



GOBIERNO DE CHILE
SERVICIO NACIONAL DE ADUANAS

Subdirección de Informática

**Manual de referencia de sistema para la
invocación de Web Services con Aduanas
(SMS v3.1.12)**

VERSION 3.1.12

MAYO 2008

Introducción

El sistema SMS (Sistema de Mensajería por SOAP) fue diseñado con el fin de servir como herramienta de apoyo a los programadores para desarrollar sistemas similares, que se encarguen de gestionar mensajería vía SOAP con los WebServices de Aduana.

Para facilitar la comprensión del sistema SMS, es necesario contar con una documentación detallada que explique el funcionamiento del sistema en sus aspectos más importantes, centrándose principalmente en la invocación de los WebServices. Este es el objetivo que pretende cubrir el presente documento.

Plataforma Software

Los ejemplos tratados en este documento se basan en:

- API SOAP V 2.0 de Apache
- Lenguaje de programación Java (desde Runtime 1.5)

Es importante destacar que nuestro ejemplo esta basado en Java, pero lo anterior *no limita la comprensión y portabilidad* a otros lenguajes, ya que SOAP precisamente fue diseñado para lograr la invocación de Servicios que pueden estar desarrollados en leguajes distintos. Por lo anterior, el demo desarrollado (SMS) es fácilmente homologable a cualquier lenguaje que utilice SOAP para invocar servicios remotos.

Por último, es importante destacar que al final de este documento, se entrega un conjunto de Links los cuales pueden ser utilizados para profundizar en los temas tratados.

INSTALACIÓN EN WINDOWS

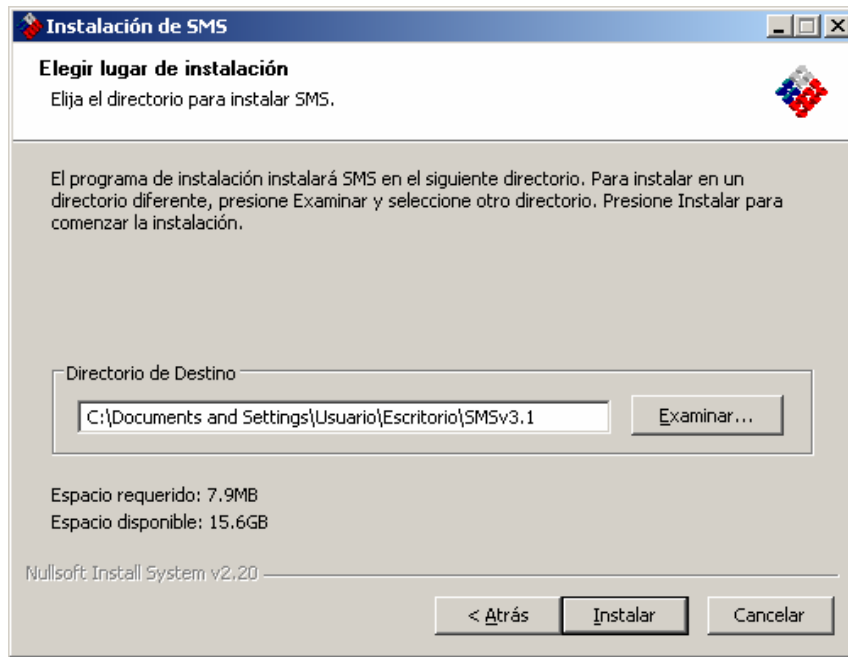
El programa requiere que en el equipo donde funcionará SMS se encuentre instalado previamente el motor Java. Este puede ser descargado desde la página www.java.com.

Obtenga el programa SMS desde la página web del Servicio www.aduana.cl, → Tramitaciones en Línea → Manifiesto Carga Electrónico → Manifiesto Carga Marítimo → Sistemas → Sistema de Mensajería por SOAP (SMS)

Ejecute el programa de instalación y siga los pasos siguientes:

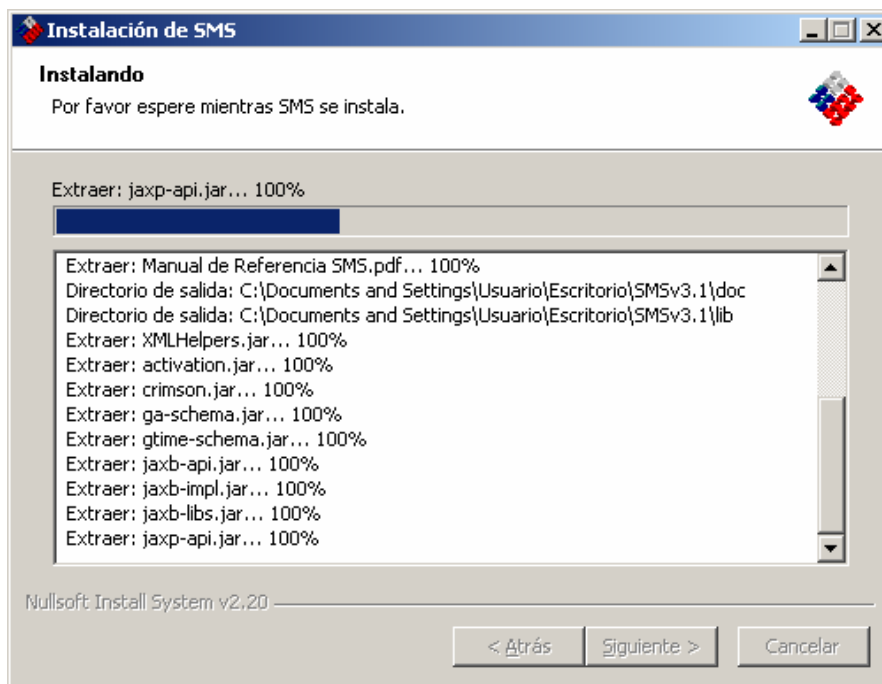


Para instalar presione el botón “Siguiente”. Con ello el programa de instalación consultará el lugar donde se ubicará el directorio de trabajo.



Puede dejar el directorio indicado o modificar la dirección. Luego presione el botón “Instalar”.

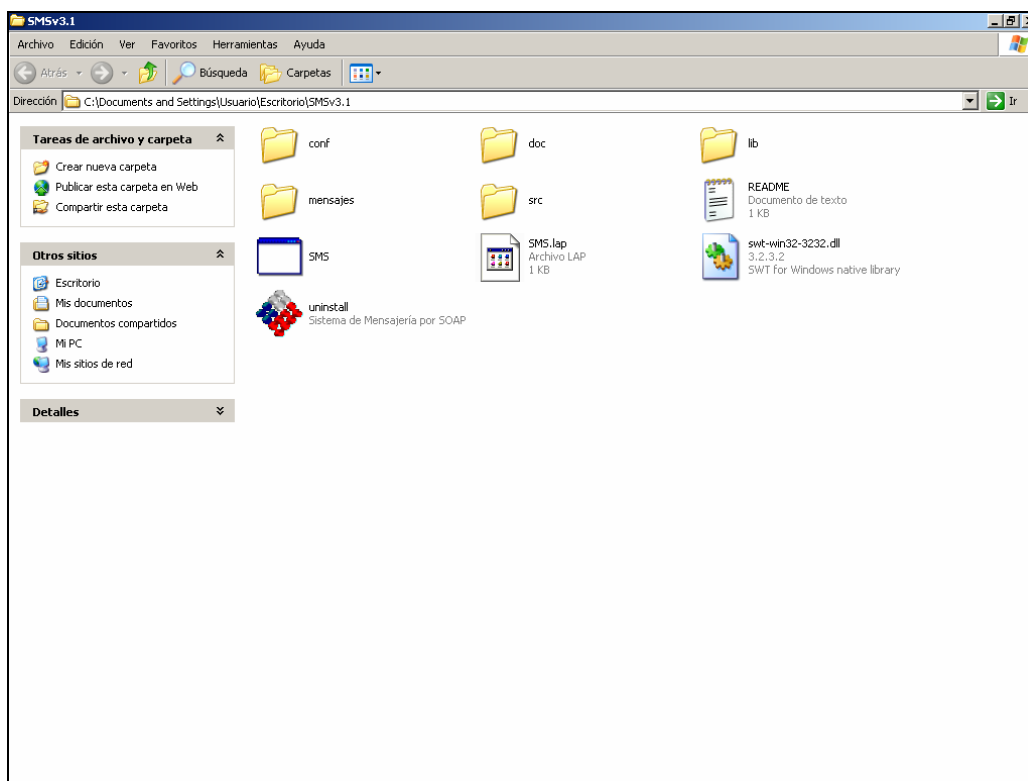
Al presionar “Instalar” el programa descomprime e instala los archivos requeridos. Al terminar presione “Siguiente”.



Para concluir la instalación presione el botón “Terminar”



El programa instalará las carpetas y archivos que se muestran en la figura siguiente:



La estructura de utilización de directorios requerida para efectuar el intercambio de mensajería se especifica en los capítulos siguientes.

Para partir el Sistema se debe ejecutar el archivo SMS.exe. Inmediatamente se activa una consola en negro y posteriormente una interfase donde se deben ingresar los datos de conexión.

Sistema de Mensajería por SOAP - SMS v3.1

Usuario:

Password:

Servidor:

Proxy IP:

Proxy Port:

SMS Home:

Tiempo:

Los datos a llenar corresponden a:

- Usuario: Login del usuario.
- Password: Clave de autorización proporcionada por Aduana.
- Servidor: Dirección electrónica donde serán enviados los documentos electrónicos.
- Proxy IP y Proxy Port: Direcciones IP y de puerto respectivamente utilizadas para invocar el servicio Proxy de una red.
- SMS Home: Lugar donde se ubican las carpetas de trabajo del sistema.
- Tiempo: Duración del intervalo de consulta al Servidor de Aduana.

NOTA: Tenga cuidado de indicar la dirección correcta de Servidor según se encuentre en las etapas de prueba o producción respectivamente. Consulte el documento “Servidores de Conexión”.

Ingresados estos datos el sistema comienza a operar siguiendo la mecánica de carpeta de mensajes a ser enviados, mensajes rechazados, aceptados, etc.

Para instalar sobre otras plataformas consultar a manifiestomaritimo@aduana.cl

Normalizaciones

Se han definido las siguientes normalizaciones en la estructura de nombre de los archivos:

1.- La estructura del nombre del archivo XML que viajará desde y hacia el servidor de Aduanas es la siguiente:

Emisor-Documento-Versión-Correlativo.XML

- **Emisor:** Sigla que identifica al emisor del mensaje (descripción libre). Formato: alfanumérico de largo máximo 4.
- **Documento:** Código que identifica el tipo de documento de acuerdo a la lista de códigos del ANEXO 51-45 del Compendio de Normas Aduaneras. Formato: alfanumérico de largo máximo 15.
- **Versión:** Identificación de la versión de la estructura del mensaje, por ejemplo 1.0. Formato: dos dígitos separados por un punto.
- **Correlativo:** número de referencia del mensaje o el identificador dado por el emisor. Formato: numérico de largo 15.

2.- El nombre de los archivos de respuesta se estructura de la siguiente forma:

respuesta-n-nombre original del mensaje.xml

- **Respuesta:** Es una constante (El nombre “respuesta”) que identifica al tipo de archivo.
- **n:** Representa un correlativo que indica el número de respuesta para el archivo original.

3.- El archivo XML con todos los tag de respuestas obtenido desde el servidor, se copia en la carpeta Terminados, con la siguiente estructura de nombre

Emisor-Documento-Versión-Número_aceptación.XML

- **Número_aceptación:** Es el número entregado por el servidor de Aduanas, cuando el documento cumple con todas las validaciones.

Las normalizaciones antes descritas, son utilizadas por MIDAS y el sistema SMS, para facilitar la gestión de los archivos.

Ciclo de vida de un mensaje

Las carpetas utilizadas en el sistema SMS son las siguientes:

Carpeta	Descripción
mensajes	En esta carpeta se deben ingresar los documentos XML, que cumplan la normalización de nombre descritas en el ítem “normalizaciones”. Esta carpeta será consultada por el sistema SMS para verificar la validez de la estructura de los archivos XML (estructura XSD) . Los archivos que superen la evaluación serán enviados, los que no, serán movidos a la carpeta invalidos .
xml/pendientes	Una vez que los archivos XML son enviados al servidor del SNA, son trasladados de la carpeta mensajes a pendientes . Los archivos permanecerán en la carpeta pendientes por las siguientes razones: <ul style="list-style-type: none"> - El archivo enviado, todavía no se procesa. - No existe conexión con el servidor. Una vez recibida una respuesta del servidor, el archivo pendiente puede pasar a la carpeta terminados o rechazados , según sea el caso.
xml/terminados	Los archivos que reciben un mensaje de “aceptación” desde el servidor del SNA, son trasladados desde la carpeta pendientes a la terminados . SMS recupera todos los nuevos tag de respuesta enviados por el servidor de Aduanas, estos tag son incorporados en el documento antes de ser copiado a la carpeta terminados . La respuesta recibida del servidor del SNA se almacena en la carpeta respuestas .
xml/rechazados	Si un archivo que fue enviado recibe una respuesta indicando la existencia de errores en el archivo enviado, se copia la respuesta recibida (El nombre tiene que cumplir lo indicado en normalizaciones) en la carpeta respuestas . A continuación se anexan tags de control en el documento original, los cuales registran los errores descritos en la respuesta del servidor del SNA.
xml/respuestas	En esta carpeta se almacenan todas las repuestas recibidas desde el servidor. Las repuestas se almacenan en archivos XML con el formato de nombres de archivos descrito en “normalizaciones”.
xml/invalido	Si un documento, antes de ser enviado, es inválido estructuralmente según validación respecto de su esquema, es movido desde carpeta mensajes a carpeta invalidos .
xml/desconocidos	Si un documento no posee registro en el servidor del SNA, será considerado como desconocido, moviéndose a carpeta de igual nombre. Lo anterior, para evitar insertar documentos no enviados en carpeta pendientes .

A continuación se presenta un grafo que describe el ciclo de vida de la mensajería por SOAP:

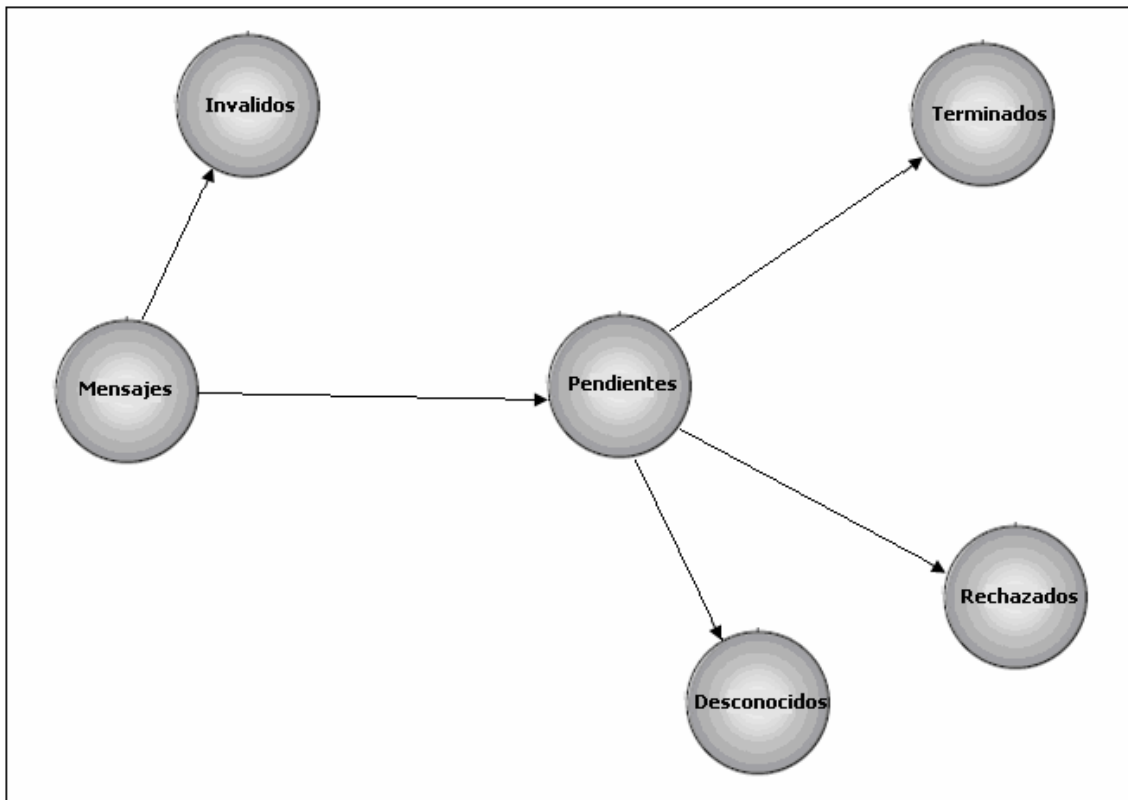


Figura 1: Grafo del ciclo de vida de los mensajes

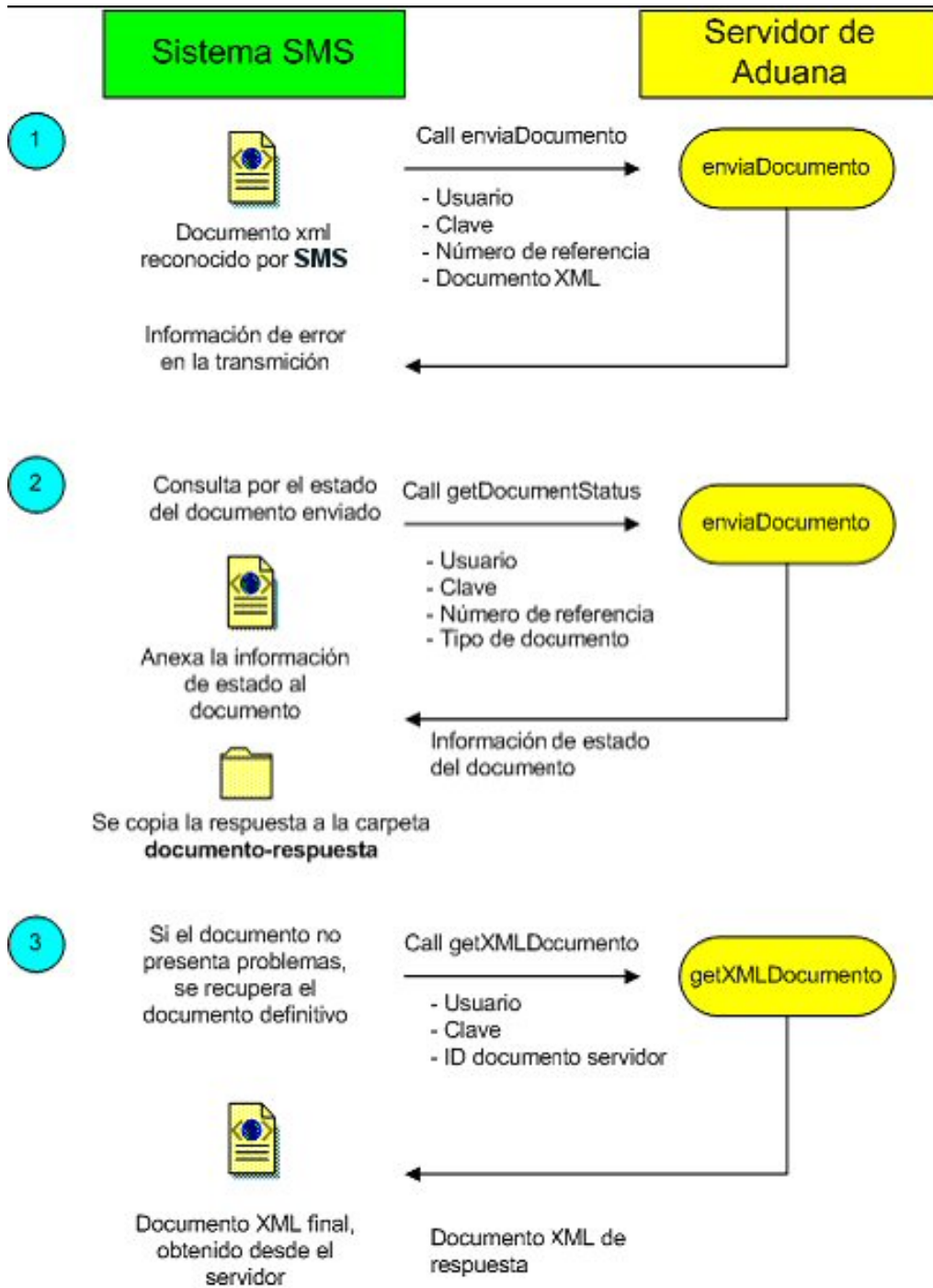


Figura 2: Esta figura muestra las llamadas desde SMS (izquierda) a los WebServices (derecha). En el diagrama se incluye los parámetros enviados, al igual que la información retornada por los WebServices y el manejo de las respuestas por parte de SMS.

Los servicios

Aduana posee un servicio para envío y otro para recepción de mensajería por SOAP. Dichos servicios son:

EnviaDocumento

Descripción:

Realiza el envío de documentos XML por SOAP.

Parámetros:

Login : Usuario del sistema de seguridad SNA
Clave : Clave del sistema de seguridad del SNA.
NumeroReferencia : Número de referencia asignado por el emisor.
Xml : String con el archivo xml que se desea enviar.

Uri:

Representa al objeto remoto, en este caso "Documentos".

Método:

El método se llama "enviaDocumento".

GetDocumentStatus

Descripción:

Realiza la consulta de las respuestas, disponibles para un envío efectuado con anterioridad.

Parámetros:

User : Usuario del sistema de seguridad SNA
Pwd : Clave del sistema de seguridad del SNA.
Referente : Número de referencia asignado por el emisor.
DocType : Tipo de documento (Ej: MIC).

Uri:

Representa al objeto remoto, en este caso "Documentos".

Método:

El método se llama "getDocumentStatus".

getXMLDocumento

Descripción:

Entrega un documento XML definitivo, con los tags de respuestas asignados por el servidor de Aduanas. Este documento, reemplaza al documento original

Parámetros:

Login : Usuario del sistema de seguridad SNA
Clave : Clave del sistema de seguridad del SNA.
Id : Id del documento entregado por el servidor, al llamar a **getDocumentStatus**

Uri:

Representa al objeto remoto, en este caso "Documentos"

Método:

El método se llama "getDocumentStatus".

Invocación de servicios

Para invocar los servicios web, se tienen que realizar los siguientes pasos:

1.- Definimos la url. Esta representa el punto final en el que reside el servicio

Ej: `URL url = new URL("http://XXXXXXXXXXXXXXXX/soap/servlet/rpcrouter");`

DONDE XXXXXXXXXXXXXXXX corresponde a dirección indicada para el proyecto.
(Ver documento Servidores de Conexión).

2.- Creamos el objeto `org.apache.soap.rpc.RPCMessage.call`. Este objeto es la interfaz principal, para ejecutar los servicios SOAP.

Ej: `Call call = new Call();`

El objeto `Call` representa la llamada a un procedimiento remoto, mediante el envío de un mensaje XML (SOAP).

3.- Definimos la URI objetivo dentro del objeto `Call` usando el método `setTargetObjectURI`

Ej: `call.setTargetObjectURI("urn:tuxpan:Documentos");`

Esto es utilizado para referenciar un objeto asociado a un descriptor de despliegue, en este caso el objeto `Documentos`.

4.- Definimos el método a invocar, dentro del objeto `Call`.

Ej: `call.setMethodName(metodo)`

Para nuestro ejemplo, la variable `metodo` podrá tomar los valores `enviaDocumento` o `getDocumentStatus`, ambos del objeto `Documentos`.

5.- Definimos la codificación

Ej: `call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC)`

6.- Asignamos los parámetros que serán enviados al servicio remoto

Ej:

```
Vector params = new Vector();
call.setParams(params);
if (nombreParametros != null) {
    for (int j=0; j<nombreParametros.length; j++) {
        params.addElement(new Parameter(nombreParametros[j], String.class,
            valorParametros[j], null));
    }
}
```

En el ejemplo, se crea un objeto **Vector** que contendrá los parámetros a enviar. Se define este objeto dentro del objeto **Call**, como portador de parámetros. Luego se llena el vector con objetos **Parameter** con nombre del parámetro, tipo de dato, valor del parámetro y null.

7.- Se invoca al servicio configurado con los pasos anteriores

Ej: `Response response = call.invoke(url, "");`

El ejemplo muestra la creación de un objeto **Response**, el cual contendrá la respuesta de la invocación. La invocación se realiza llamando al método **invoke** del objeto **Call** que antes habíamos creado.

8.- Chequeamos el objeto **Response** para ver si se ha generado algún fallo usando el método `generatedFault()`.

Ej:

```
if( !response.generatedFault() ) {
    ....
}
```

9.- Por último, si el servicio retorna alguna respuesta, recuperamos la respuesta.

Ej:

```
Parameter result = response.getReturnValue();
if (result != null)
    return result.getValue();
else
    return null;
```

Se define un objeto **Parameter** al cual le asignaremos la salida del método **getReturnValue** del objeto `response`. Luego recuperaremos el valor de la respuesta con el método **getValue** del objeto **Parameter** creado.

Los fragmentos de códigos mostrados, corresponden a la clase ServidorSOAP método invoca Servicio del sistema SMS.

Referencias

1.- Información sobre documentos de Aduana

<http://www.aduana.cl>

Entrar a la opción Tramitaciones en Línea → Manifiesto Carga Electrónico →
Manifiesto Carga Marítimo

2.- Web site de Java

<http://java.sun.com>