffer();
    out.close();
}
```

### 11.2.2. Envío de la respuesta al usuario

Ejemplo en Java utilizando HttpClient 3 como cliente HTTP (ver [HTTPCLIENT]):

```
//Se inicia el objeto HTTP
HttpClient client = new HttpClient();
client.setStrictMode(true);

//Se fija el tiempo máximo de espera de la respuesta del servidor
client.setTimeout(60000);

//Se fija el tiempo máximo de espera para conectar con el servidor
client.setConnectionTimeout(5000);

PostMethod post = null;

//Se fija la URL sobre la que enviar la petición POST
//Como ejemplo la petición se envía a www.altiria.net/sustituirPOSTmt
//Se debe reemplazar la cadena '/sustituirPOSTmt' por la parte correspondiente
//de la URL suministrada por Altiria al dar de alta el servicio
post = new PostMethod("http://www.altiria.net/sustituirPOSTmt");

//Se fija la codificación de caracteres en la cabecera de la petición
post.setRequestHeader("Content-type",
```

```
"application/x-www-form-urlencoded; charset=UTF-8");

//Se crea la lista de parámetros a enviar en la petición POST
NameValuePair[] parametersList = new NameValuePair[6];

//XX, YY y ZZ se corresponden con los valores de identificación del
//usuario en el sistema.
parametersList[0] = new NameValuePair("alias", alias);
parametersList[1] = new NameValuePair("telnum", telnum);
parametersList[2] = new NameValuePair("keyword", keyword);
parametersList[3] = new NameValuePair("shortnum", shortnum);
parametersList[4] = new NameValuePair("type", "0");
parametersList[5] = new NameValuePair("text", "Su petición ha sido procesada con éxito");

//Se rellena el cuerpo de la petición POST con los parámetros
post.setRequestBody(parametersList);

int httpstatus = 0;
String response = null;

try {
    //Se envía la petición
    httpstatus = client.executeMethod(post);

    //Se consigue la respuesta
    response = post.getResponseBodyAsString();
}
catch (Exception e) {
    //Habrà que prever la captura de excepciones
    return;
}
finally {
    //En cualquier caso se cierra la conexión
    post.releaseConnection();
}
//Habrà que prever posibles errores en la respuesta del servidor
if (httpStatus!=200){
    return;
}
else {
    //Se procesa la respuesta capturada en la cadena ‘response’
}
}
```

## Referencias

[PHP\_POST] *PHP predefined variables*, <http://www.php.net/manual/en/reserved.variables.php>.

[GSM\_03.38] *GSM 03.38 Digital cellular telecommunications system (Phase 2+). Alphabets and language-specific information, version 5.3.0*, <http://pda.etsi.org/pda/queryform.asp>.

[HTTPCLIENT] *El proyecto HTTPCLIENT de Jakarta*: <http://hc.apache.org/httpclient-3.x/>