



Servicio de facturación electrónica del Sector Público de la CAE

Definición del API de Web Services

Interfaz ERP





Índice

1	INTRODUCCIÓN	5
2	ALCANCE	5
3	CONSIDERACIONES GENERALES SERVICIOS WEB	6
3.1	DEFINICIÓN DE SERVICIO WEB	6
3.1.1	WSDL	6
4	INTERACCIÓN CON LOS WEB SERVICES	6
4.1	AUTENTICACIÓN Y CONTROL DE ACCESO	7
4.2	IDENTIFICACIÓN DE LAS FACTURAS	9
4.3	OBJETOS RETORNADOS POR LOS WEB SERVICES	10
4.3.1	Objetos retornados por el servicio eFacturaWS - java	11
4.3.2	Objetos retornados por el servicio eFacturaWSNet - <XML>	11
5	SERVICIO WEB EFACTURAWS - JAVA	12
5.1.1	Descripción	12
5.1.2	Métodos definidos	12
5.2	LOGIN	12
5.2.1	Descripción	12
5.2.2	Parámetros de entrada	13
5.2.3	Parámetros de salida	13
5.2.4	Posibles códigos de terminación	13
5.2.5	Especificación de la librería	13
5.3	LOGOUT	14
5.3.1	Descripción	14
5.3.2	Parámetros de entrada	14
5.3.3	Parámetros de salida	14
5.3.4	Posibles códigos de terminación	14
5.3.5	Especificación de la librería	14
5.4	SENDINVOICES	15
5.4.1	Descripción	15
5.4.2	Parámetros de entrada	15
5.4.3	Parámetros de salida	16
5.4.4	Posibles códigos de terminación	16
5.5	GETINVOICESDATA	17
5.5.1	Descripción	17
5.5.2	Parámetros de entrada	17
5.5.3	Parámetros de salida	17
5.5.4	Posibles códigos de terminación	18
5.6	ADDATTACHMENT	18

5.6.1	<i>Descripción</i>	18
5.6.2	<i>Parámetros de entrada</i>	18
5.6.3	<i>Parámetros de salida</i>	19
5.6.4	<i>Posibles códigos de terminación</i>	19
5.7	LISTATTACHMENTS	19
5.7.1	<i>Descripción</i>	19
5.7.2	<i>Parámetros de entrada</i>	20
5.7.3	<i>Parámetros de salida</i>	20
5.7.4	<i>Posibles códigos de terminación</i>	20
5.8	DOWNLOADATTACHMENT	21
5.8.1	<i>Descripción</i>	21
5.8.2	<i>Parámetros de entrada</i>	21
5.8.3	<i>Parámetros de salida</i>	21
5.8.4	<i>Posibles códigos de terminación</i>	21
5.9	SETCOMMENTS	22
5.9.1	<i>Descripción</i>	22
5.9.2	<i>Parámetros de entrada</i>	22
5.9.3	<i>Parámetros de salida</i>	22
5.9.4	<i>Posibles códigos de terminación</i>	22
5.10	SETMETADATA.....	23
5.10.1	<i>Descripción</i>	23
5.10.2	<i>Parámetros de entrada</i>	23
5.10.3	<i>Parámetros de salida</i>	24
5.10.4	<i>Posibles códigos de terminación</i>	24
5.11	GETMETADATA	25
5.11.1	<i>Descripción</i>	25
5.11.2	<i>Parámetros de entrada</i>	25
5.11.3	<i>Parámetros de salida</i>	25
5.11.4	<i>Posibles códigos de terminación</i>	25
5.12	GETRECEIVEDINVOICES	26
5.12.1	<i>Descripción</i>	26
5.12.2	<i>Parámetros de entrada</i>	26
5.12.3	<i>Parámetros de salida</i>	27
5.12.4	<i>Posibles códigos de terminación</i>	27
5.13	GETSENTINVOICES.....	27
5.13.1	<i>Parámetros de entrada</i>	27
5.13.2	<i>Parámetros de salida</i>	28
5.13.3	<i>Posibles códigos de terminación</i>	28
5.14	GETINVOICESDATASTATUS	29
5.14.1	<i>Descripción</i>	29
5.14.2	<i>Parámetros de entrada</i>	29
5.14.3	<i>Parámetros de salida</i>	29
5.14.4	<i>Posibles códigos de terminación</i>	29
5.15	SETINVOICESDATASTATUS	30
5.15.1	<i>Descripción</i>	30

5.15.2	Parámetros de entrada.....	30
5.15.3	Parámetros de salida.....	30
5.15.4	Posibles códigos de terminación	30
5.15.5	Estados permitidos en el servicio setInvoicesStatus.....	31
5.16	REJECTINVOICES	32
5.16.1	Descripción	32
5.16.2	Parámetros de entrada.....	32
5.16.3	Parámetros de salida.....	32
5.16.4	Posibles códigos de terminación	32
5.17	GETINVOICES	33
5.17.1	Descripción	33
5.17.2	Parámetros de entrada.....	33
5.17.3	Parámetros de salida.....	33
5.17.4	Posibles códigos de terminación	34
5.18	SENDINVOICESSIGN	34
5.18.1	Descripción	34
5.18.2	Parámetros de entrada.....	34
5.18.3	Parámetros de salida.....	35
5.18.4	Posibles códigos de terminación	35
6	SERVICIO WEB EFACTURAWSNET - <XML>	36
6.1	LOGIN	39
6.1.1	Descripción	39
6.1.2	Parámetros de entrada.....	39
6.1.3	Parámetros de salida.....	40
6.1.4	Posibles códigos de terminación.....	41
6.2	LOGOUT	42
6.2.1	Descripción	42
6.2.2	Parámetros de entrada.....	42
6.2.3	Parámetros de salida.....	42
6.2.4	Posibles códigos de terminación.....	42
6.3	SENDINVOICESBASE64	43
6.3.1	Descripción	43
6.3.2	Parámetros de entrada.....	43
6.3.3	Parámetros de salida.....	45
6.3.4	Posibles códigos de terminación.....	45
6.4	GETINVOICESDATABASE64.....	46
6.4.1	Descripción	46
6.4.2	Parámetros de entrada.....	46
6.4.3	Parámetros de salida.....	48
6.4.4	Posibles códigos de terminación.....	49
6.5	ADDATTACHMENTBASE64	49
6.5.1	Descripción	49
6.5.2	Parámetros de entrada.....	49
6.5.3	Parámetros de salida.....	50

6.5.4	Posibles códigos de terminación.....	51
6.6	LISTATTACHMENTS.....	52
6.6.1	Descripción.....	52
6.6.2	Parámetros de entrada.....	52
6.6.3	Parámetros de salida.....	53
6.6.4	Posibles códigos de terminación.....	54
6.7	DOWNLOADATTACHMENTBASE64.....	54
6.7.1	Descripción.....	54
6.7.2	Parámetros de entrada.....	55
6.7.3	Parámetros de salida.....	56
6.7.4	Posibles códigos de terminación.....	56
6.8	SETCOMMENTS.....	57
6.8.1	Descripción.....	57
6.8.2	Parámetros de entrada.....	57
6.8.3	Parámetros de salida.....	58
6.8.4	Posibles códigos de terminación.....	59
6.9	SETMETADATA.....	60
6.9.1	Descripción.....	60
6.9.2	Parámetros de entrada.....	60
6.9.3	Parámetros de salida.....	61
6.9.4	Posibles códigos de terminación.....	62
6.10	GETMETADATA.....	63
6.10.1	Descripción.....	63
6.10.2	Parámetros de entrada.....	63
6.10.3	Parámetros de salida.....	64
6.10.4	Posibles códigos de terminación.....	65
6.11	REJECTINVOICES.....	65
6.11.1	Descripción.....	65
6.11.2	Parámetros de entrada.....	65
6.11.3	Parámetros de salida.....	66
6.11.4	Posibles códigos de terminación.....	68
6.12	GETINVOICESDATASTATUS.....	68
6.12.1	Descripción.....	68
6.12.2	Parámetros de entrada.....	69
6.12.3	Parámetros de salida.....	70
6.12.4	Posibles códigos de terminación.....	71
6.13	SETINVOICESDATASTATUS.....	71
6.13.1	Descripción.....	71
6.13.2	Parámetros de entrada.....	72
6.13.3	Parámetros de salida.....	73
6.13.4	Posibles códigos de terminación.....	74
6.14	GETINVOICESBASE64.....	75
6.14.1	Descripción.....	75
6.14.2	Parámetros de entrada.....	75
6.14.3	Parámetros de salida.....	76



6.14.4	Posibles códigos de terminación	78
6.15	GETSENTINVOICESBASE64	79
6.15.1	Descripción	79
6.15.2	Parámetros de entrada.....	79
6.15.3	Parámetros de salida	80
6.15.4	Posibles códigos de terminación	82
6.16	GETRECEIVEDINVOICESBASE64	82
6.16.1	Descripción	82
6.16.2	Parámetros de entrada.....	83
6.16.3	Parámetros de salida	84
6.16.4	Posibles códigos de terminación	86
ANEXO A	87
CONSTANTE TYPEID		87
URL	87	



1 Introducción

Este documento va dirigido especialmente al personal técnico encargado del desarrollo y/o mantenimiento de los sistemas externos que explotan y hacen uso de la plataforma, puesto que contiene información detallada y técnica del Sistema.

La arquitectura del Sistema eFactura, de forma resumida, contiene dos módulos principales:

- **Módulo Web:** Aplicación Web que da acceso a las funcionalidades de la plataforma tanto para los clientes de la misma como para los clientes externos.
- **Módulo Central:** Encargado de realizar toda la lógica de negocio, tanto de validación de los documentos recibidos, como de emisión de facturas o custodia de documentos. Contiene a su vez dos módulos:
 - **Núcleo:** Este módulo, denominado anteriormente frontend, proporciona la mayor parte de las funcionalidades del sistema eFactura. Los servicios Web definidos en este módulo podrán ser invocados desde frontales disponibles en las entidades promotoras.
 - **MOA (Módulo de Operaciones Asíncronas):** Este módulo, denominado anteriormente backend, contiene el conjunto de procesos asíncronos que ejecutarán una serie de tareas necesarias para el correcto funcionamiento del sistema. Estas tareas son fundamentalmente la emisión de facturas y la notificación de eventos a los usuarios del sistema.

Las Entidades Promotoras ofrecen este servicio de facturación electrónica a sus clientes, llamadas Entidades Adscritas, mediante aplicaciones Web o aplicaciones ERP, conectadas al sistema central vía Web Services.

Para ello Sistema eFactura proporciona un API que contienen los web services destinados a ser explotados por cada aplicación:

- **WS-API ERP:** API diseñado para ser explotado por aplicaciones ERP (web o de escritorio) para conectar, la funcionalidad que proporcionan como ERP a la entidad adscrita, con la facturación electrónica ofrecida por la plataforma.

En este documento se presentan en detalle cada uno de los métodos ofrecidos por el WS-API ERP.

2 Alcance

El objetivo de este documento es presentar en detalle cada uno de los métodos ofrecidos por los Web services destinado a los ERP, describiendo las operaciones que ofrecen, los parámetros de entrada y salida y la forma de utilizarlos.

En primer lugar, se ofrece una pequeña introducción sobre los conceptos más importantes sobre Servicios Web, para pasar después a detallar los Servicios Web de eFactura que se emplean para la interrelación con los ERP y sus operaciones. De cada una de estas operaciones se indicarán los parámetros de entrada y salida,



y se describe brevemente su funcionamiento. Los parámetros de salida hacen referencia a objetos Java que están recogidos en el Javadoc correspondiente.

Este documento contiene la descripción del interfaz ERP, y cubre la descripción de los métodos para la librería Java.

3 Consideraciones generales Servicios Web

Este capítulo describe en detalle los Web Services del Interfaz ERP que forman el componente central del sistema eFactura.

3.1 Definición de Servicio Web

Un Servicio Web es un componente software con las siguientes características:

- Es accesible a través del interface [SOAP](#) (Simple Object Access Protocol).
- Su interface se describe en un documento [WSDL](#) (Web Service Description Language).

SOAP es un protocolo de mensajería XML que forma la base de los Servicios Web. SOAP proporciona un mecanismo simple y consistente que permite a una aplicación enviar mensajes XML a otra aplicación. Un mensaje SOAP es una transmisión desde un emisor SOAP a un receptor SOAP, y cualquier aplicación puede participar en este intercambio como emisor o receptor. SOAP es un protocolo completamente independiente del protocolo de transporte subyacente, por eso los mensajes SOAP se pueden intercambiar sobre HTTP, JMS o protocolos de transporte de e-mail. Actualmente el protocolo HTTP es el único soportado por el Sistema eFactura.

WSDL es un documento XML que contiene un conjunto de definiciones que describen un Servicio Web. Proporciona toda la información necesaria para acceder y utilizar un Servicio Web. Un documento WSDL describe qué hace el Servicio Web, cómo se comunica, y dónde reside. Usamos el documento WSDL en el momento del despliegue para crear nuestros interfaces de servicio.

3.1.1 WSDL

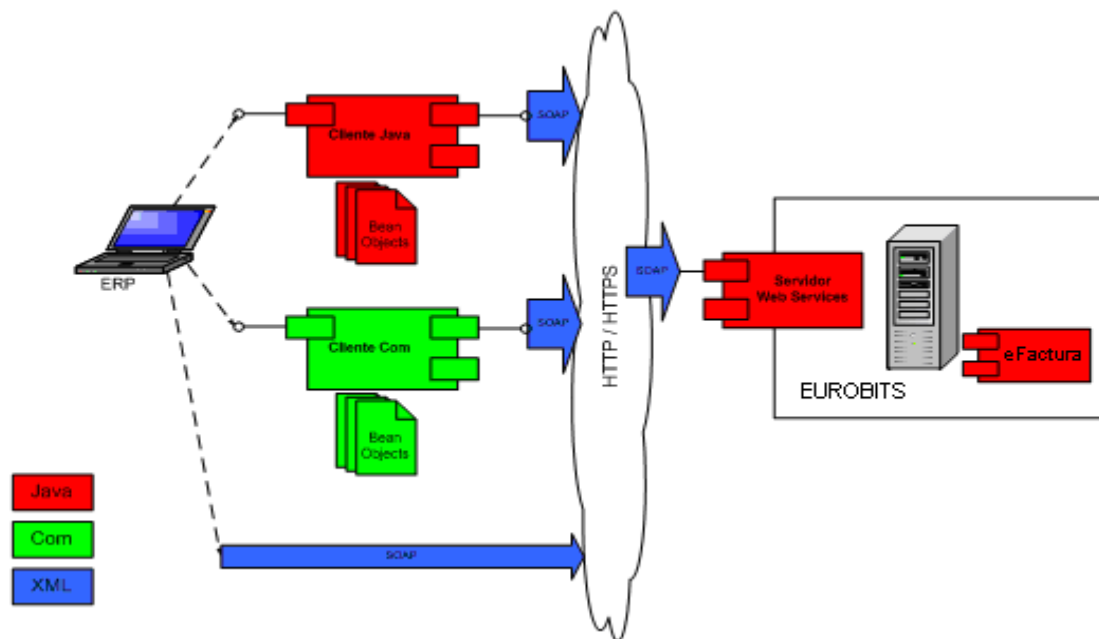
El WSDL (Web Services Description Language, Lenguaje de Descripción de Servicios Web) describe el comportamiento de los servicios Web e instruye en la forma de interactuar con ellos. Se muestra como un archivo con formato XML para cada servicio Web.

4 Interacción con los Web services

A continuación pasamos a detallar cómo se realiza la interacción con los Web services de eFactura. Existe una librería cliente (Java) que sirve de unión entre el ERP que es origen de la invocación y los Web services propiamente dichos. Esta librería posee un método por cada uno de los servicios que se ofrecen: *login*,

logout, Cada uno de estos métodos del cliente realiza una invocación a la operación correspondiente del servicio Web a través de una petición SOAP con los parámetros proporcionados, y obtiene la respuesta que devuelve en forma de objeto.

En el siguiente gráfico se la interacción entre los nuevos servicios Web y los componentes clientes que será facilitada para la integración con los mismos.



Junto con las librerías cliente se suministra un objeto que contiene todos los códigos de error catalogados, así como los valores admisibles para determinados parámetros de los Web Services. Para el caso de las aplicaciones que invoquen directamente al servicio mediante SOAP sin utilizar las librerías clientes se suministrará la documentación oportuna.

4.1 Autenticación y control de acceso

Para la autenticación en los servicios ofrecidos, se emplea un **ticket de sesión** que se obtiene mediante un servicio de **login** y que deberá emplearse en los subsiguientes peticiones al resto de métodos que se ofrecen a través del servicio *eFacturaWS* o *eFacturaXMLWS*.

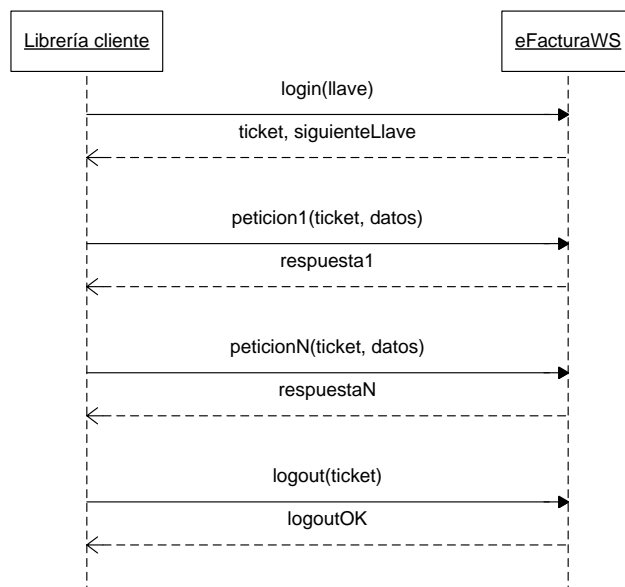
Mediante el ticket el servicio es capaz en todo momento de identificar al usuario que realiza la petición. El usuario puede utilizar el ticket todas las veces que sea preciso dentro del período de *timeout* establecido por el servidor. Cuando se considere que la sesión ha terminado, es preciso realizar un **logout** con el fin de asegurarse que el ticket será invalidado.

En todos los demás servicios proporcionados se incluye un parámetro de tipo cadena de texto con el ticket obtenido mediante el método *login*, el cual permite autenticar al usuario y mantener la sesión para el mismo. Según se ha indicado anteriormente, dicho ticket de sesión debe ser invalidado al terminar de trabajar



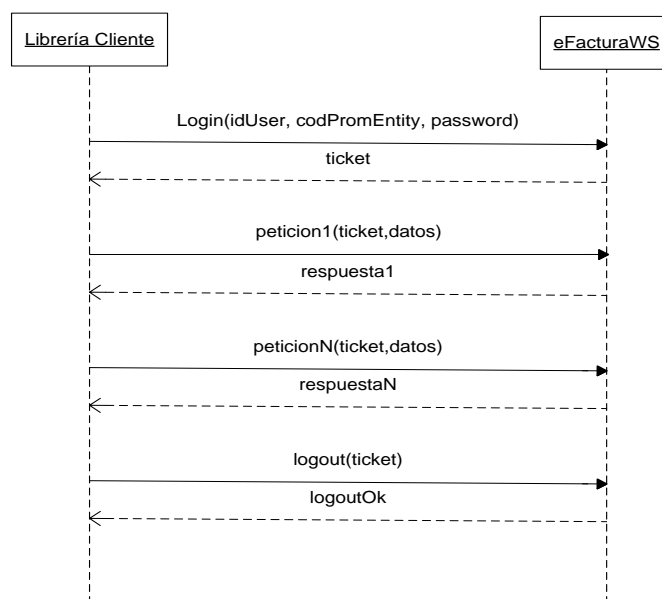
con el mismo mediante la invocación al servicio *logout*. No se servirá ninguna petición que no incluya un ticket de sesión válido.

Se presenta a continuación un diagrama de secuencia que describe el proceso de autenticación y posterior manejo del ticket de sesión proporcionado por el mismo:



El método *login* invalida la **llave** que se proporciona al mismo y devuelve la que deberá emplearse en la próxima invocación al mismo (mecanismo *One-time-password*), todo esto si el usuario tiene configurado la opción OTP como true (Opción por defecto), de lo contrario no se invalida. Para obtener la primera llave es preciso realizar el alta del servicio a través del portal de la Entidad Promotora correspondiente. Esta llave no estará en ningún caso relacionada con el login y usuario de la entidad para prevenir posibles casos de phishing.

Además de este login hay disponible otro tipo de login que recibe como parámetros **el cif, código de la entidad promotora y un password**, a continuación se presenta un diagrama de secuencia que describe el proceso de autenticación y posterior manejo del ticket de sesión proporcionado por el mismo:



4.2 Identificación de las facturas

Es necesario destacar que en todos los servicios que desean consultar, actualizar el estado o descargar facturas, la forma de identificar unívocamente una factura es indicando:

- **guidDefinitivo** (o número definitivo de factura) y **CIFProveedor**
- **Id y TypeId**

Esto es debido a que en eFactura se utiliza una numeración para las facturas basada en un número de serie (dependiente del tipo de factura) y número de factura correlativo para cada proveedor. Para un mismo proveedor, no podrán existir dos facturas con el mismo número de serie y número de factura.

La segunda opción es válida para enviar solicitar información enviando un Id y su respectiva tipología, teniendo en cuenta que typeId especifica qué tipo de identificador se está utilizando: ID_FISCAL, ID_INTERNO, ID_EXTERNO.

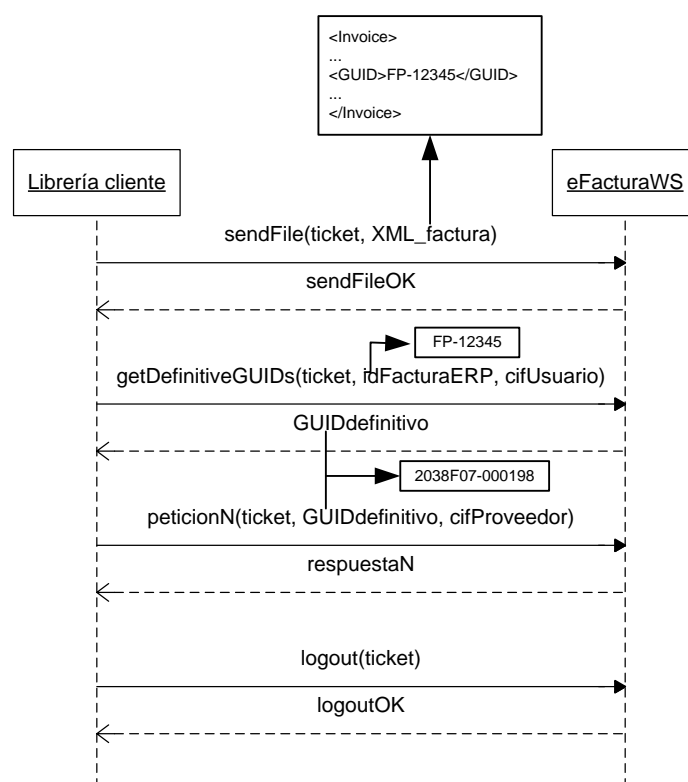
De acuerdo con lo anterior, para identificar una factura deberemos indicar los dos datos referidos al principio de un apartado (guidDefinitivo -o número de serie- y CIFProveedor).

Respecto a los códigos que identificarán las facturas (guidDefinitivo), consideramos importante realizar las siguientes puntualizaciones:

- A la hora de importar una factura mediante el servicio **sendFile**, habrá que indicar en el campo Invoice/GUID del XML en formato UBL1.0 el código de dicha factura en el ERP (idFacturaERP).

- Una vez importada una factura (servicio *sendFile*) será preciso invocar al servicio *getDefinitiveGUIDs* para obtener el identificador que se ha asignado a la factura tras su emisión, **guidDefinitivo**, formado por un número de serie y código. Un mismo proveedor no puede tener dos facturas con la misma serie y código de factura, por lo que estos datos (número de serie y código de factura más el CIF del proveedor) serán los utilizados en el resto de servicios para acceder a las facturas. Estos guidDefinitivos serán los que habrá que proporcionar a los servicios en el parámetro *guidDefinitivo*.

En el siguiente diagrama se muestra el proceso de envío de una factura, la recuperación de su GUIDdefinitivo y la realización de una operación referente a dicha factura:



4.3 Objetos retornados por los Web Services

En las versiones anteriores a la v2.1 de la plataforma, todos los objetos retornados por los servicios web de eFactura eran objetos *Bean*, es decir, objetos que contienen datos y métodos de acceso a los mismos, de modo que la información retornada esté claramente tipada y definida.

Desde dicha versión, se ha creado un nuevo servicio eFacturaXMLWS que permite acceder a las funcionalidades para ERP de la plataforma utilizando XML en lugar de objetos *Bean*. La lógica de negocio que atiende dichas peticiones es la misma que

para el servicio eFacturaWS, cambiando únicamente la forma de realizar las consultas y de tratar las respuestas del servicio.

4.3.1 Objetos retornados por el servicio eFacturaWS - java

El objeto devuelto por el Web service tiene siempre tres métodos, *getError*, *getSubCode* y *getMessage* que devuelve el **resultado de la invocación al servicio**, así como información adicional sobre los errores que se hayan podido producir en la misma. Mediante el método *getError* se puede saber si la operación solicitada ha terminado correctamente o no. En caso de haber ocurrido un error, mediante *getSubCode* es posible obtener un código asociado al tipo de error y mediante *getMessage* se obtiene una explicación textual de la causa del error.

Adicionalmente, el objeto devuelto por el servicio puede contener **información adicional sobre el resultado de la operación** solicitada.

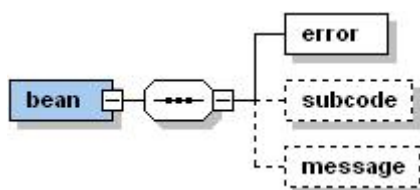
Los códigos devueltos por los Web Services están establecidos de antemano para que las aplicaciones invocantes sean capaces de decidir qué comportamiento deben seguir a partir de los mismos.

Los objetos de resultado pueden ser **Bean simples** (con los tres métodos de acceso al resultado de la invocación del servicio más los métodos de acceso a los datos), o bien objetos **BeanSet**, compuestos por los tres métodos de acceso al resultado de la invocación del servicio más un método que devolverá un vector de objetos *Bean* con la información solicitada. Nótese que **únicamente el objeto devuelto directamente por el Web service contará con la información del resultado de la invocación**, en caso de que éste contenga un vector de objetos *Bean*, éstos contendrán solamente la información proporcionada por el servicio.

4.3.2 Objetos retornados por el servicio eFacturaWSNet - <XML>

Este servicio puede retornar objetos de dos tipos:

1. Un `java.lang.String` con formato XML que entre otros contendrá una rama `bean` que informará sobre la terminación de la invocación:



- `error`: nos dirá si se ha producido un error en la invocación o no, pudiendo tomar como valores `false` o `true`.
 - `subCode`: puede tomar valores numéricos y en caso de error contendrá un código asociado al tipo de error
 - `message`: tomará valores de cadena y en caso de error contendrá una explicación contextual de la causa del error.
2. `javax.activation.DataHandler`: que contendrá un archivo o en caso de error saltará una excepción con un mensaje descriptivo del error si se conoce.



5 Servicio Web eFacturaWS - java

5.1.1 Descripción

A continuación, se muestra una descripción detallada de los métodos que ofrece el Web service eFacturaWS.

Para cada uno de los métodos se indica:

1. Una breve descripción funcional del mismo
2. Sus parámetros de entrada (clase y nombre del parámetro), descritos en el mismo orden que tienen en la declaración del método, con una descripción funcional de los mismos. En el caso de que sus valores estén acotados, un listado de dichos valores.
3. Su parámetro de salida, indicando la clase a la que pertenecen y describiendo su funcionalidad. La información de estos objetos se completa con la Javadoc correspondiente.

Posibles códigos de terminación, indicando los valores que pueden tomar los campos descritos en el apartado 4.3 para dicho método.

5.1.2 Métodos definidos

A continuación se describen en detalle todos los métodos que forman parte del Servicio Web. Para más información consultar el *javadoc*.

5.2 Login

5.2.1 Descripción

Este método identifica y autentica ante Sistema eFactura a la entidad adscrita que desea utilizar los servicios de la plataforma mediante el WS-API ERP.

Permite a una entidad adscrita obtener un ticket de sesión válido para ser utilizado durante un tiempo limitado en las siguientes llamadas. Para ello debe proporcionarse una contraseña que fue proporcionada por el usuario a la entidad promotora mediante el módulo web.

Para información más detallada sobre el mecanismo de autenticación y control de acceso del Web service eFacturaWS, consultar el apartado 4.1 Autenticación y Control de Acceso.

Si el usuario ya está logado, por ejemplo mediante otra petición de login anterior, el nuevo login no inhabilita el ticket previo, por lo que la sesión anterior se puede seguir utilizando.

5.2.2 Parámetros de entrada

Parámetro	Obligatorio (S/N)	Descripción
userCIF	S	java.lang.String que contiene una cadena de texto que representa el identificador de la entidad adscrita (CIF, NIE, NIF...).
codPromEntity	S	java.lang.String que contiene una cadena de texto que representa el código de la entidad promotora a la que se le proporcione la contraseña.
password	S	java.lang.String que contiene una cadena de texto con la contraseña que se proporcionó a la entidad promotora.

5.2.3 Parámetros de salida

Parámetro	Descripción
<i>LoginBean</i>	Entre sus atributos contiene un ticket válido para las siguientes peticiones, en esta ocasión la llave cifrada tiene el valor null. Así como información sobre el resultado de la invocación al Web service que contendrá la información de si la operación se ha completado con éxito o si por el contrario ha ocurrido algún error, en cuyo caso se indicará el error en un mensaje

5.2.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_006	El login proporcionado es erróneo

5.2.5 Especificación de la librería

Al igual que para el resto de métodos la librería JAVA proporciona algunas ventajas para el usuario sobre el uso directo de la operación del web services.



Este método además de identifica y autentica ante Sistema eFactura a la entidad adscrita almacena internamente el ticket de forma que las demás funciones lo utilizan sin necesidad de que el usuario tenga que estar manejando el ticket en las siguientes peticiones. Por ello, esta función también requiere que se le pase como parámetro de entrada la "url" de conexión con los web services del WS-API ERP.

5.3 Logout

5.3.1 Descripción

Mediante este método se desconecta del Sistema eFactura.

5.3.2 Parámetros de entrada

Parámetro	Obligatorio (S/N)	Descripción
ticket	S	java.lang.String contiene una cadena con formato XML.

5.3.3 Parámetros de salida

Parámetro	Descripción
LogoutBean	Objeto LogoutBean que envía la respuesta al cliente.

5.3.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada

5.3.5 Especificación de la librería

Al igual que para el resto de métodos la librería JAVA proporcionadas también proporciona esta, aportando algunas ventajas para el usuario sobre el uso directo de la operación del web services.



Este método además de identifica y autentica ante Sistema eFactura a la entidad adscrita almacena internamente el ticket de forma que las demás funciones lo utilizan sin necesidad de que el usuario tenga que estar manejando el ticket en las siguientes peticiones. Por ello, esta función también requiere que se le pase como parámetro de entrada la "url" de conexión con los web services del WS-API ERP.

5.4 sendInvoices

5.4.1 Descripción

Este servicio permite la importación de un fichero XML con una factura en formato FACTURAE32, o bien un archivo zip con varias facturas en dichos formatos. Devuelve un objeto *SendFileBeanSet* que aporta información sobre las facturas importadas y el estado de la importación. Los estados en los que puede quedar la factura tras la importación están definidos de antemano de tal forma que a partir de esta información la aplicación pueda decidir como comportarse al respecto.

El sistema emitirá automáticamente las facturas incorporadas mediante este método, asignándoles número de serie y factura (salvo error de formato de las mismas o del proceso de emisión).

5.4.2 Parámetros de entrada

- java.lang.String ticket: ticket de sesión válido.
- javax.activation.DataHandler ficheroEntrada : DataHandler con el fichero de facturas a importar.
- java.lang.String tipoFactura: Tipo de factura que se desea importar a eFactura. Podrá tomar los valores "380" para facturas, "389" para autofacturas, o bien "384" para rectificativas, valores todos ellos catalogados en archivo de constantes (constantes *WS_INVOICE_TYPE_FACTURA*, *WS_INVOICE_TYPE_AUTOFACTURA*, *WS_INVOICE_TYPE_FACTURA_RECTIFICATIVA*, *WS_INVOICE_TYPE_AUTOFACTURA_RECTIFICATIVA*). Para el caso de las facturas rectificativas, se indica siempre el tipo "384" siendo el sistema internamente quien detecta si se trata de una factura o de una autofactura a partir del código de la factura que se pretende rectificar. Los valores de este campo **no** se chequean (se coge el tipo de factura directamente de la factura), únicamente se tiene en cuenta si se pasa "1" y la factura no es factura "4" y no es autofactura. De todos modos, se recomienda mantener la especificación como está y estudiar el validar estos valores.



- `java.lang.String` formato: Formato en que se incorporan al sistema las facturas. Únicamente se permiten los siguientes valores "FACTURAE32" valores catalogados en archivo de constantes.

IDENTIFICADOR	LITERAL
CM10	CM10
FACTURAE30	FACTURAE 3.0
FACTURAE31	FACTURAE 3.1
FACTURAE32	FACTURAE 3.2
PROPIETARIO	Formato propietario
UBL10	UBL 1.0

Estos formatos se encuentran configurados en la tabla formato.

5.4.3 Parámetros de salida

Devuelve un objeto *SendFileBeanSet* con información sobre el resultado de la invocación al Web service y un vector de objetos *SendFileBean* con la información particular del resultado de la importación de cada una de las facturas importadas. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

En este caso concreto, es necesario distinguir entre el resultado de la invocación al servicio (obtenida a partir de los métodos heredados de la clase Bean y que puede tomar los valores descritos en el siguiente punto) y de la importación de cada una de las facturas del zip enviado (descrito en cada uno de los objetos *SendFileBean* devueltos por el servicio).

5.4.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_012	Ya existen facturas con las series o número de factura solicitadas

WS_ERROR_014	Debe configurar el separador entre la serie y el número de la factura
WS_ERROR_017	Error con el prefijo serie. Tiene que tener entre 1 y 4 caracteres alfanumericos
WS_ERROR_018	Error, el prefijo serie no está asociado a la delegación correspondiente
WS_ERROR_019	Error con el prefijo serie

5.5 getInvoicesData

5.5.1 Descripción

Este servicio puede ser invocado para descargar facturas que han sido emitidas en el sistema. Permite obtener un archivo zip con las facturas solicitadas a través de su GUIDdefinitivo.

5.5.2 Parámetros de entrada

- java.lang.String ticket: ticket de sesión válido.
- java.lang.String [] guidsDefinitivos: Identificador universal de la facturas que se quieren descargar (obtenido mediante *getDefinitiveGUIDs*)..
- java.lang.String [] CIFsProveedor: CIF del proveedor de la facturas que se quieren descargar. Es necesario crear una entrada en este vector por cada uno de los *guidDefinitivos* indicados en el anterior parámetro.
- java.lang.String formato: Formato en que sean descargar las facturas indicadas. Se permiten los siguientes valores catalogados en archivo de constantes:

IDENTIFICADOR	LITERAL
CM10	CM10
COPIA	Copia del original firmada digitalmente
PDF	Un SOLO fichero PDF
PROPIETARIO	Formato propietario
UBL10	UBL 1.0
FACTURAE30	FACTURAE 3.0
FACTURAE31	FACTURAE 3.1
FACTURAE32	FACTURAE 3.2

Estos formatos provienen de la tabla formato.

5.5.3 Parámetros de salida

Devuelve un objeto *DataHandlerBean* con información del resultado de la invocación al servicio y un *DataHandler* con un zip que contiene los XMLs en el formato seleccionado de las facturas que se están solicitando mediante su



GUIDdefinitivo. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.5.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos

5.6 addAttachment

5.6.1 Descripción

Servicio que permite adjuntar un fichero a una factura existente en el sistema. Puede usarse para asociar a una factura ficheros que permitan aclarar su contenido o que por cualquier causa se desea que permanezcan asociados a la misma y accesibles desde su detalle. No hay límite en el número de ficheros que se pueden adjuntar a una factura, si bien hay un límite en tamaño total de adjuntos a una factura (en la actualidad, 512k).

5.6.2 Parámetros de entrada

- `java.lang.String ticket`: ticket de sesión válido.
- `java.lang.String guidDefinitivo`: Identificador universal de la factura a la que se quiere adjuntar el archivo (obtenido mediante el método `getDefinitiveGUIDs`).
- `javax.activation.DataHandler ficheroEntrada`: Objeto *DataHandler* con el fichero que se desea adjuntar.
- `java.lang.String descripcion`: Descripción del adjunto.
- `java.lang.String nombreFichero`: Nombre del fichero que se adjunta.



- `java.lang.String CIFProveedor`: CIF del proveedor de la facturas que se quieren consultar.

5.6.3 Parámetros de salida

Un objeto *AddAttachmentBean* con información sobre el resultado de la invocación al Web service y sobre el resultado de la inclusión del fichero adjunto. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.6.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_010	No se ha podido obtener el identificador de la factura
WS_ERROR_025	Error. Se ha intentado adjuntar un fichero a una factura en custodia

El objeto retornado por este método, además de la información genérica sobre el resultado de la invocación al Servicio web, posee dos campos para devolver información específica sobre el resultado de la operación (*errorOperativa*, *messageOperativa*). Si *errorOperativa* es *true*, significa que el fichero no se ha adjuntado debido al error indicado en *messageOperativa*.

5.7 listAttachments

5.7.1 Descripción

Método de consulta que devuelve una lista con la información de los ficheros adjuntos de una factura determinada.



5.7.2 Parámetros de entrada

- `java.lang.String ticket`: ticket de sesión válido.
- `java.lang.String guidDefinitivo`: Identificador universal de la factura a la que se quiere adjuntar el archivo (obtenido mediante el método `getDefinitiveGUIDs`).
- `java.lang.String CIFProveedor`: CIF del proveedor de la facturas que se quieren consultar.

5.7.3 Parámetros de salida

Un objeto `AttachmentFileBeanSet` con información del resultado de la invocación al servicio y un vector de objetos `AttachmentFileBean` que contendrán información acerca de los ficheros adjuntos de la factura consultada. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.7.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo `error` del objeto devuelto por el servicio (se puede consultar con el método `getError` del mismo) será `false`. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_010	No se ha podido obtener el identificador de la factura



5.8 downloadAttachment

5.8.1 Descripción

Método que permite descargar un fichero adjunto, indicando la factura a la que pertenece y el número de adjunto.

5.8.2 Parámetros de entrada

- `java.lang.String ticket`: ticket de sesión válido.
- `java.lang.String guidDefinitivo`: Identificador universal de la factura de la que se quiere descargar el archivo (obtenido mediante el método `getDefinitiveGUIDs`).
- `java.lang.Long idAdjunto`: Número de adjunto de dicha factura que se desea descargar.
- `java.lang.String CIFProveedor`: CIF del proveedor de la facturas que se quieren consultar.

5.8.3 Parámetros de salida

Un objeto `DataHandlerBean` con información del resultado de la invocación al servicio y el fichero adjunto solicitado. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.8.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo `error` del objeto devuelto por el servicio (se puede consultar con el método `getError` del mismo) será `false`. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos



WS_ERROR_010	No se ha podido obtener el identificador de la factura
WS_ERROR_011	No se ha podido obtener el contenido del fichero adjunto

5.9 setComments

5.9.1 Descripción

Servicio que permite insertar una observación a una factura indicando el `guidDefinitivo`, el `cifProveedor` de esta, el tipo de observación y el texto de la observación.

5.9.2 Parámetros de entrada

- `java.lang.String ticket`: ticket de sesión válido.
- `java.lang.String guidDefinitivo`: Identificador universal de la factura a la que se quiere establecer la observación (obtenido mediante el método `getDefinitiveGUIDs`).
- `java.lang.String cifProveedor`: CIF del proveedor de la factura a la que se quiere establecer la observación.
- `java.lang.String commentType`: Identificador del tipo de observación. Ha de ser uno de los siguientes valores:
 - 1 observación de rechazo
 - 2 observación de error
 - 3 observación de recepción
 - 4 observación de concepto receptor).
- `java.lang.String comment`: Texto de la observación.

5.9.3 Parámetros de salida

Un objeto `CommentsBeanSet` con información sobre el resultado de la invocación al Web service así como un vector de objetos `CommentsBean` con información de la observación insertada. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.9.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo `error` del objeto devuelto por el servicio (se puede consultar con el método `getError` del



mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_015	Tipo de observación no permitida
WS_ERROR_010	No se ha podido obtener el identificador de la factura

5.10 setMetaData

5.10.1 Descripción

Esta operación permite asociar información adicional (metadato) a un documento, de forma que sea almacenada por la plataforma para su posterior consulta o modificación.

La información almacenada se compone de una pareja: clave, valor. Utilizándose la clave como identificador del tipo de información adicional, por ejemplo: "Fecha Registro", actuando valor como la información adicional, por ejemplo: "28-03-2010".

Si en el momento de asociar a un documento un metadato que ya tiene asociado, existe un metadato con la misma clave, es decir, dicho documento ya tiene información asociada a la misma clave, el valor será remplazado

Las limitaciones de la información almacenada son las siguientes:

- 20 byte (alfanuméricos) para la clave informada.
- 100 byte (alfanuméricos) para el valor informado.

En ambos casos se prohíben los caracteres raros: ? ¿ ! ¡ []... y no se recomienda el uso de tildes...

5.10.2 Parámetros de entrada

Parámetro	Obligatorio (S/N)	Descripción
-----------	-------------------	-------------

ticket	S	java.lang.String ticket de sesión válido que autoriza la operación.
keyData	S	java.lang.String Nombre de la información adicional, que se utilizará como clave para recuperar el valor posteriormente.
valueData	S	java.lang.String Valor de la información adicional asociado a keyData.
typeld	S	java.lang.String Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver anexo A.
ids	S	java.util.List Lista de identificadores de documentos, cuyo valor será asociado al typeld.
sellerCIF	N	java.lang.String Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como typeld se ha especificado ID_FISCAL.

5.10.3 Parámetros de salida

Parámetro	Descripción
<i>IdInvoiceBeanSet</i>	Un objeto con información sobre el resultado de la invocación al Web service así como un vector de objetos <i>IdInvoiceBean</i> con información de las facturas relacionadas a la inserción. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.10.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif:..
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.



5.11 getMetaData

5.11.1 Descripción

Esta operación permite obtener el valor de la información adicional (metadato) asociada a un documento.

Si el documento no contiene ninguna información adicional con la clave especificada la operación no devolverá nada.

5.11.2 Parámetros de entrada

Parámetro	Obligatorio (S/N)	Descripción
ticket	S	java.lang.String ticket de sesión válido que autoriza la operación.
keyData	S	java.lang.String Nombre de la información adicional, que se utilizará como clave para recuperar el valor posteriormente.
typeld	S	java.lang.String Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver anexo A.
Ids	S	java.util.List Lista de identificadores de documentos, cuyo valor será asociado al typeld.
sellerCIF	N	java.lang.String Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como typeld se ha especificado ID_FISCAL.

5.11.3 Parámetros de salida

Un objeto *IdInvoiceBeanSet* con información sobre el resultado de la invocación al Web service así como un vector de objetos *IdInvoiceBean* con información de las facturas relacionadas a la inserción. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.11.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación

	solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif: ..
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.

5.12 getReceivedInvoices

5.12.1 Descripción

Este servicio devuelve la relación de facturas recibidas entre dos fechas indicadas y que están en un estado determinado. Las facturas se devuelven en el formato especificado codificadas en formato binario DataHandler (más eficiente que Base64) en un fichero ZIP, en caso de no especificar ningún formato se devuelve una colección de beans que resumen el contenido de la factura.

Todas las facturas devueltas quedan marcadas en el servidor como descargadas automáticamente por ERP. El parámetro de tipo booleano `onlyNew` permite filtrar de la relación de facturas recibidas que cumplen el filtro o aquellas que no han sido descargadas anteriormente desde ERP.

5.12.2 Parámetros de entrada

- `java.lang.String ticket`: ticket de sesión válido.
- `java.util.Date initDate`: Fecha inicial de intervalo de búsqueda de facturas recibidas.
- `java.util.Date endDate`: Fecha final de intervalo de búsqueda de facturas recibidas.
- `int statusCode`: Estado de la factura.
- `java.lang.String downloadFormat`: Formato en que sean descargar las facturas indicadas. Se permiten los siguientes valores catalogados en archivo de constantes:

IDENTIFICADOR	LITERAL
CM10	CM10
COPIA	Copia del original firmada digitalmente
PDF	Un SOLO fichero PDF
UBL10	UBL 1.0
FACTURAE30	FACTURAE 3.0
FACTURAE31	FACTURAE 3.1
FACTURAE32	FACTURAE 3.2

Estos formatos provienen de la tabla formato.

- `boolean onlyNew`: Valor booleano que indica si se desean descargar todas las facturas recibidas entre `initDate` y `endDate` y que aún no han sido descargadas por el ERP (valor `true`) o bien si se desean descargar todas las



facturas comprendidas entre dichas fechas independientemente de si han sido descargadas con anterioridad (valor *false*).

5.12.3 Parámetros de salida

Un objeto *InvoiceDataBeanSet* con información sobre el resultado de la invocación al Web service así como un vector de objetos *InvoiceDataBean* con información de las facturas recibidas, un objeto *DataHandler* que contendrá un archivo ZIP con las facturas recibidas, un boolean que indicara si existen mas facturas pendientes por descargar. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.12.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_024	Error al intentar descargar facturas en formato:..
WS_ERROR_027	Warning. No existen datos para los parámetros solicitados

5.13 getSentInvoices

Este servicio devuelve la relación de facturas enviadas entre dos fechas indicadas y que están en un estado determinado. Las facturas se devuelven en el formato especificado codificadas en formato binario *DataHandler* (más eficiente que Base64) en un fichero ZIP, en caso de no especificar ningún formato se devuelve una colección de beans que resumen el contenido de la factura.

Todas las facturas devueltas quedan marcadas en el servidor como descargadas automáticamente por ERP. El parámetro de tipo booleano *onlyNew* permite filtrar de la relación de facturas enviadas que cumplen el filtro o aquellas que no han sido descargadas anteriormente desde ERP.

5.13.1 Parámetros de entrada

Parámetro	Obligatorio (S/N)	Descripción
initDate	S	java.util.Date Fecha inicial de intervalo de búsqueda de facturas enviadas.
endDate	S	java.util.Date Fecha final de intervalo de búsqueda de facturas enviadas.
statusCode	S	int Estado en el que deben estar las facturas enviadas.
downloadFormat	S	java.lang.String Formato en que sean descargar las facturas indicadas. Se permiten los siguientes valores catalogados en archivo de constantes.
onlyNew	S	boolean Indica si se desean descargar todas las facturas enviadas entre initDate y endDate y que aún no han sido descargadas por el ERP (valor true) o bien si se desean descargar todas las facturas comprendidas entre dichas fechas independientemente de si han sido descargadas con anterioridad (valor false).

IDENTIFICADOR	LITERAL
CM10	CM10
COPIA	Copia del original firmada digitalmente
PDF	Un SOLO fichero PDF
UBL10	UBL 1.0
FACTURAE30	FACTURAE 3.0
FACTURAE31	FACTURAE 3.1
FACTURAE32	FACTURAE 3.2

5.13.2 Parámetros de salida

Parámetro	Descripción
<i>InvoiceDataBeanSet</i>	información sobre el resultado de la invocación al Web service así como un vector de objetos InvoiceDataBean con información de las facturas recibidas, un objeto DataHandler que contendrá un archivo ZIP con las facturas recibidas, un boolean que indicara si existen mas facturas pendientes por descargar. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.13.3 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_024	Error al intentar descargar facturas en formato:..

5.14 getInvoicesDataStatus

5.14.1 Descripción

Servicio que permite obtener el estado de un grupo de facturas.

5.14.2 Parámetros de entrada

- java.lang.String ticket: ticket de sesión válido.
- java.lang.String typeId: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver **anexo A**.
- java.util.List ids: Lista de ids de facturas, cuyo valor será asociado al typeId.
- java.lang.String sellerCIF: Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como typeId se ha especificado ID_FISCAL.

5.14.3 Parámetros de salida

Un objeto *InvoiceStatusDataBeanSet* con información sobre el resultado de la invocación al Web service así como un vector de objetos *InvoiceStatusDataBean* con información de las facturas consultadas. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.14.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o

	el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif: ..
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.

5.15 setInvoicesDataStatus

5.15.1 Descripción

Servicio que permite cambiar el estado de una factura o varias facturas, indicando el id, y su tipo de Identificador de ID.

5.15.2 Parámetros de entrada

- java.lang.String ticket: ticket de sesión válido.
- int statusCode: Estado de la factura.
- java.lang.String typeId: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver **anexo A**.
- java.util.List ids: Lista de ids de facturas, cuyo valor será asociado al typeId.
- java.lang.String sellerCIF: Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como typeId se ha especificado ID_FISCAL.

5.15.3 Parámetros de salida

Un objeto *InvoiceStatusDataBeanSet* con información sobre el resultado de la invocación al Web service así como un vector de objetos *InvoiceStatusDataBean* con información de las facturas consultadas. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.15.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o

	el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_005	Cambio de estado no permitido
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif: ..
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.

5.15.5 Estados permitidos en el servicio setInvoicesStatus

En el servicio *setInvoicesStatus*, y para mantener la coherencia de los estados de las facturas en BBDD, únicamente se permitirá modificar las facturas a los siguientes estados:

ESTADOS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	29	30	31	32	33	34	
Sin Estado Previo																														
Sin Emitir	E																													
Pendiente de Emisión																														
Pendiente de leer						R	R	R	ER	ER	ER	ER	P		R											R	ER	ER		
Error en Emisión																														
Leida						R	R	ER	ER	ER	ER	P														R	ER	ER	ER	
Aceptada						R	ER	ER	ER	ER	P																		ER	
Rechazada						R	ER	ER	ER	ER	P																	ER	ER	ER
Cobro Externo																														
Emitido Cobro						R	R			ER	ER	ER	P																ER	
Pago Externo																														
Emitido Pago						R	R	ER	R	ER	ER	P																		
Cobrada						R	R	ER	ER	ER	P																		ER	
Pagada						R	R	ER	ER	ER	EP																		ER	
Rectificada								ER	ER	ER	ER																		ER	
Notificada						R	R	R	ER	ER	ER	ER	P		R														ER	
Descargada automáticamente						R	R	R	ER	ER	ER	ER	P																ER	
Enviada a plataforma externa																														
Rechazada por plataforma externa																														
Pro forma enviada																														
Pro forma leida																														
Pro forma aceptada																														
Pro forma enviada																														
Pro forma rechazada																														
En custodia																														
Registrada							R	R																				R	R	
En trámite de autorización							R	R																				R	R	
En trámite de pago							R	R	ER		ER	ER	R																R	
Gestión de incidencia							R	R																				R		

E La transición es forzada por el Emisor
R La transición es forzada por el Receptor
P La transición es forzada por la Plataforma

Para ello, se validará que el CIF asociado al ticket de sesión incluido en la llamada al servicio sea el del emisor/receptor de la factura antes de proceder al cambio de estado. En caso de solicitarse un cambio de estado que no cumpla estas condiciones, se devolverá el error *WS_ERROR_005*.



5.16 rejectInvoices

5.16.1 Descripción

Servicio que permite rechazar una factura, indicando la razón, el id, y su tipo de Identificador de ID, al final este método termina llamando a un `setInvoiceDataStatus`, por lo cual se ve sometido a la matriz de validación anterior.

5.16.2 Parámetros de entrada

- `java.lang.String ticket`: ticket de sesión válido.
- `java.lang.String reasonReject`: razón del rechazo de la factura.
- `java.lang.String typeId`: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver **anexo A**.
- `java.util.List ids`: Lista de ids de facturas, cuyo valor será asociado al `typeId`.
- `java.lang.String sellerCIF`: Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como `typeId` se ha especificado `ID_FISCAL`.

5.16.3 Parámetros de salida

Un objeto `InvoiceStatusDataBeanSet` con información sobre el resultado de la invocación al Web service así como un vector de objetos `InvoiceStatusDataBean` con información de las facturas consultadas. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.16.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo `error` del objeto devuelto por el servicio (se puede consultar con el método `getError` del mismo) será `false`. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_005	Cambio de estado no permitido
WS_ERROR_023	Intento de acceso ilegal desde un usuario con CIF: ..
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.

5.17 getInvoices

5.17.1 Descripción

Servicio que permite obtener un grupo de facturas dado sus IDs asociados como en otros métodos a un tipo de identificador. Posee además un parámetro de tipo booleano que permite indicar si se desea la relación de todas las facturas recibidas entre dichas fechas o únicamente aquéllas que no han sido descargadas desde el ERP, además si se informa un formato, se puede tener acceso a un archivo ZIP con las facturas solicitadas a través de los parámetros suministrados, de lo contrario se obtiene un array de objetos tipo *InvoiceDataBean*

5.17.2 Parámetros de entrada

- java.lang.String ticket: ticket de sesión válido.
- java.lang.String downloadFormat: Formato en que sean descargar las facturas indicadas. Se permiten los siguientes valores catalogados en archivo de constantes:

IDENTIFICADOR	LITERAL
CM10	CM10
COPIA	Copia del original firmada digitalmente
PDF	Un SOLO fichero PDF
UBL10	UBL 1.0
FACTURAE30	FACTURAE 3.0
FACTURAE31	FACTURAE 3.1
FACTURAE32	FACTURAE 3.2

Estos formatos provienen de la tabla formato.

- java.lang.String typeId: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver **anexo A**.
- java.util.List ids: Lista de ids de facturas, cuyo valor será asociado al typeId.
- java.lang.String sellerCIF: CIF del proveedor, necesario para identifica unívocamente una factura si como typeId se ha especificado ID_FISCAL.
- boolean onlyNew: Valor booleano que indica si se desean descargar todas las facturas recibidas entre initDate y endDate y que aún no han sido descargadas por el ERP (valor *true*) o bien si se desean descargar todas las facturas comprendidas entre dichas fechas independientemente de si han sido descargadas con anterioridad (valor *false*).

5.17.3 Parámetros de salida

Un objeto *InvoiceDataBeanSet* con información sobre el resultado de la invocación al Web service así como un vector de objetos *InvoiceDataBean* con información de las facturas recibidas, un objeto *DataHandler* que contendrá un archivo ZIP con las facturas recibidas, un boolean que indicara si existen mas facturas pendientes por



descargar. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.17.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* del objeto devuelto por el servicio (se puede consultar con el método *getError* del mismo) será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_024	Error al intentar descargar facturas en formato:..
WS_ERROR_027	Warning. No existen datos para los parámetros solicitados.
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.

5.18 sendInvoicesSign

5.18.1 Descripción

Este servicio permite la importación de un fichero XML con una factura en formato FACTURAE32, o bien un archivo zip con varias facturas en dichos formatos. Devuelve un objeto *SendFileBeanSet* que aporta información sobre las facturas importadas y el estado de la importación. Los estados en los que puede quedar la factura tras la importación están definidos de antemano de tal forma que a partir de esta información la aplicación pueda decidir como comportarse al respecto.

El sistema emitirá automáticamente las facturas incorporadas mediante este método, asignándoles número de serie y factura (salvo error de formato de las mismas o del proceso de emisión).

5.18.2 Parámetros de entrada

- `java.lang.String ticket`: ticket de sesión válido.
- `javax.activation.DataHandler ficheroEntrada` : `DataHandler` con el fichero de facturas a importar.
- `java.lang.String tipoFirma`:



- `java.lang.String tipoFactura`: Tipo de factura que se desea importar a eFactura. Podrá tomar los valores "380" para facturas, "389" para autofacturas, o bien "384" para rectificativas, valores todos ellos catalogados en archivo de constantes (constantes `WS_INVOICE_TYPE_FACTURA`, `WS_INVOICE_TYPE_AUTOFACTURA`, `WS_INVOICE_TYPE_FACTURA_RECTIFICATIVA`, `WS_INVOICE_TYPE_AUTOFACTURA_RECTIFICATIVA`). Para el caso de las facturas rectificativas, se indica siempre el tipo "384" siendo el sistema internamente quien detecta si se trata de una factura o de una autofactura a partir del código de la factura que se pretende rectificar. Los valores de este campo **no** se chequean (se coge el tipo de factura directamente de la factura), únicamente se tiene en cuenta si se pasa "1" y la factura no es factura "4" y no es autofactura. De todos modos, se recomienda mantener la especificación como está y estudiar el validar estos valores.
- `java.lang.String formato`: Formato en que sean incorporar al sistema las facturas. Únicamente se permiten los siguientes valores "FACTURAE32" valores catalogados en archivo de constantes.

IDENTIFICADOR	LITERAL
CM10	CM10
FACTURAE30	FACTURAE 3.0
FACTURAE31	FACTURAE 3.1
FACTURAE32	FACTURAE 3.2
PROPIETARIO	Formato propietario
UBL10	UBL 1.0

Estos formatos se encuentran configurados en la tabla formato.

5.18.3 Parámetros de salida

Devuelve un objeto `ResumenValidacionBeanSet` con información sobre el resultado de la invocación al Web service y un vector de objetos con la información particular del resultado de la importación de cada una de las facturas importadas. Consultar la Javadoc al respecto para más información sobre este objeto y sus miembros.

5.18.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo `error` del objeto devuelto por el servicio (se puede consultar con el método `getError` del mismo) será `false`. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_012	Ya existen facturas con las series o número de factura solicitadas
WS_ERROR_014	Debe configurar el separador entre la serie y el número de la factura
WS_ERROR_017	Error con el prefijo serie. Tiene que tener entre 1 y 4 caracteres alfanumericos
WS_ERROR_018	Error, el prefijo serie no está asociado a la delegación correspondiente
WS_ERROR_019	Error con el prefijo serie

6 Servicio Web eFacturaWSNet - <XML>

Este es el servicio que ofrece las funcionales del sistema para ERPs con una interfaz basada en XML.

A continuación, se muestra una descripción detallada de los métodos que ofrece el Web service eFacturaWSNet.

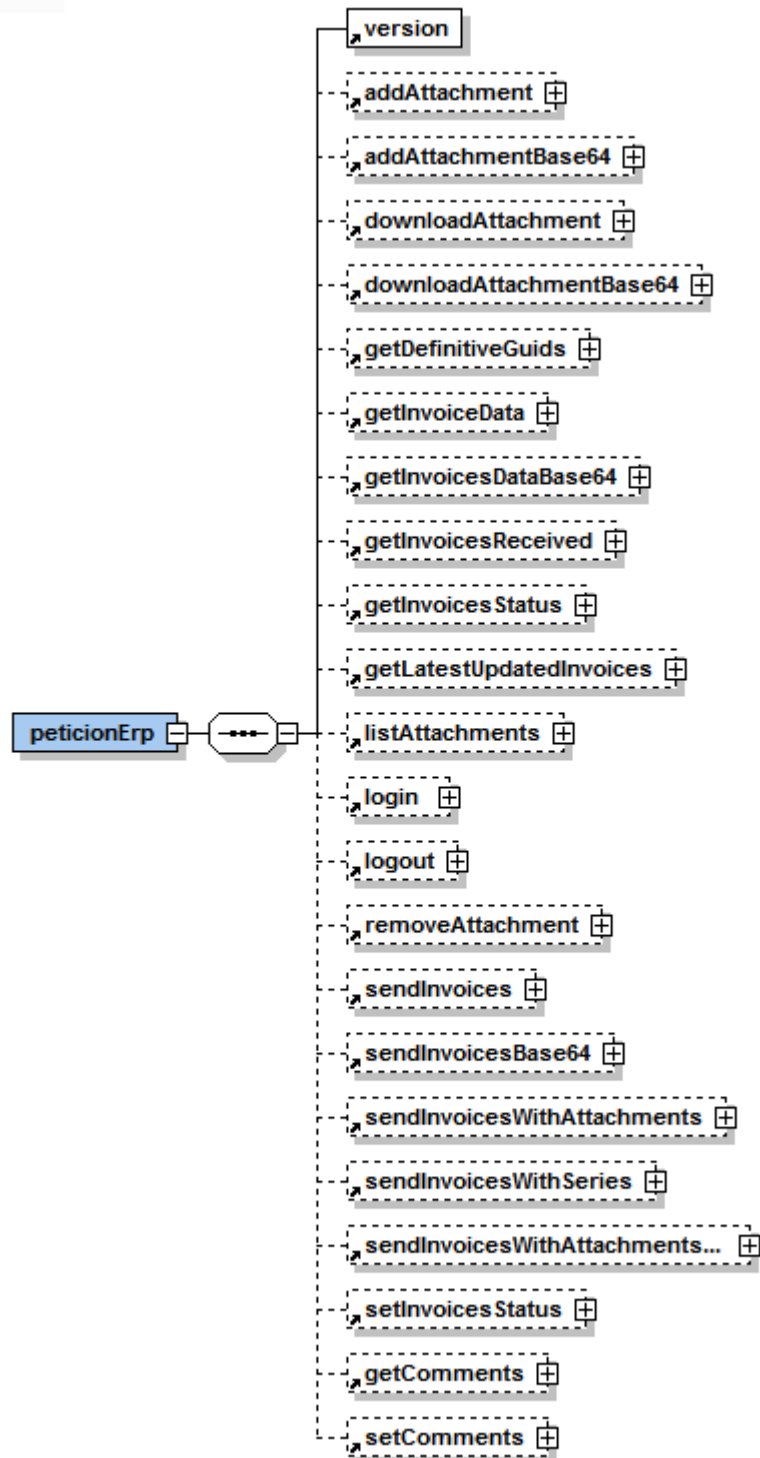
Para cada uno de los métodos se indica:

1. Sus parámetros de entrada (versión, clase y nombre del parámetro), la versión, que es obligatoria para todos los métodos salvo para logout, informa acerca de la versión del XML de petición, los parámetros del método descritos en el mismo orden que tienen en la declaración del método, con una descripción funcional de los mismos. En el caso de tratarse de Strings que contienen XML, se completa con la rama del XSD correspondiente. En el caso de que sus valores estén acotados, un listado de dichos valores.
2. Su parámetro de salida, indicando la clase a la que pertenecen y describiendo su funcionalidad. La información de estos objetos, en el caso de tratarse de Strings que contienen XML, se completa con la rama del XSD correspondiente.

Dado que los métodos de este servicio son idénticos a los del eFacturaWS, no se incluye en este apartado la descripción funcional de los mismos. Para consultar la misma, acudir a la definición del método equivalente del servicio eFacturaWS.

Estructura general de los XSDs:

1. Entrada: Tenemos dos XSDs diferentes de entrada.
 - a. Uno que contiene la versión y el nombre del método con los parámetros. Su estructura general es:





- b. Y otro que únicamente contiene el ticket.

ticket

2. Salida:

- a. Un XSD que contiene el nombre y los datos devueltos. Su estructura general es:



6.1 Login

6.1.1 Descripción

Este método identifica y autentica ante Sistema eFactura a la entidad adscrita que desea utilizar los servicios de la plataforma mediante el WS-API ERP.

Permite a una entidad adscrita obtener un ticket de sesión válido para ser utilizado durante un tiempo limitado en las siguientes llamadas. Para ello debe proporcionarse la llave que ha sido obtenida desde el módulo web, una vez haya sido descifrada con la contraseña introducida por el usuario en el módulo web en el momento de la solicitud de la llave.

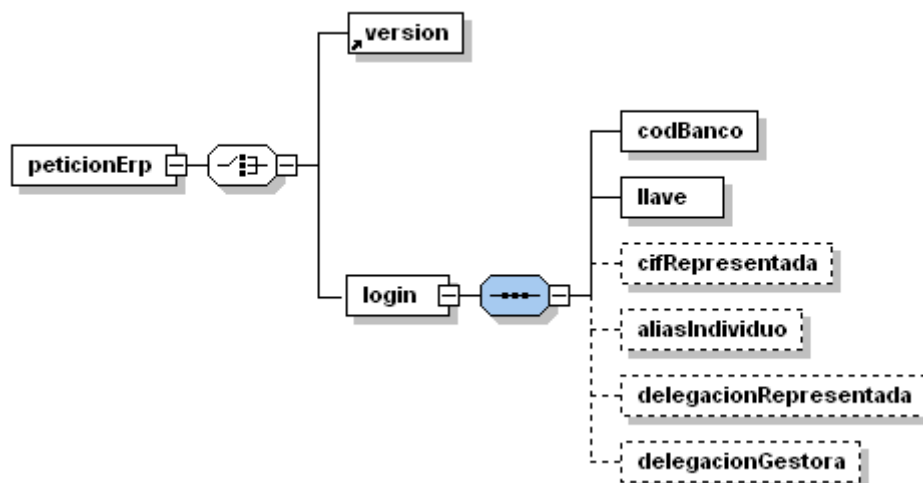
Además del ticket, se obtiene una nueva llave cifrada con la misma contraseña, esta llave es la que deberá utilizarse en el próximo login. La entidad promotora puede permitir a la entidad adscrita que la llave no cambie en cada petición de login, en este caso la llave devuelta no tiene porque ser preservada para el siguiente login pues es idéntica a la que se acaba de utilizar.

Para información más detallada sobre el mecanismo de autenticación y control de acceso del Web service eFacturaWSNet, consultar el apartado 4.1 Autenticación y Control de Acceso.

Si el usuario ya esta logado, por ejemplo mediante otra petición de login anterior, el nuevo login no inhabilita el ticket previo, por lo que la sesión anterior se puede seguir utilizando.

6.1.2 Parámetros de entrada

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- *version*: Indica la versión de la plataforma que cumple el xml de petición.

- o *codBanco*: Contiene una cadena con formato numérico que indica el Código de Banco.
- o *llave*: Una cadena de texto indicando los datos necesarios para establecer la sesión del usuario. No estará relacionada con el login y usuario de la entidad para prevenir posibles casos de phishing.
- o *cifRepresentada*: El cif de la empresa sobre cuyas facturas se va a trabajar. Es un campo opcional. De no venir informado, se entenderá que se quiere trabajar sobre la misma empresa que se descargó la llave.
- o *aliasIndividuo*: El alias del individuo que ejecuta las peticiones. Es un campo opcional.
- o *delegacionRepresentada*: La delegación de la empresa representada. Es un campo opcional.
- o *delegacionGestora*: La delegación de la empresa gestora. Es un campo opcional.

Ejemplo:

```

<peticionErp>
  <version>2.4</version>
  <login>
    <codBanco>Aquí un código de banco</codBanco>
    <llave>Aquí una llave</llave>
    <cifRepresentada>Aquí el cif de la empresa representada</cifRepresentada>
    <aliasIndividuo>Aquí el alias del individuo</aliasIndividuo>
    <delegacionRepresentada>Aquí la delegacion
representada</delegacionRepresentada>
    <delegacionGestora>Aquí la delegacion gestora</delegacionGestora>
  </login>
</peticionErp>

```

6.1.3 Parámetros de salida

java.lang.String xmlOut: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *ticket*: un ticket válido para la llave proporcionada.
- *llaveCifrada*: una nueva llave cifrada con la clave proporcionada por el usuario en el momento de la solicitud del acceso vía Web services.



Ejemplo:

```
<respuestaErp>
  <loginBean>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <ticket>Aquí un ticket</ticket>
    <llaveCifrada>Aquí una llave cifrada</llaveCifrada>
  </loginBean>
</respuestaErp>
```

6.1.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_006	El login proporcionado es erróneo
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido

6.2 Logout

6.2.1 Descripción

Mediante este método se desconecta del Sistema eFactura.

6.2.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:



- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

6.2.3 Parámetros de salida

`java.lang.String xmlOut`: una cadena con formato XML con la siguiente estructura:



- El árbol `bean`: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.

Ejemplo:

```
<respuestaErp>
  <logoutBean>
    <bean>
      <error>Aquí true o false<error>
      <subCode>Aquí código del error si lo hay<subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
  </logoutBean>
</respuestaErp>
```

6.2.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo `error` de la rama `bean` del xml devuelto será `false`. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada

WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_021	El XML de entrada no es válido

6.3 sendInvoicesBase64

6.3.1 Descripción

Este servicio permite la importación de un fichero XML con una factura en formato reportado en el método, o bien un archivo zip con varias facturas en dichos formatos. La factura o el ZIP de facturas deben estar codificados en Base 64. Devuelve información sobre la factura importada y el estado de la importación. Los estados en los que puede quedar la factura tras la importación están definidos de antemano de tal forma que a partir de esta información la aplicación pueda decidir como comportarse al respecto.

El sistema emitirá automáticamente la factura incorporada mediante este método, asignándole número de serie y factura o utilizando los que vienen en el fichero XML si así lo indican los parámetros de entrada (salvo error de formato de las mismas o del proceso de emisión).

6.3.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

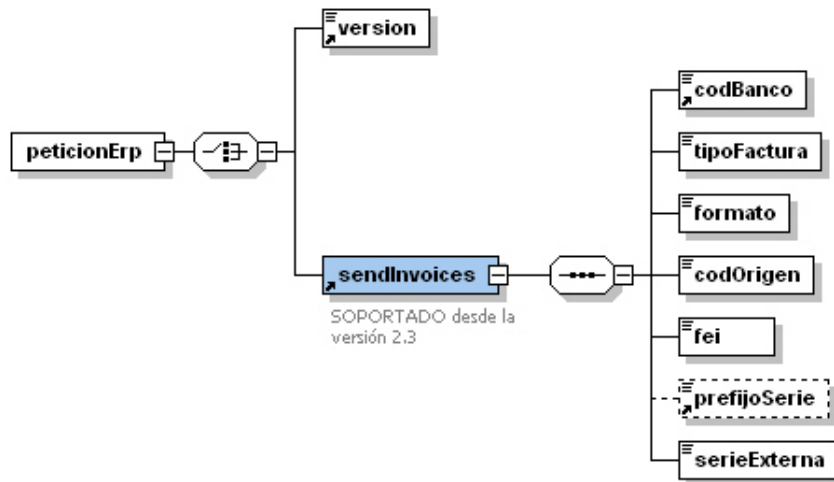


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- o *version*: Indica la versión de la plataforma que cumple el xml de petición.
- o *codBanco*: Contiene una cadena con formato numérico que indica el Código de Banco.
- o *tipoFactura*: Tipo de factura que se desea importar a eFactura. Valores posibles mirar el método sendFile de eFacturaWS.
- o *formato*: formato en que se incorporan al sistema las facturas. Valores posibles mirar el método sendFile de eFacturaWS.
- o *codOrigen*: Código origen que identifica el ERP. Valores posibles de 105 a 200, de lo contrario se devolverá el error WS_ERROR_007.
- o
- o *fei*: Indica si se trata de una factura emitida para imprimir. Puede tomar los valores *true* o *false*.
- o *prefijoSerie*: El prefijo de la serie a la que asociar la/s factura/s a emitir. Es un campo opcional. De no indicarse nada en el campo, se intenta emitir la/s factura/s asociándose a la serie por defecto, es decir, sin anteponer ningún prefijo.
- o *serieExterna*: Indica si hay que respetar la serie y el número que viene en la factura importada. Puede tomar los valores *true* o *false*.

Ejemplo:

```

<peticionErp>
  <version>2.4</version>
  <sendInvoices>
    <codBanco>Aquí un código de banco</codBanco>
    <tipoFactura>Aquí un tipo de factura</tipoFactura>
    <formato>Aquí un formato de factura</formato>
  
```

```

<codOrigen>Aquí un código origen</codOrigen>
<fei>>true o false</fei>
<prefijoSerie>Aquí el prefijo serie</prefijoSerie>
<serieExterna>>true o false</serieExterna>

```

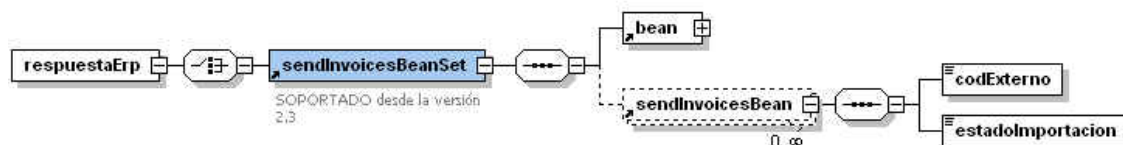
```
</sendInvoices>
```

```
</peticionErp>
```

- `java.lang.String ficheroEntrada`: String base 64 que representa el fichero de la factura a importar.

6.3.3 Parámetros de salida

`java.lang.String xmlOut`: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *sendInvoicesBean*: Hay uno para cada una de las facturas importadas, con el resultado de la importación.

6.3.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_001	Falta de comunicación con el servicio o el servicio no responde
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_012	Ya existen facturas con las series o número de factura solicitadas
WS_ERROR_014	Debe configurar el separador entre la serie y el número de la factura

WS_ERROR_017	Error con el prefijo serie. Tiene que tener entre 1 y 4 caracteres alfanumericos
WS_ERROR_018	Error, el prefijo serie no está asociado a la delegación correspondiente
WS_ERROR_019	Error con el prefijo serie
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido

Ejemplo:

```

<respuestaErp>
  <sendInvoicesBeanSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <sendInvoicesBean>
      <codExterno>Aquí un código externo</codExterno>
      <estadoImportacion>Aquí un estado importación</estadoImportacion>
    </sendInvoicesBean>
    .....
    .....
    <sendInvoicesBean>
      <codExterno>Aquí un código externo</codExterno>
      <estadoImportacion>Aquí un estado importación</estadoImportacion>
    </sendInvoicesBean>
  </sendInvoicesBeanSet>
</respuestaErp>

```

6.4 getInvoicesDataBase64

6.4.1 Descripción

Este servicio puede ser invocado para descargar facturas que han sido emitidas en el sistema. Permite obtener un archivo zip con las facturas solicitadas a través de su GUIDdefinitivo. El archivo ZIP recuperado se encuentra codificado en Base 64.

6.4.2 Parámetros de entrada

- java.lang.String xmlTicketIn: que contiene una cadena con formato XML con la siguiente estructura:



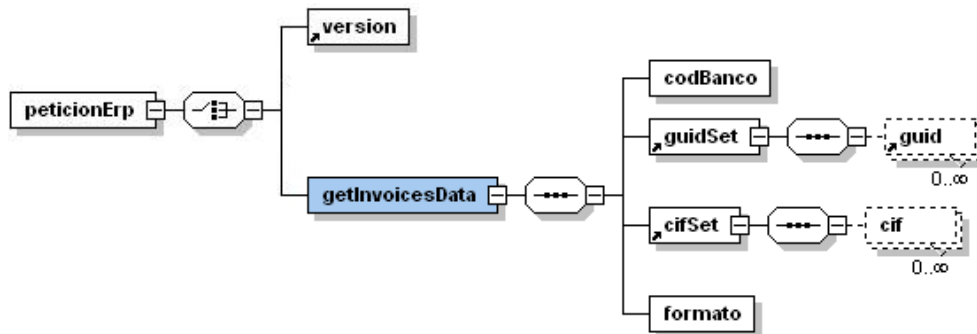
ticket

- o *ticket*: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- o *version*: Indica la versión de la plataforma que cumple el xml de petición.
- o *codBanco*: Contiene una cadena con formato numérico que indica el Código de Banco.
- o *guidSet*: Identificador universal de la facturas que se quieren descargar. En la rama hay que informar de nodos *guid*.
- o *cifSet*: CIF del proveedor de la facturas que se quieren descargar. Es necesario crear una entrada en este vector por cada uno de los *guid* indicados en el anterior parámetro. En la rama hay que informar de nodos *cif*.
- o *formato*: Formato en que sean descargar las facturas indicadas, ver el método correspondiente en eFacturaWS para ver los posibles valores.

Ejemplo:

```
<peticionErp>
  <versión>Aquí el número de versión</versión>
  <getInvoicesData>
    <codBanco>Aquí un código de banco</codBanco>
    <guidSet>
      <guid>Aquí un guid</guid>
      .....
      .....
      <guid>Aquí un guid</guid>
    </guidSet>
  </getInvoicesData>
</peticionErp>
```



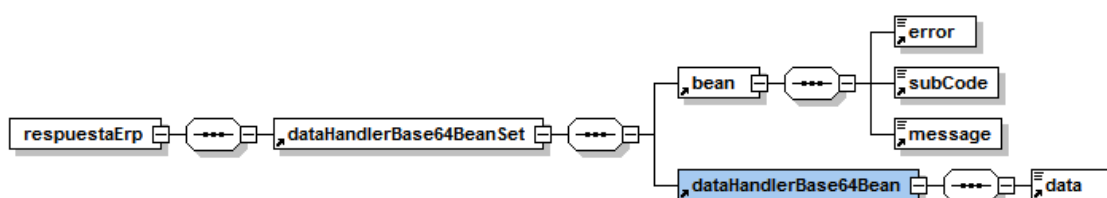
```

</guidSet>
<cifSet>
  <cif>Aquí un cif</cif>
  .....
  .....
  <cif>Aquí un cif</cif>
</cifSet>
<formato>Aquí un formato de factura</formato>
</getInvoicesData>
</peticionErp>

```

6.4.3 Parámetros de salida

java.lang.String xmlOut: una cadena con formato XML con la siguiente estructura:



- El árbol bean: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- dataHandlerBase64Bean: contendrá un campo *data* que contendrá un DataHandler con las facturas solicitadas convertido a Base64. Si se produjo un error el contenido estará vacío.

Ejemplo:

```

<respuestaErp>
  <dataHandlerBase64BeanSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <dataHandlerBase64Bean>
      <data>Aquí llega el DataHandler en Base 64</data>
    </dataHandlerBase64Bean>
  </dataHandlerBase64BeanSet>
</respuestaErp>

```

6.4.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido

6.5 addAttachmentBase64

6.5.1 Descripción

Servicio que permite adjuntar un fichero a una factura existente en el sistema. El fichero que se pretende adjuntar debe estar codificado en Base 64. Puede usarse para asociar a una factura ficheros que permitan aclarar su contenido o que por cualquier causa se desea que permanezcan asociados a la misma y accesibles desde su detalle. No hay límite en el número de ficheros que se pueden adjuntar a una factura, si bien hay un límite en tamaño total de adjuntos a una factura (en la actualidad, 512k).

6.5.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

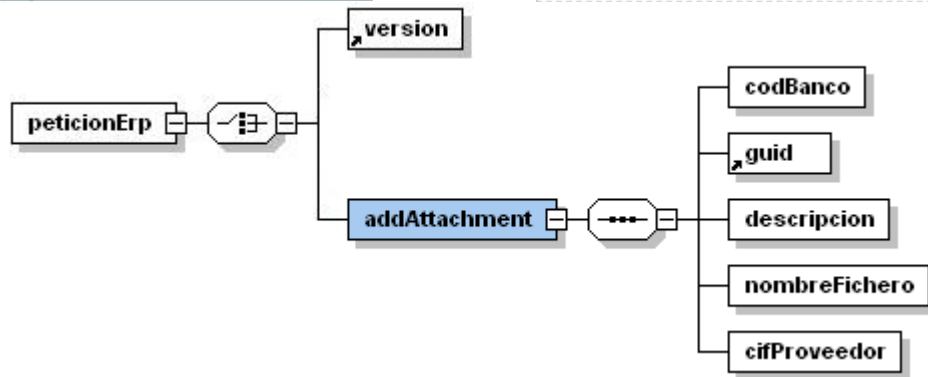


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- *version*: Indica la versión de la plataforma que cumple el xml de petición.
- *codBanco*: Contiene una cadena con formato numérico que indica el Código de Banco.
- *guid*: Identificador universal de la factura a la que se quiere adjuntar el archivo.
- *descripcion*: Descripción del adjunto.
- *nombreFichero*: Nombre del fichero que se adjunta.
- *cifProveedor*: CIF del proveedor de la facturas que se quieren consultar.
- `java.lang.String ficheroEntrada`: String en Base64 que representa el fichero que se desea adjuntar.

Ejemplo:

```

<peticionErp>
  <version>2.4</version>
  <addAttachment>
    <codBanco>Aquí un código de banco</codBanco>
    <guid>Aquí un guid </guid>
    <descripcion>Aquí una descripción</descripcion>
    <nombreFichero>Aquí el nombre del fichero</nombreFichero>
    <cifProveedor>Aquí el cif proveedor</cifProveedor>
  </addAttachment>
</peticionErp>
  
```

6.5.3 Parámetros de salida

`java.lang.String xmlOut`: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *addAttachmentBean*: con información sobre el resultado de la invocación al Web service y sobre el resultado de la inclusión del fichero adjunto.

Ejemplo:

```

<respuestaErp>
  <addAttachmentBean>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <errorOperativa>Un error operativa si ha habido</errorOperativa>
    <messageOperativa>un mensaje operativa si ha habido error</messageOperativa>
  </addAttachmentBean>
</respuestaErp>

```

6.5.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_010	No se ha podido obtener el identificador de la factura
WS_ERROR_020	Error, versión no soportada

6.6 listAttachments

6.6.1 Descripción

Método de consulta que devuelve una lista con la información de los ficheros adjuntos de una factura determinada.

6.6.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

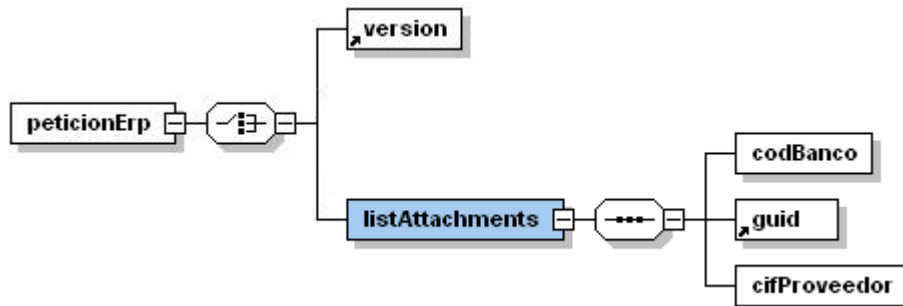
ticket

- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- `version`: Indica la versión de la plataforma que cumple el xml de petición.
- `codBanco`: Contiene una cadena con formato numérico que indica el Código de Banco.
- `guid`: Identificador universal de la factura a la que se quiere adjuntar el archivo.
- `cifProveedor`: CIF del proveedor de la facturas que se quieren consultar.

Ejemplo:

```

<peticionErp>
  <version>2.4</version>
  <listAttachments>
    <codBanco>Aquí un código de banco</codBanco>
    <guid>Aquí un guid</guid>
  
```



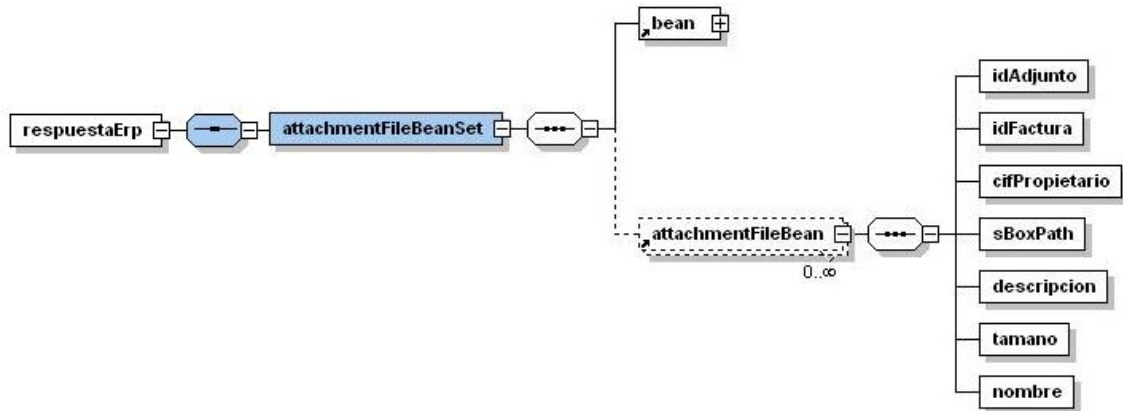
```

<cifProveedor>Aquí un cif proveedor</cifProveedor>
</listAttachemnts>
</peticionErp>

```

6.6.3 Parámetros de salida

java.lang.String *xmlOut*: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *attachmentFileBeanSet*: con información sobre los ficheros adjuntos de la factura consultada.

Ejemplo:

```

<respuestaErp>
  < attachmentFileBeanSet >
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <attachmentFileBean>
      <idAdjunto>Aquí un id de fichero adjunto</idAdjunto>
      <idFactura>Aquí un id factura</idFactura>
      <cifPropietario>Aquí un cif propietario</cifPropietario>
      <sBoxPath>Aquí la sBoxPath</sBoxPath>
      <descripcion>Aquí una descripcion</descripcion>
      <tamano>Aquí el tamaño</tamano>
      <nombre>Aquí un nombre</nombre>
    </attachmentFileBean>
    .....
    .....

```



```
<attachmentFileBean>
    <idAdjunto>Aquí un id de fichero adjunto</idAdjunto>
    <idFactura>Aquí un id factura</idFactura>
    <cifPropietario>Aquí un cif propietario</cifPropietario>
    <sBoxPath>Aquí la sBoxPath</sBoxPath>
    <descripcion>Aquí una descripcion</descripcion>
    <tamano>Aquí el tamaño</tamano>
    <nombre>Aquí un nombre</nombre>
</attachmentFileBean>
< attachmentFileBeanSet >
</respuestaErp>
```

6.6.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama bean del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_010	No se ha podido obtener el identificador de la factura
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido

6.7 downloadAttachmentBase64

6.7.1 Descripción

Método que permite descargar un fichero adjunto, indicando la factura a la que pertenece y el número de adjunto. El fichero adjunto se encuentra codificado en Base 64 como un nodo del XML de respuesta.

6.7.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

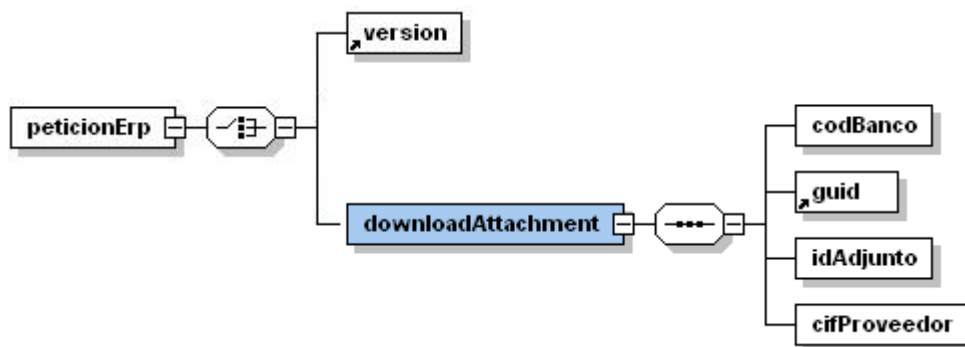


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



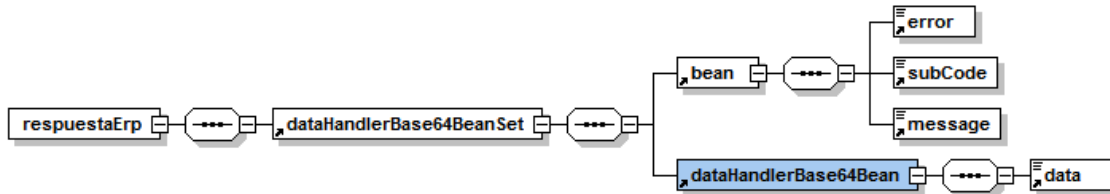
- `version`: Indica la versión de la plataforma que cumple el xml de petición.
- `codBanco`: Contiene una cadena con formato numérico que indica el Código de Banco.
- `guid`: Identificador universal de la factura de la que se quiere descargar el archivo.
- `idAdjunto`: Número de adjunto de dicha factura que se desea descargar.
- `cifProveedor`: CIF del proveedor de la facturas que se quieren consultar.

Ejemplo:

```
<peticionErp>
  <version>2.4</version>
  <downloadAttachment>
    <codBanco>Aquí un codigo de banco</codBanco>
    <guid>Aquí un guid definitivo</guid>
    <idAdjunto>Aquí un id de adjunto</idAdjunto>
    <cifProveedor>Aquí un cif proveedor</cifProveedor>
  </downloadAttachment>
</peticionErp>
```

6.7.3 Parámetros de salida

java.lang.String xmlOut: una cadena con formato XML con la siguiente estructura:



- El árbol bean: contiene información acerca de la terminación de la invocación detallada en el punto 4.3.2.
- dataHandlerBase64Bean: contendrá un campo *data* que contendrá un DataHandler con las facturas solicitadas convertido a Base64. Si se produjo un error el contenido estará vacío.

Ejemplo:

```
<respuestaErp>
  <dataHandlerBase64BeanSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <dataHandlerBase64Bean>
      <data>Aquí llega el DataHandler en Base 64</data>
    </dataHandlerBase64Bean>
  </dataHandlerBase64BeanSet>
</respuestaErp>
```

6.7.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID

	solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_010	No se ha podido obtener el identificador de la factura
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido

6.8 setComments

6.8.1 Descripción

Servicio que permite insertar una observación a una factura indicando el `guidDefinitivo`, el `cifProveedor` de esta, el tipo de observación y el texto de la observación.

6.8.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

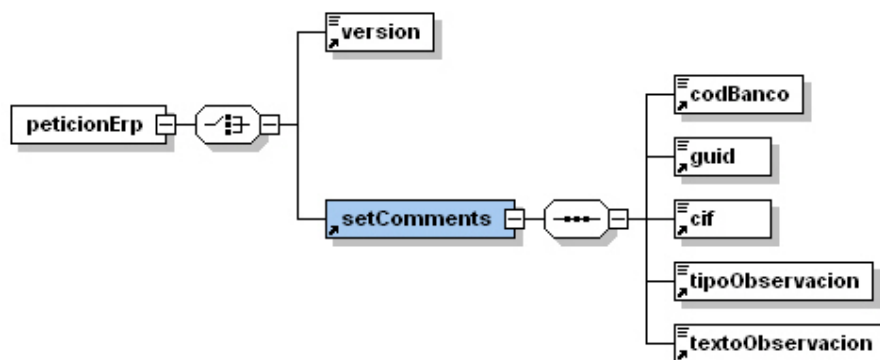


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- `version`: Indica la versión de la plataforma que cumple el xml de petición.

- o *codBanco*: Contiene una cadena con formato numérico que indica el Código de Banco.
- o *guid*: Identificador universal de la factura a la que se quiere establecer la observación.
- o *cif*: CIF del proveedor de la factura a la que se quiere establecer la observación.
- o *tipoObservacion*: Identificador del tipo de observación.
- o *textoObservacion*: Texto con la observación a insertar.

Ejemplo:

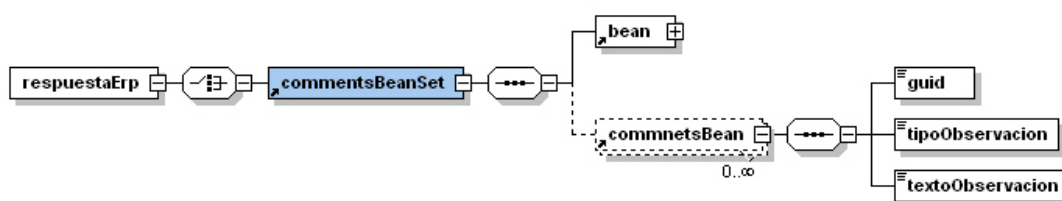
```

<peticionErp>
  <version>2.4</version>
  <setComments>
    <codBanco>Aquí un código de banco</codBanco>
    <guid>Aquí un guid</guid>
    <cif>Aquí un cif</cif>
    <tipoObservacion>Aquí un tipo de observación</tipoObservacion>
    <textoObservacion>Aquí un texto con la observación</textoObservacion>
  </setComments>
</peticionErp>

```

6.8.3 Parámetros de salida

java.lang.String *xmlOut*: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *commnetsBeanSet*: contiene información individual de cada observación.

Ejemplo:

```

<respuestaErp>
  <commentsBeanSet>
    <bean>

```



```

<error>Aquí true o false</error>
<subCode>Aquí código del error si lo hay</subCode>
<message>Aquí descripción del error si lo hay</message>
</bean>
  <commentsBean>
    <guid>Aquí un guid</guid>
    <tipoObservacion>Aquí un tipo de observación</tipoObservacion>
    <textoObservacion>Aquí un ltexto con la observación</textoObservacion>
  </commentsBean>
  .....
  .....
  <commentsBean>
    <guid>Aquí un guid</guid>
    <tipoObservacion>Aquí un tipo de observación</tipoObservacion>
    <textoObservacion>Aquí un ltexto con la observación</textoObservacion>
  </commentsBean>
</commentsBeanSet>
</respuestaErp>

```

6.8.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_004	Acceso no permitido para los GUID solicitados o GUID inexistentes
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_010	No se ha podido obtener el identificador de la factura
WS_ERROR_015	Tipo de observación no permitida
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido

6.9 setMetadata

6.9.1 Descripción

Esta operación permite asociar información adicional (metadato) a un documento, de forma que sea almacenada por la plataforma para su posterior consulta o modificación.

La información almacenada se compone de una pareja: clave, valor. Utilizándose la clave como identificador del tipo de información adicional, por ejemplo: "Fecha Registro", actuando valor como la información adicional, por ejemplo: "28-03-2010".

Si en el momento de asociar a un documento un metadato que ya tiene asociado, existe un metadato con la misma clave, es decir, dicho documento ya tiene información asociada a la misma clave, el valor será remplazado

Las limitaciones de la información almacenada son las siguientes:

- 20 byte (alfanuméricos) para la clave informada.
- 100 byte (alfanuméricos) para el valor informado.

En ambos casos se prohíben los caracteres raros: ? ¿ ! ¡ [] ... y no se recomienda el uso de tildes...

6.9.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

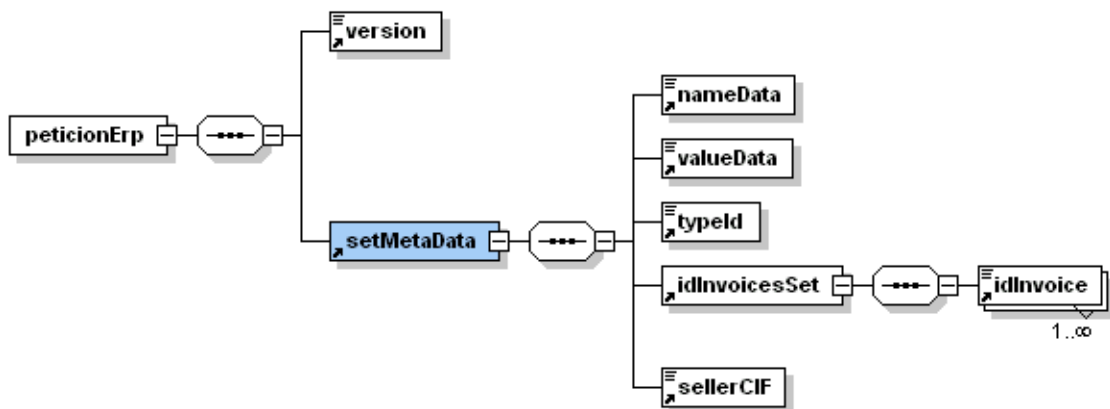


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- `version`: Indica la versión de la plataforma que cumple el xml de petición.

- o *nameData*: Nombre de la información adicional, que se utilizará como clave para recuperar el valor posteriormente.
- o *valueData*: Valor de la información adicional asociado a *nameData*.
- o *typeId*: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver anexo A.
- o *idInvoicesSet*: Lista de identificadores de documentos.
- o *idInvoice*: Identificador de documento.
- o *sellerCif*: Valor del CIF del proveedor, necesario para identificar unívocamente una factura si como *typeId* se ha especificado ID_FISCAL.

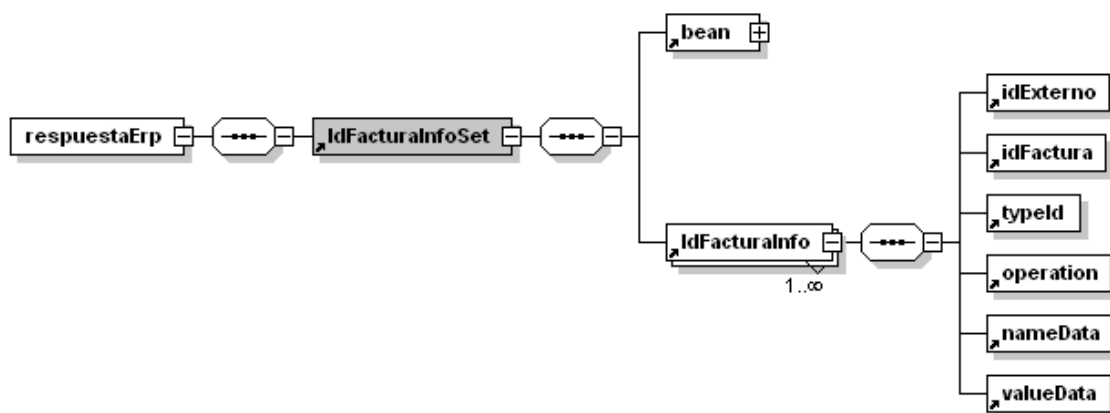
Ejemplo:

```

<peticionErp>
  <version>3.0</version>
  <setMetaData>
    <nameData>00000001</nameData>
    <valueData>valor del metdato</valueData>
    <typeId>ID_FISCAL</typeId>
    <idInvoicesSet>
      <idInvoice>001</idInvoice>
      <idInvoice>002</idInvoice>
      <idInvoice>003</idInvoice>
    </idInvoicesSet>
    <sellerCIF>11111111H</sellerCIF>
  </setMetaData>
</peticionErp>
  
```

6.9.3 Parámetros de salida

java.lang.String *xmlOut*: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.



- *IdFacturaInfoSet*:

Ejemplo:

```
<respuestaErp>
  <IdFacturaInfoSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <IdFacturaInfo>
      <idExterno>0001</idExterno>
      <idFactura>0001</idFactura>
      <typeId>ID_FISCAL</typeId>
      <operation>UPDATE</operation>
      <nameData>Nombre del metadato</nameData>
      <valueData>Valor del metadato</valueData>
    </IdFacturaInfo>
  </IdFacturaInfoSet>
</respuestaErp>
```

6.9.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif.
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.
WS_ERROR_029	No ha informado el CIF del proveedor.

6.10 getMetadata

6.10.1 Descripción

Esta operación permite obtener el valor de la información adicional (metadato) asociada a un documento.

Si el documento no contiene ninguna información adicional con la clave especificada la operación no devolverá nada.

6.10.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

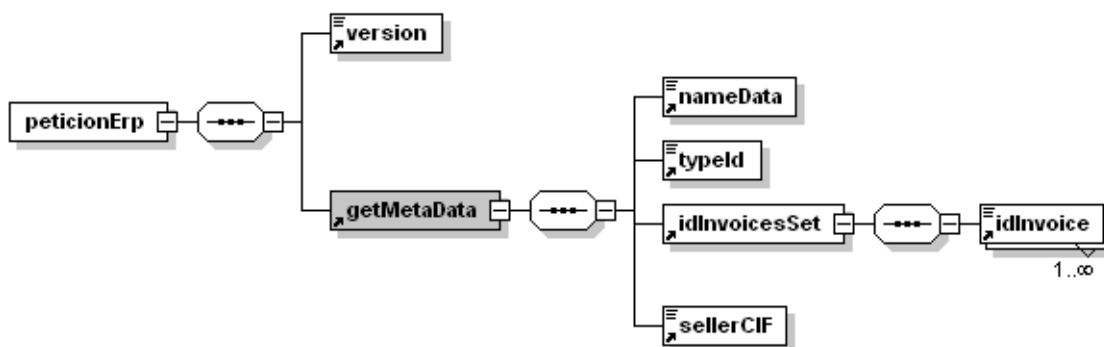
`ticket`

- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- `version`: Indica la versión de la plataforma que cumple el xml de petición.
- `nameData`: Nombre de la información adicional, que se utilizará como clave para recuperar el valor posteriormente.
- `typeId`: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver anexo A.
- `idInvoiceSet`: Lista de identificadores de documentos, cuyo valor será asociado al `typeId`.
- `idInvoice`: identificador de documento.
- `sellerCif`: Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como `typeId` se ha especificado ID_FISCAL.

Ejemplo:

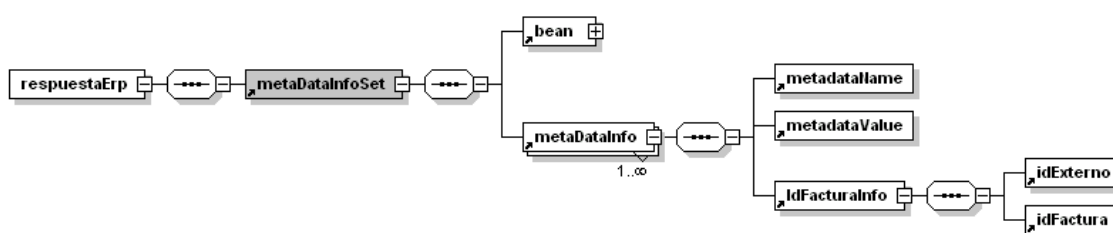
```

<peticionErp>
  <version>3.0</version>
  <getMetaData>
    <nameData>nombre metadato</nameData>
    <typeId>ID_FISCAL</typeId>
    <idInvoicesSet>
      <idInvoice>000001</idInvoice>
      <idInvoice>000002</idInvoice>
    </idInvoicesSet>
    <sellerCIF>11111111H</sellerCIF>
  </getMetaData>
</peticionErp>

```

6.10.3 Parámetros de salida

java.lang.String *xmlOut*: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *metaDataInfoSet*: contiene información individual de cada observación

Ejemplo:

```

<respuestaErp>
  <metaDataInfoSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <metaDataInfo>
      <metadataName>nombre metadato</metadataName>
      <metadataValue>valor del metadato</metadataValue>
      <IdFacturalInfo>
        <idExterno>000001</idExterno>
        <idFactura>30000</idFactura>
      </IdFacturalInfo>
    </metaDataInfo>
  </metaDataInfoSet>
</respuestaErp>

```



</metaDataInfo>

</metaDataInfoSet >

</respuestaErp>



6.10.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama bean del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif.
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.
WS_ERROR_029	No ha informado el CIF del proveedor.

6.11 rejectInvoices

6.11.1 Descripción

Servicio que permite rechazar una factura, indicando la razón, el id, y su tipo de Identificador de ID, al final este método termina llamando a un setInvoiceDataStatus, por lo cual se ve sometido a la matriz de validación anterior.

6.11.2 Parámetros de entrada

- java.lang.String *xmlTicketIn*: que contiene una cadena con formato XML con la siguiente estructura:

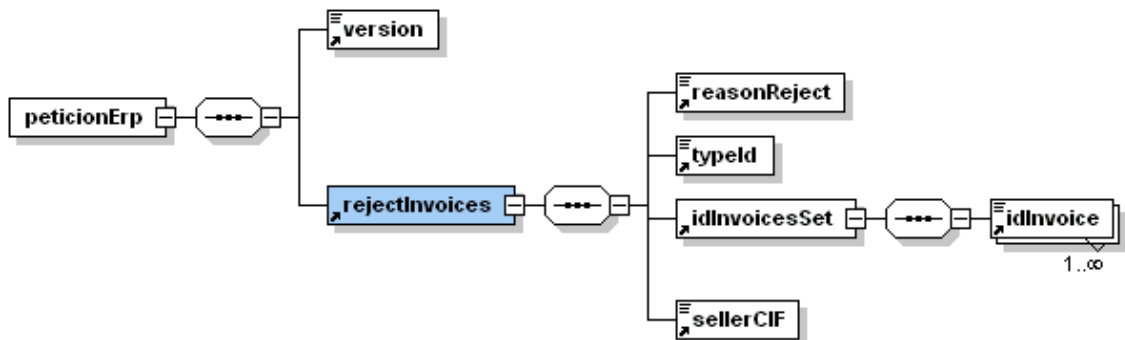


- *ticket*: ticket de sesión válido.

Ejemplo:

<ticket>Aquí un ticket</ticket>

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



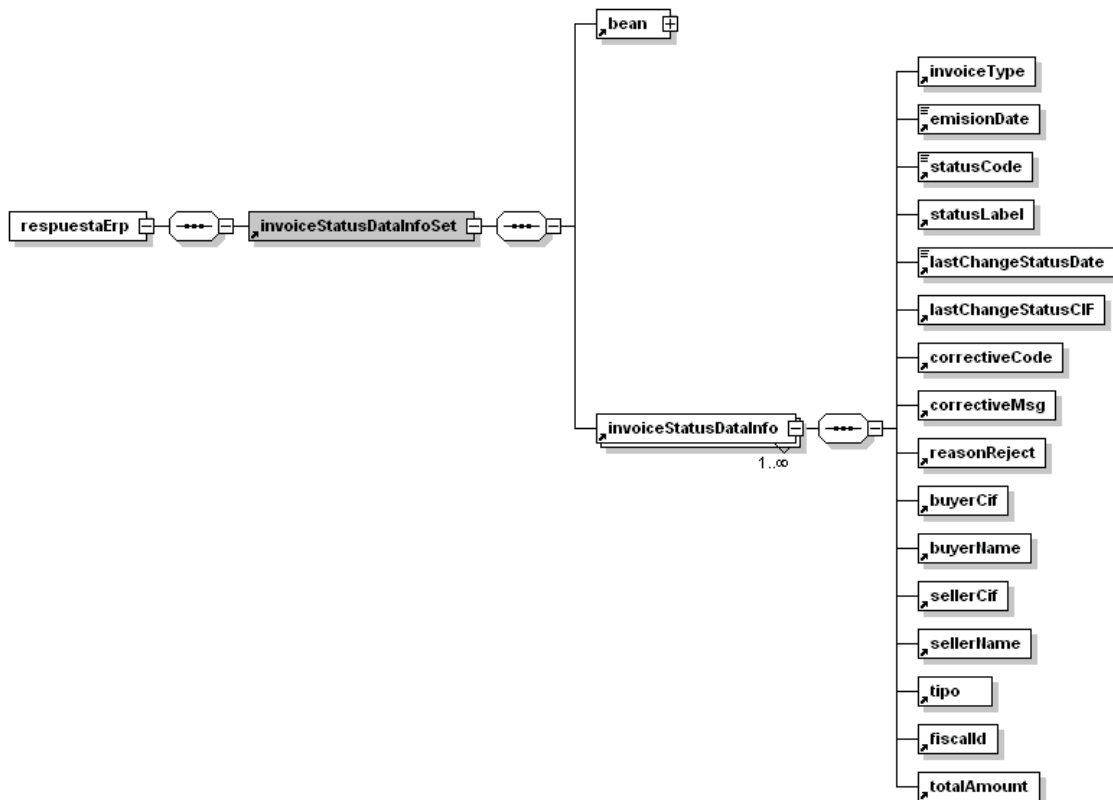
- *version*: Indica la versión de la plataforma que cumple el xml de petición.
- *reasonReject*: razón del rechazo de la factura.
- *typeId*: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver **anexo A**.
- *idInvoicesSet*: Lista de ids de facturas, cuyo valor será asociado al typeId.
- *idInvoice*: Identificador de la factura.
- *sellerCif*: Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como typeId se ha especificado ID_FISCAL.

Ejemplo:

```
<peticionErp>
  <version>3.0</version>
  <rejectInvoices>
    <reasonReject>Motivo de rechazo de la factura</reasonReject>
    <typeId>ID_FISCAL</typeId>
    <idInvoicesSet>
      <idInvoice>000001</idInvoice>
      <idInvoice>000002</idInvoice>
    </idInvoicesSet>
    <sellerCIF>11111111H</sellerCIF>
  </rejectInvoices>
</peticionErp>
```

6.11.3 Parámetros de salida

`java.lang.String xmlOut`: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *invoiceStatusDataInfoSet*:

Ejemplo:

```

<respuestaErp>
  <invoiceStatusDataInfoSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <invoiceStatusDataInfo>
      <invoiceType></invoiceType>
      <emisionDate>13/03/2011</emisionDate>
      <statusCode>0001</statusCode>
      <statusLabel>Aceptada</statusLabel>
      <lastChangeStatusDate>14/03/2011</lastChangeStatusDate>
      <lastChangeStatusCIF>11845560P</lastChangeStatusCIF>
      <correctiveCode></correctiveCode>
      <correctiveMsg></correctiveMsg>
      <reasonReject></reasonReject>
    </invoiceStatusDataInfo>
  </invoiceStatusDataInfoSet>
</respuestaErp>

```



```

<buyerCif>11111111H</buyerCif>
<buyerName>Nombre del proveedor</buyerName>
<sellerCif>11845560P</sellerCif>
<sellerName>Nombre del pagador</sellerName>
<tipo>1</tipo>
<fiscalId>0001</fiscalId>
<totalAmount>380</totalAmount>
</invoiceStatusDataInfo>
</invoiceStatusDataInfoSet>
</respuestaErp>

```

6.11.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama bean del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_005	Cambio de estado no permitido
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif.
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.
WS_ERROR_029	No ha informado el CIF del proveedor.
WS_ERROR_030	Error. Cambio de estado no permitido por las plataformas donde ha sido importada la factura.

6.12 getInvoicesDataStatus

6.12.1 Descripción

Servicio que permite obtener el estado de un grupo de facturas.

6.12.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

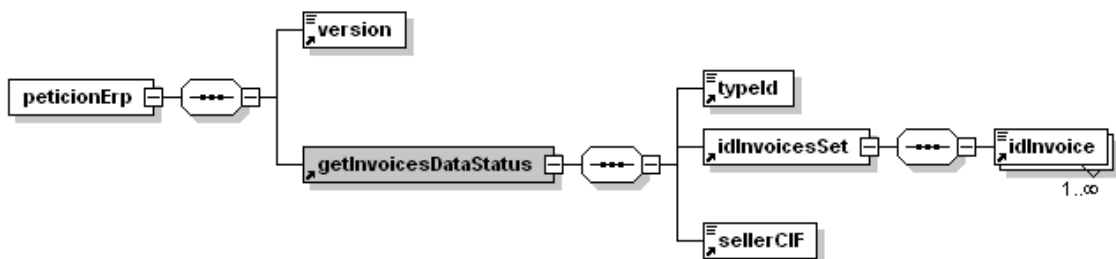


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



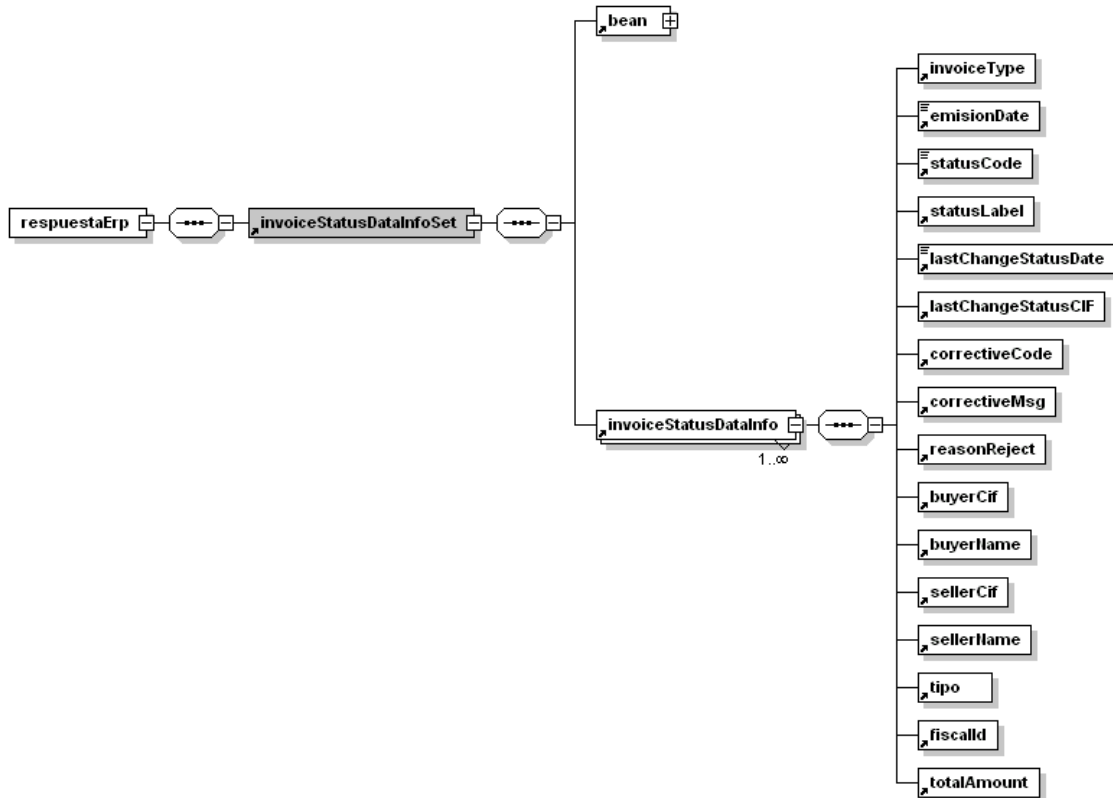
- `version`: Indica la versión de la plataforma que cumple el xml de petición.
- `typeld`: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver **anexo A**.
- `idInvoicesSet`: Lista de ids de facturas, cuyo valor será asociado al `typeld`.
- `idInvoice`: identificador de factura.
- `sellerCif`: Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como `typeld` se ha especificado `ID_FISCAL`.

Ejemplo:

```
<peticionErp>
  <version>3.0</version>
  <getInvoicesDataStatus>
    <typeld>ID_FISCAL</typeld>
    <idInvoicesSet>
      <idInvoice>000001</idInvoice>
      <idInvoice>000002</idInvoice>
    </idInvoicesSet>
    <sellerCIF>11111111H</sellerCIF>
  </getInvoicesDataStatus>
</peticionErp>
```

6.12.3 Parámetros de salida

java.lang.String *xmlOut*: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *invoiceStatusDataInfoSet*:

Ejemplo:

```

<respuestaErp>
  <invoiceStatusDataInfoSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <invoiceStatusDataInfo>
      <invoiceType></invoiceType>
      <emisionDate>13/03/2011</emisionDate>
      <statusCode>3</statusCode>
      <statusLabel>Pendiente de leer</statusLabel>
      <lastChangeStatusDate>14/03/2011</lastChangeStatusDate>
    </invoiceStatusDataInfo>
  </invoiceStatusDataInfoSet>
</respuestaErp>

```




```
<lastChangeStatusCIF></lastChangeStatusCIF>
<correctiveCode></correctiveCode>
<correctiveMsg></correctiveMsg>
<reasonReject></reasonReject>
<buyerCif>11111111H</buyerCif>
<buyerName>Nombre del proveedor</buyerName>
<sellerCif>11845560P</sellerCif>
<sellerName>Nombre del pagador</sellerName>
<tipo>1</tipo>
<fiscalId>0001</fiscalId>
<totalAmount>320</totalAmount>
</invoiceStatusDataInfo>
</idFacturaInfoSet>
</respuestaErp>
```

6.12.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif.
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.
WS_ERROR_029	No ha informado el CIF del proveedor.

6.13 setInvoicesDataStatus

6.13.1 Descripción

Servicio que permite cambiar el estado de una factura o varias facturas, indicando el id, y su tipo de Identificador de ID.

6.13.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

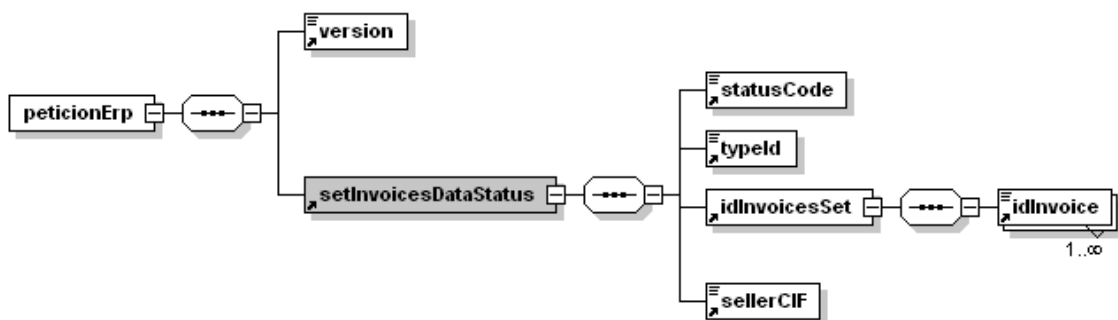


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



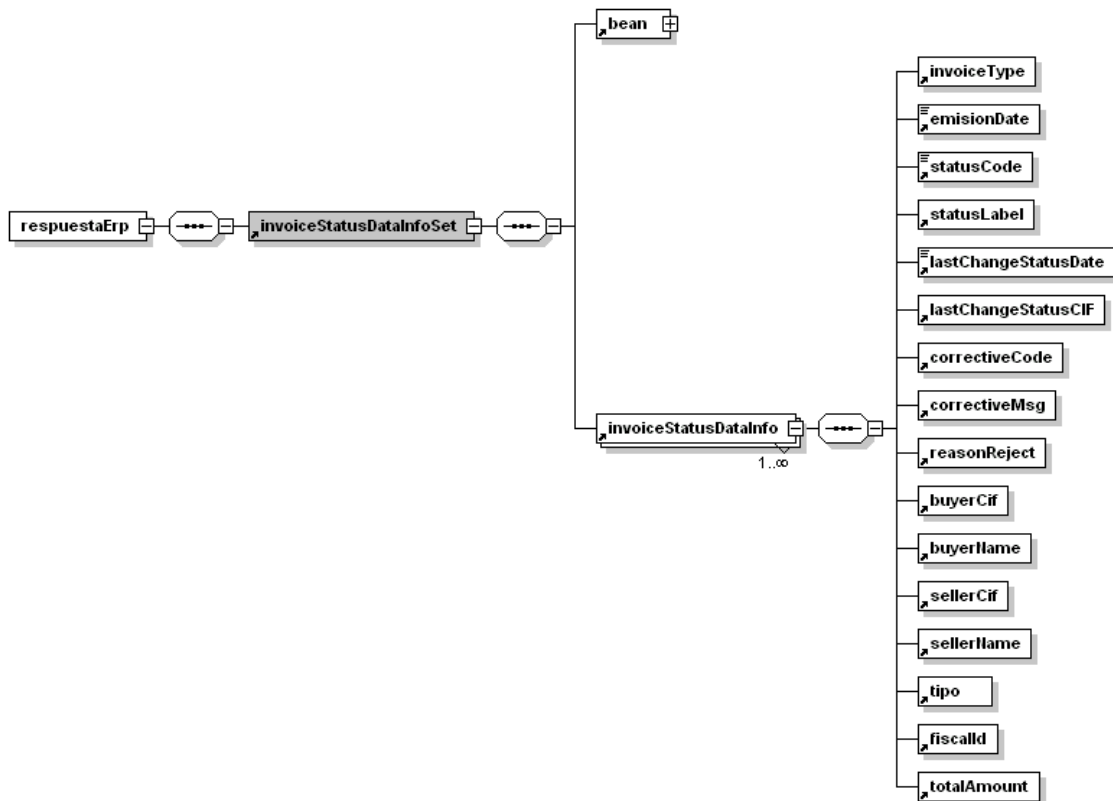
- `version`: Indica la versión de la plataforma que cumple el xml de petición.
- `statusCode`: Estado de la factura.
- `typeId`: Categorización del tipo de identificador usado para ubicar la o las facturas, posibles valores ver **anexo A**.
- `idInvoicesSet`: de ids de facturas, cuyo valor será asociado al `typeId`.
- `idInvoice`: Identificador de la factura.
- `sellerCif`: Valor del CIF del proveedor, necesario para identifica unívocamente una factura si como `typeId` se ha especificado `ID_FISCAL`.

Ejemplo:

```
<peticionErp>
  <version>3.0</version>
  <setInvoicesDataStatus>
    <statusCode>CODE</statusCode>
    <typeId>ID_FISCAL</typeId>
    <idInvoicesSet>
      <idInvoice>000001</idInvoice>
      <idInvoice>000002</idInvoice>
    </idInvoicesSet>
    <sellerCIF>11111111H</sellerCIF>
  </setInvoicesDataStatus>
</peticionErp>
```

6.13.3 Parámetros de salida

java.lang.String *xmlOut*: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *invoiceStatusDataInfoSet*:

Ejemplo:

```
<respuestaErp>
  <invoiceStatusDataInfoSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <invoiceStatusDataInfo>
      <invoiceType></invoiceType>
      <emisionDate>13/03/2011</emisionDate>
      <statusCode>5</statusCode>
```



```

<statusLabel>Aceptada</statusLabel>
<lastChangeStatusDate>14/03/2011</lastChangeStatusDate>
<lastChangeStatusCIF>11845560P</lastChangeStatusCIF>
<correctiveCode></correctiveCode>
<correctiveMsg></correctiveMsg>
<reasonReject></reasonReject>
<buyerCif>11111111H</buyerCif>
<buyerName>Nombre del proveedor</buyerName>
<sellerCif>11845560P</sellerCif>
<sellerName>11845560P</sellerName>
<tipo>4</tipo>
<fiscalId>0001</fiscalId>
<totalAmount>350</totalAmount>
</invoiceStatusDataInfo>
</idFacturalInfoSet>
</respuestaErp>

```

6.13.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_005	Cambio de estado no permitido
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif.
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.
WS_ERROR_029	No ha informado el CIF del proveedor.
WS_ERROR_030	Error. Cambio de estado no permitido por las plataformas donde ha sido importada la factura.

6.14 getInvoicesBase64

6.14.1 Descripción

Servicio que permite obtener un grupo de facturas dado sus IDs asociados como en otros métodos a un tipo de identificador. Posee además un parámetro de tipo booleano que permite indicar si se desea la relación de todas las facturas dados sus IDs o únicamente aquéllas que no han sido descargadas desde el ERP, además si se informa un formato, se puede tener acceso a un String en Base64 de un archivo ZIP con las facturas solicitadas a través de los parámetros suministrados, de lo contrario se obtiene un conjunto de etiquetas del tipo "InvoiceDataInfo".

6.14.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

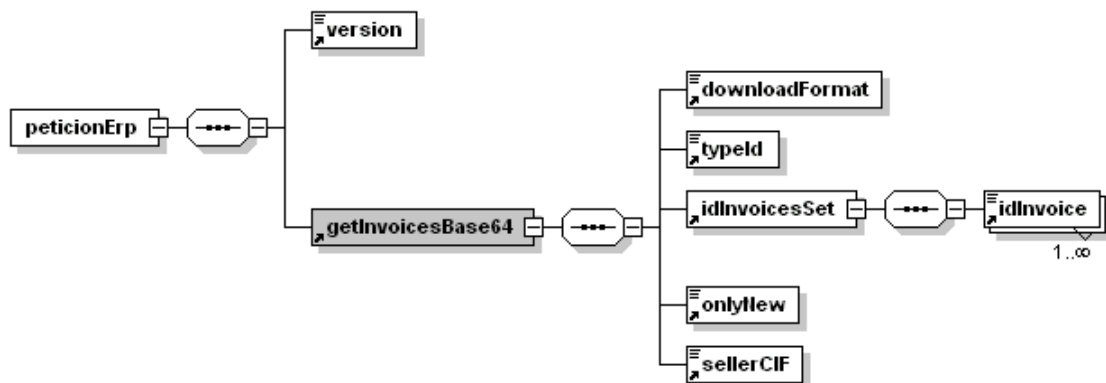


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- `version`: Indica la versión de la plataforma que cumple el xml de petición.
- `downloadFormat`: `downloadFormat`: Formato en que sean descargar las facturas indicadas. Se permiten los siguientes valores catalogados en archivo de constantes:

IDENTIFICADOR	LITERAL
CM10	CM10
COPIA	Copia del original firmada digitalmente
PDF	Un SOLO fichero PDF
UBL10	UBL 1.0
FACTURAE30	FACTURAE 3.0
FACTURAE31	FACTURAE 3.1

Estos formatos provienen de la tabla formato.

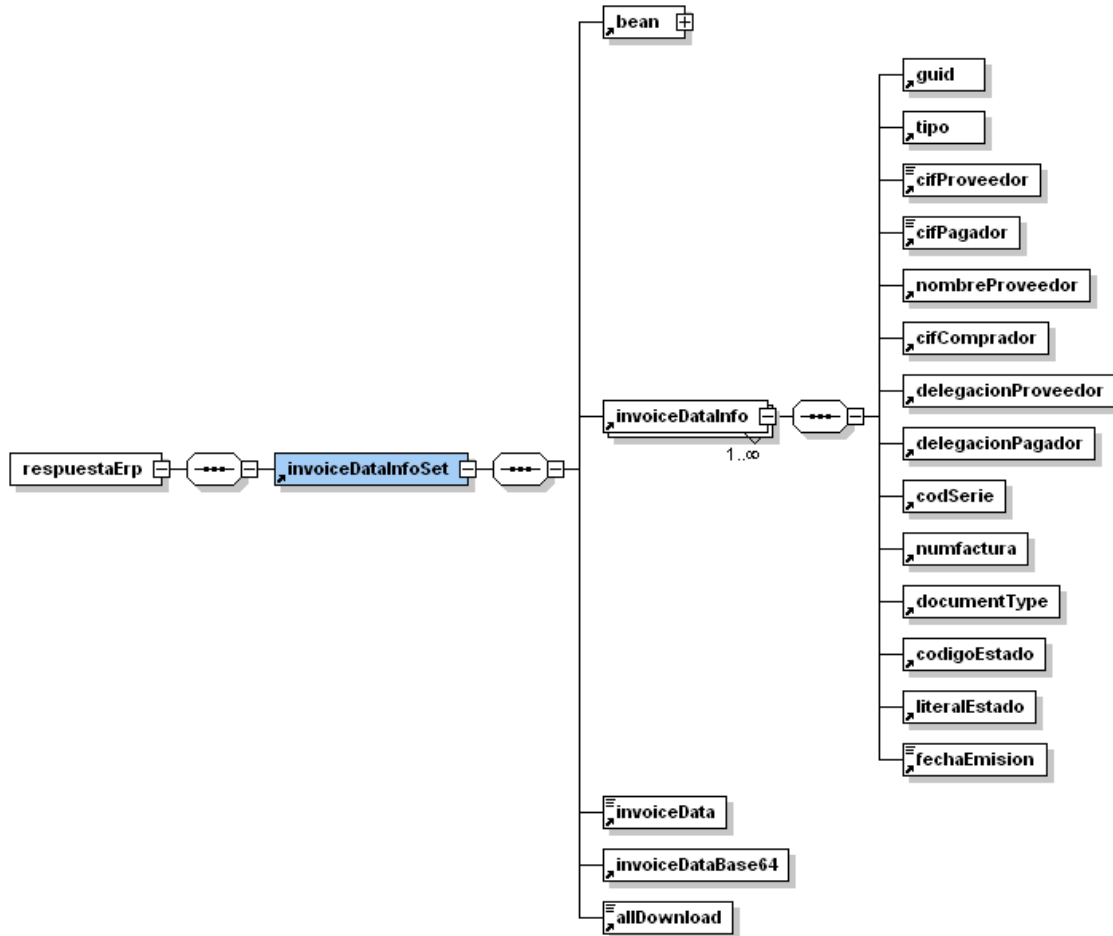
- o *typeld*: Identificador universal de la factura a la que se quiere establecer la observación.
- o *idInvoicesSet*: conjunto de ids de facturas, cuyo valor será asociado al *typeld*.
- o *idInvoice*: identificador de la factura.
- o *onlyNew*: Valor booleano que indica si se desean descargar todas las facturas recibidas entre *initDate* y *endDate* y que aún no han sido descargadas por el ERP (valor *true*) o bien si se desean descargar todas las facturas comprendidas entre dichas fechas independientemente de si han sido descargadas con anterioridad (valor *false*).
- o *sellerCif*: CIF del proveedor, necesario para identifica unívocamente una factura si como *typeld* se ha especificado *ID_FISCAL*.

Ejemplo:

```
<peticionErp>
  <version>3.0</version>
  <getInvoicesBase64>
    <downloadFormat>UBL10</downloadFormat>
    <typeld>ID_FISCAL</ typeld >
    <idInvoicesSet>
      <idInvoice>0001</idInvoice>
      <idInvoice>0002</idInvoice>
    </ idInvoicesSet>
    < onlyNew >true</ onlyNew >
    <sellerCIF>11111111H</ sellerCIF >
  </ getInvoicesBase64 >
</peticionErp>
```

6.14.3 Parámetros de salida

`java.lang.String xmlOut`: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *invoiceDataInfoSet*:

Ejemplo:

```

<respuestaErp>
  <invoiceDataInfoSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
  <invoiceDataInfo>
    <guid>0001</guid>
    <tipo>1</tipo>
    <cifProveedor>11111111H</cifProveedor>
    <cifPagador>22222222H</cifPagador>
    <nombreProveedor>Proveedor</nombreProveedor>
    <cifComprador>Comprador</cifComprador>
  </invoiceDataInfo>
</invoiceDataInfoSet>
</respuestaErp>

```



```
<delegacionProveedor>delegacion proveedor</delegacionProveedor>
<delegacionPagador>delegacion pagador</delegacionPagador>
<codSerie></codSerie>
<numfactura>0001</numfactura>
<documentType>380</documentType>
<codigoEstado>3</codigoEstado>
<literalEstado>Pendiente de leer</literalEstado>
<fechaEmision>13/03/2011</fechaEmision>
</invoiceDataInfo>
<invoiceData>Componente DataHandler</invoiceData>
<invoiceDataBase64></invoiceDataBase64>
<allDownload>>true </allDownload>
</invoiceDataInfoSet>
</respuestaErp>
```

6.14.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido
WS_ERROR_023	Intento de acceso ilegal desde un usuario con cif.
WS_ERROR_028	Error. Tipo de identificador solicitado inexistente.
WS_ERROR_029	No ha informado el CIF del proveedor.

6.15 getSentInvoicesBase64

6.15.1 Descripción

Este servicio devuelve la relación de facturas recibidas entre dos fechas indicadas y que están en un estado determinado. Las facturas se devuelven en el formato especificado codificadas en formato Base64 en un fichero ZIP contenidas en el tag "invoiceDataBase64", en caso de no especificar ningún formato se devuelve una colección de beans que resumen el contenido de la factura.

Todas las facturas devueltas quedan marcadas en el servidor como descargadas automáticamente por ERP. El parámetro de tipo booleano `onlyNew` permite filtrar de la relación de facturas recibidas que cumplen el filtro o aquellas que no han sido descargadas anteriormente desde ERP.

6.15.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

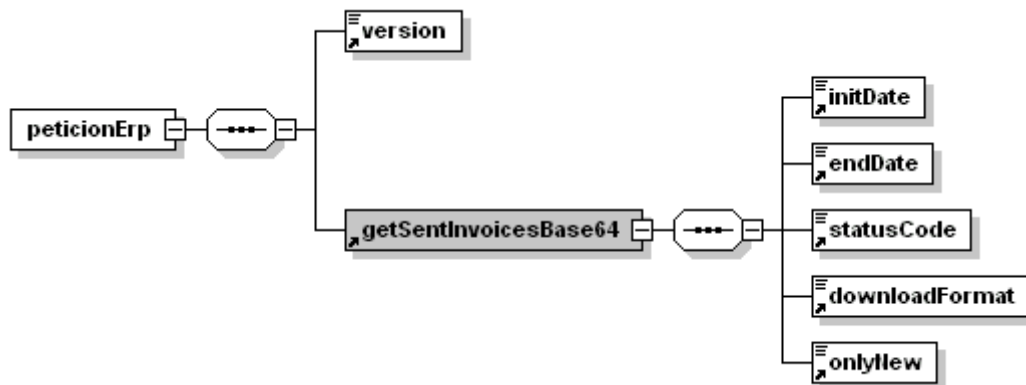
`ticket`

- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- `version`: Indica la versión de la plataforma que cumple el xml de petición.
- `initDate`: Fecha inicial de intervalo de búsqueda de facturas enviadas.
- `endDate`: Fecha final de intervalo de búsqueda de facturas enviadas.
- `statusCode`: Estado en el que deben estar las facturas enviadas.
- `downloadPending`: Formato en que sean descargar las facturas indicadas.
Se permiten los siguientes valores catalogados en archivo de constantes.



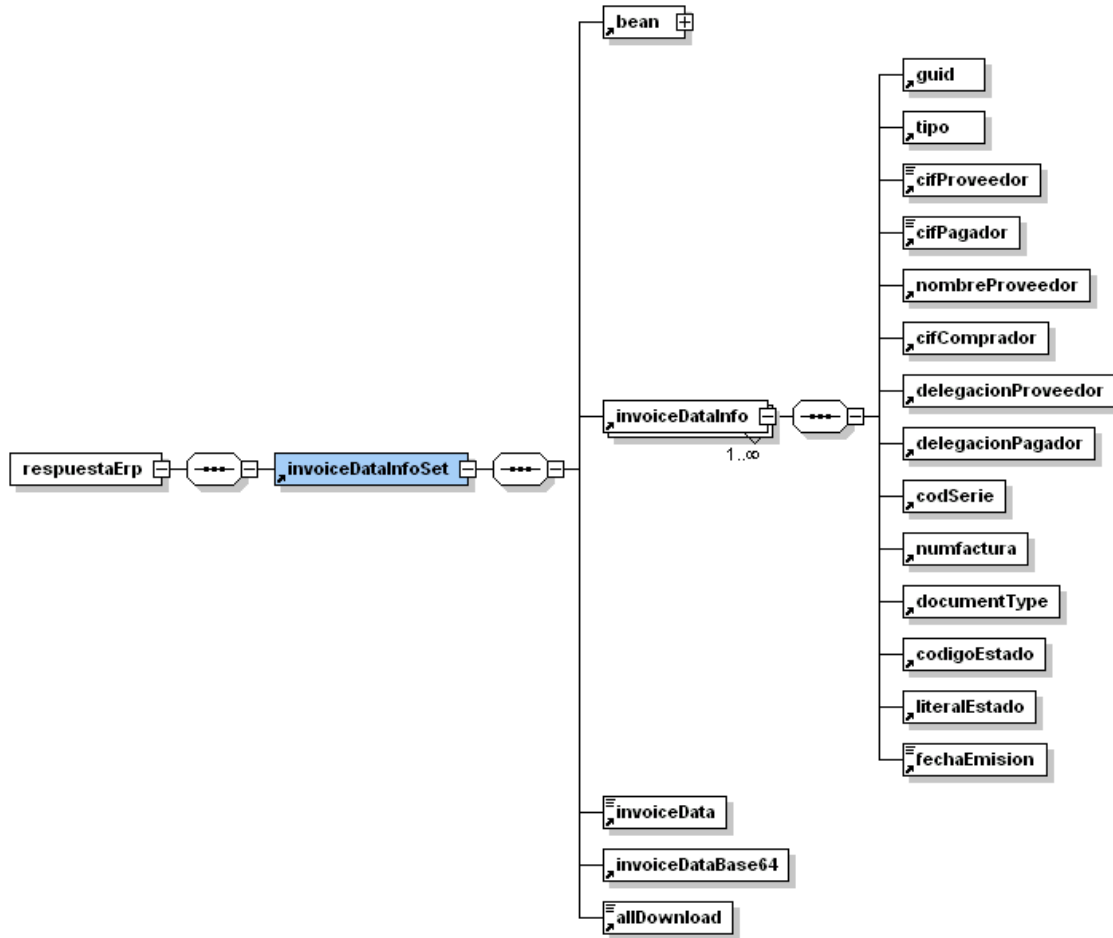
- o *onlyNew*: Indica si se desean descargar todas las facturas enviadas entre *initDate* y *endDate* y que aún no han sido descargadas por el ERP (valor *true*) o bien si se desean descargar todas las facturas comprendidas entre dichas fechas independientemente de si han sido descargadas con anterioridad (valor *false*).

Ejemplo:

```
<peticionErp>
  <version>3.0</version>
  <getSentInvoicesBase64>
    <initDate>13/03/2011</initDate>
    <endDate>14/03/2011</endDate>
    <statusCode>3</statusCode>
    <downloadPending>>false</downloadPending>
    <onlyNew>>true</onlyNew>
  </getSentInvoicesBase64>
</peticionErp>
```

6.15.3 Parámetros de salida

`java.lang.String xmlOut`: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información a cerca de la terminación de la invocación detallada en el punto 4.3.2.
- *invoiceDataInfoSet*: contiene información individual de cada observación

Ejemplo:

```

<respuestaErp>
  <invoiceDataInfoSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
    <invoiceDataInfo>
      <guid>0001</guid>
      <tipo>1</tipo>
      <cifProveedor>11111111H</cifProveedor>
      <cifPagador>22222222H</cifPagador>
    </invoiceDataInfo>
  </invoiceDataInfoSet>
</respuestaErp>

```



```

<nombreProveedor>Nombre del proveedor</nombreProveedor>
<cifComprador>22222222H </cifComprador>
<delegacionProveedor>00_COMMON</delegacionProveedor>
<delegacionPagador>00_COMMON</delegacionPagador>
<codSerie></codSerie>
<numfactura>0001</numfactura>
<documentType>1</documentType>
<codigoEstado>5</codigoEstado>
<literalEstado>Aceptada</literalEstado>
<fechaEmision>13/03/2011</fechaEmision>
</invoiceDataInfo>
<invoiceData>Componente DataHandler</invoiceData>
<invoiceDataBase64></invoiceDataBase64>
<allDownload>>true or false</allDownload>
</invoiceDataInfoSet>
</respuestaErp>

```

6.15.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido
WS_ERROR_024	Error al intentar descargar facturas en formato

6.16 getReceivedInvoicesBase64

6.16.1 Descripción

Este servicio devuelve la relación de facturas recibidas entre dos fechas indicadas y que están en un estado determinado. Las facturas se devuelven en el formato especificado codificadas en base 64 en un fichero ZIP., en caso de no especificar

ningún formato se devuelve una colección de tags "InvoiceDataInfo" en el XML resultado que resumen el contenido de la factura.

Todas las facturas devueltas quedan marcadas en el servidor como descargadas automáticamente por ERP. El parámetro de tipo booleano `onlyNew` permite filtrar de la relación de facturas recibidas que cumplen el filtro o aquéllas que no han sido descargadas anteriormente desde ERP.

6.16.2 Parámetros de entrada

- `java.lang.String xmlTicketIn`: que contiene una cadena con formato XML con la siguiente estructura:

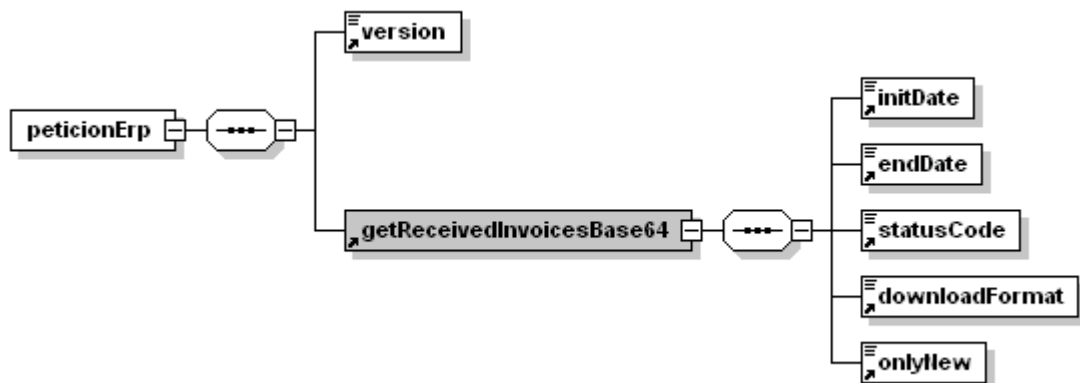


- `ticket`: ticket de sesión válido.

Ejemplo:

```
<ticket>Aquí un ticket</ticket>
```

- `java.lang.String xmlIn`: que contiene una cadena con formato XML con la siguiente estructura:



- `version`: Indica la versión de la plataforma que cumple el xml de petición.
- `initDate`: Fecha inicial de intervalo de búsqueda de facturas recibidas.
- `endDate`: Fecha final de intervalo de búsqueda de facturas recibidas.
- `statusCode`: Estado de la factura.
- `downloadPending`: Formato en que sean descargar las facturas indicadas. Se permiten los siguientes valores catalogados en archivo de constantes:

IDENTIFICADOR	LITERAL
CM10	CM10
COPIA	Copia del original firmada digitalmente
PDF	Un SOLO fichero PDF
UBL10	UBL 1.0
FACTURAE30	FACTURAE 3.0
FACTURAE31	FACTURAE 3.1
FACTURAE32	FACTURAE 3.2



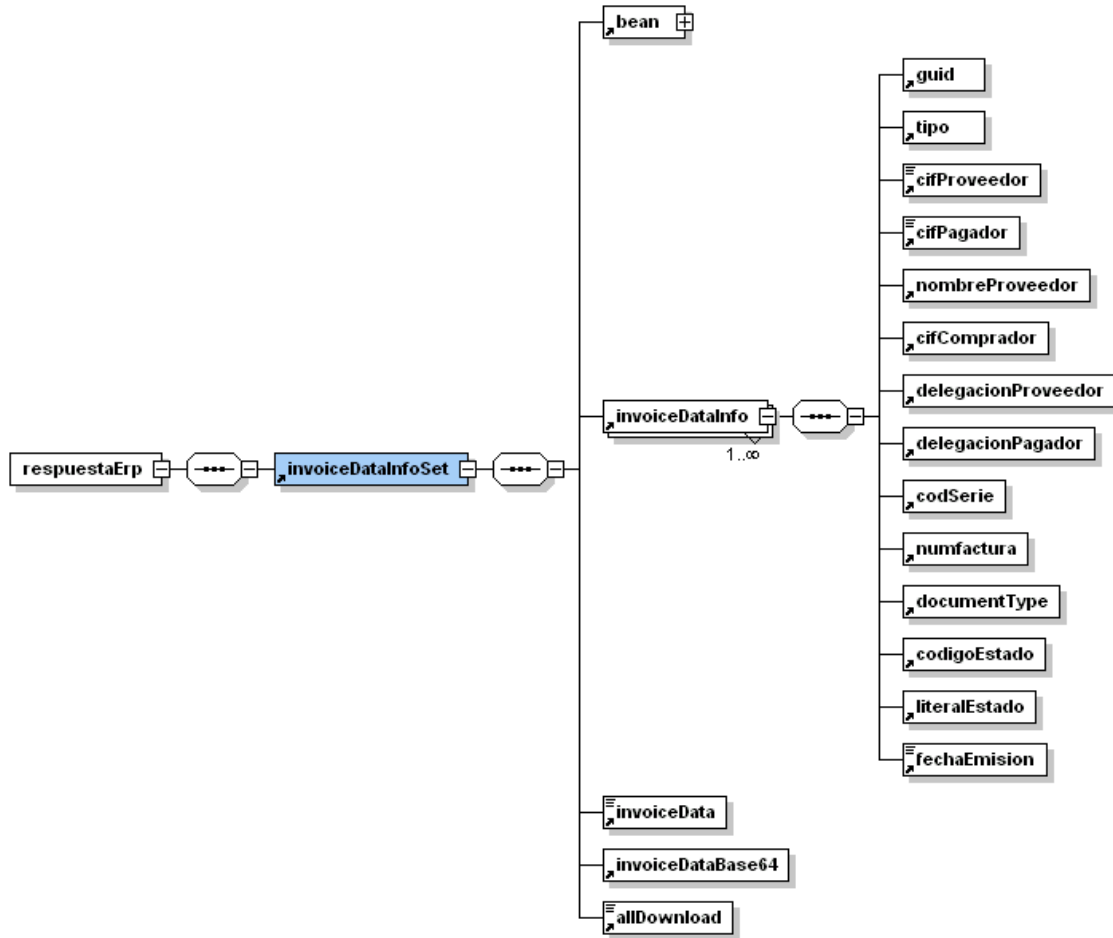
- o *onlyNew*: Valor booleano que indica si se desean descargar todas las facturas recibidas entre *initDate* y *endDate* y que aún no han sido descargadas por el ERP (valor *true*) o bien si se desean descargar todas las facturas comprendidas entre dichas fechas independientemente de si han sido descargadas con anterioridad (valor *false*).

Ejemplo:

```
<peticionErp>
  <version>3.0</version>
  <getReceivedInvoicesBase64>
    <initDate>13/03/2011</initDate>
    <endDate>14/03/2011</endDate>
    <statusCode>3</statusCode>
    <downloadPending>>true</downloadPending>
    <onlyNew>>true</onlyNew>
  </getReceivedInvoicesBase64>
</peticionErp>
```

6.16.3 Parámetros de salida

`java.lang.String xmlOut`: una cadena con formato XML con la siguiente estructura:



- El árbol *bean*: contiene información acerca de la terminación de la invocación detallada en el punto 4.3.2.
- *invoiceDataInfoSet* Ejemplo:

```

<respuestaErp>
  <invoiceDataInfoSet>
    <bean>
      <error>Aquí true o false</error>
      <subCode>Aquí código del error si lo hay</subCode>
      <message>Aquí descripción del error si lo hay</message>
    </bean>
  </invoiceDataInfoSet>
  <invoiceDataInfo>
    <guid>0001</guid>
    <tipo>1</tipo>
    <cifProveedor>11111111H</cifProveedor>
    <cifPagador>22222222H</cifPagador>
    <nombreProveedor>Nombre del proveedor</nombreProveedor>
    <cifComprador>22222222H </cifComprador>
  </invoiceDataInfo>
</respuestaErp>

```



```

<delegacionProveedor>00_COMMON</delegacionProveedor>
<delegacionPagador></delegacionPagador>
<codSerie></codSerie>
<numfactura>0001</numfactura>
<documentType>380</documentType>
<codigoEstado>5</codigoEstado>
<literalEstado>Aceptada</literalEstado>
<fechaEmision>13/03/2011</fechaEmision>
</invoiceDataInfo>
<invoiceData>Componente DataHandler</invoiceData>
<invoiceDataBase64></invoiceDataBase64>
<allDownload>>true or false</allDownload>
</invoiceDataInfoSet>
</respuestaErp>

```

6.16.4 Posibles códigos de terminación

En el caso de que la operación se haya realizado correctamente, el campo *error* de la rama *bean* del xml devuelto será *false*. En caso contrario, para esta operación se contemplan los siguientes mensajes de error:

CÓDIGO DE TERMINACIÓN	LITERAL DESCRIPTIVO
WS_ERROR_002	Error interno realizando la operación solicitada
WS_ERROR_003	El ticket no es correcto
WS_ERROR_007	Los parámetros de la llamada no son correctos
WS_ERROR_020	Error, versión no soportada
WS_ERROR_021	El XML de entrada no es válido
WS_ERROR_024	Error al intentar descargar facturas en formato



ANEXO A

Constante typeId

```
TIPO_IDENTIFICACION_ID_FISCAL="ID_FISCAL";  
TIPO_IDENTIFICACION_ID_INTERNO="ID_INTERNO";  
TIPO_IDENTIFICACION_ID_EXTERNO="ID_EXTERNO";
```

URL

Pruebas

<http://svc.integracion.test.euskadi.net/ctxweb/w42ajEFactura>

Producción

<http://svc.integracion.euskadi.net/ctxweb/w42ajEFactura>