



Unión Europea
Fondo Europeo de
Desarrollo Regional

“Una manera de hacer Europa”




eniDoc Web Services

Manual de Integración

Versión: v06r00
Fecha: 13/03/2017



Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito de la Junta de Andalucía.

 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	<p>Consejería de Hacienda y Administración Pública</p>	<p>eniDoc Web Services</p> <p>Manual de Integración</p>
---	--	---


HOJA DE CONTROL

Título	Manual de Integración		
Entregable	Manual de Integración		
Nombre del Fichero	ENIDOCWS_MIC_Manual_Integración_v06r00.odt		
Autor	CHAP		
Versión/Edición	v06r00	Fecha Versión	13/03/17
Aprobado por	-	Fecha Aprobación	13/03/17
		Nº Total Páginas	42

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Área	Fecha del Cambio
v01r00	Versión inicial	CHAP	-	08/10/2015
v02r00	Actualización del documento	CHAP	-	18/11/2015
v03r00	Inclusión metadatos complementarios y acceso desde aplicaciones clientes HCV.	CHAP	-	29/12/2015
v04r00	Actualización del documento. Ampliación del servicio insertarDocumentoEni.	CHAP	-	28/01/2016
V05r00	Actualización del documento	CHAP	-	18/01/2017
V06r00	Actualización del documento	CHAP	-	13/03/2017

Nombre y Apellidos	Cargo	Área	Nº Copias
Manuel Perera Domínguez	Jefe de Servicio	Servicio de Coordinación de Administración Electrónica	1
Francisco González Guillén	Director de Proyecto	Servicio de Coordinación de Administración Electrónica	1
José Ignacio Cortés Santos	Director de Proyecto	Servicio de Coordinación de Administración Electrónica	1
Francisco Mesa Villalba	Director de Proyecto	Servicio de Coordinación de Administración Electrónica	1
Pedro José Casanova Luis	Jefe de Proyecto	UTE	1

	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--


ÍNDICE

1 - INTRODUCCIÓN.....	6
2 - DESCRIPCIÓN DE LAS OPERACIONES ENIDOCWS.....	8
2.1 - Operación de acceso a documento electrónico ENI (verificarFirma).....	8
2.1.1 - Estructura de la petición.....	8
2.1.2 - Flujo del servicio de integración.....	13
2.1.3 - Estructura de la respuesta.....	14
2.1.4 - Incorporación de metadatos complementarios.....	17
2.1.5 - Gestión de respuestas y errores en HCV como cliente.....	18
2.2 - Operación de solicitud de custodia de nuevo documento electrónico ENI (insertarDocumentoEni).....	19
2.2.1 - Flujo del servicio de integración.....	19
2.2.2 - Estructura de la petición.....	19
2.2.3 - Estructura de la respuesta.....	20
2.2.4 - Gestión de respuestas y errores.....	21
3 - IMPLEMENTACIÓN DE LAS OPERACIONES ENIDOCWS COMO SERVICIO OFRECIDO.....	22
3.1 - Incorporación de la librería “enidocWS”.....	23
3.1.1 - Dependencias software.....	23
3.1.2 - Fichero “web.xml”.....	24
3.1.3 - Parámetros de configuración.....	26
3.1.3.1 - Fichero de propiedades.....	26
3.1.3.2 - Parámetros de contexto.....	26
3.1.4 - Implementación de la clase invocada por el WS.....	28
3.1.4.1 - A) Acceso al documento ENI.....	28
3.1.4.2 - B) Inserción del Documento ENI.....	30
3.2 - Despliegue de la aplicación “enidocWS”.....	31
3.3 - Cumpliendo los xml de intercambio de información. Petición y Respuesta.....	31
4 - IMPLEMENTACIÓN DE LAS OPERACIONES ENIDOCWS COMO CLIENTE CONSUMIDOR.....	33

5 - ANEXO I. GENERACIÓN DE CÓDIGO SEGURO DE VERIFICACIÓN	34
6 - ANEXO II. UTILIZACIÓN DE OTRO FRAMEWORK REST. RESTEASY.....	35
7 - ANEXO III. COMPROBACIÓN DE LA INTEGRACIÓN CON ENIDOCWS	36
7.1 - Postman: Cliente REST para Chrome.....	36
8 - ANEXO IV. VALIDACIÓN DOCUMENTO ENI.....	39
9 - ANEXO V. TIPIFICACIÓN DEL CÓDIGO DE RESPUESTA.....	40
10 - REFERENCIAS.....	42

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Diagrama de clases EnidocWS (verificarFirma).....	6
Ilustración 2. Diagrama de clases Peticion.....	7
Ilustración 3. Diagrama de clases Respuesta EniDoc.....	11
Ilustración 4. Diagrama de clases EnidocWS (insertarDocumentoEni).....	16
Ilustración 5. Diagrama de clases Respuesta DatosEni.....	17
Ilustración 6. Diagrama de integración con el Servicio Web.....	18
Ilustración 7. Pantalla del complemento de Chrome.....	27
Ilustración 8. Formulario de autenticación.....	27
Ilustración 9. Especificar los parámetros de inserción.....	28
Ilustración 10. Especificar los parámetros de inserción.....	28
Ilustración 10. Respuesta correcta del servicio verificarFirma.....	29
Ilustración 11. Respuesta correcta del servicio insertarDocumentoEni.....	29

 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	<p>Consejería de Hacienda y Administración Pública</p>	<p>eniDoc Web Services</p> <p>Manual de Integración</p>
---	--	---

1 INTRODUCCIÓN

El objetivo del presente documento es servir de guía de integración para los repositorios de documentos electrónicos y/o sistemas que intervengan en procesos de intercambio de documentos electrónicos haciendo uso del protocolo denominado enidocWS.

La Norma Técnica de Interoperabilidad de Documento electrónico define una estructura XML de intercambio en la que es posible incorporar los componentes que constituyen un documento electrónico: contenido, firmas y metadatos mínimos obligatorios. La definición de esta estructura XML de intercambio facilita la interoperabilidad de los sistemas de tramitación y los repositorios de documentos electrónicos al emplear todos ellos una misma sintaxis y semántica. Sin embargo, el Esquema Nacional de Interoperabilidad es incompleto ya que define “*qué*” se intercambia, pero no se realiza una definición de “*cómo*” ha de producirse el intercambio, lo cual deja abierta la puerta a múltiples y variadas implementaciones que imposibilitan el fin último del ENI: la interoperabilidad entre los sistemas.


Dado que el Esquema Nacional de Interoperabilidad no define un protocolo de intercambio de documentos electrónicos entre sistemas, la Junta de Andalucía define un nuevo protocolo de servicios web que al menos permita esta interoperabilidad internamente en su ámbito de actuación. Este protocolo de intercambio, denominado “enidocWS”, se basa en tecnología REST e incorpora las siguientes operaciones:

1. Una operación de acceso: un sistema cliente solicita a un repositorio de documentos electrónicos el acceso a uno de ellos a partir de su código seguro de verificación.
2. Una operación de solicitud de custodia: un sistema cliente solicita a un repositorio de documentos electrónicos la custodia de un nuevo documento. En esta operación, el repositorio de documentos receptor de la petición de custodia es responsable de la asignación del CSV y los metadatos mínimos obligatorios al nuevo documento capturado.


La implementación efectiva de estas operaciones por parte de un repositorio de documentos o por parte de un sistema de información dependerá de sus requisitos funcionales. Así, por ejemplo, un repositorio que deba integrarse con la Herramienta Centralizada de Verificación (HCV) deberá ofrecer al menos la operación de acceso, de modo que en los procesos de validación de documentos desde HCV, ésta pueda acceder al repositorio documental donde reside el documento que se pretende verificar.

El componente “enidocWS” tiene, por tanto, una doble vertiente de uso:

- ✦ Deberá ser implementado como servicio por los repositorios de documentos electrónicos que que deban permitir el acceso a sus documentos y/o atender solicitudes de custodia de nuevos documentos electrónicos. Como ejemplo, puede considerarse un repositorio de documentos integrado con la Herramienta Centralizada de Verificación, el cual deberá ofrecer al menos la operación de acceso a un documento.
- ✦ Deberá ser implementado, como cliente, por parte de los sistemas que deban acceder a documentos custodiados en un repositorio o que deban requerir la custodia de uno nuevo. Como ejemplo, puede considerarse a la herramienta Compul@ que como cliente enidocWS puede solicitar a repositorios externo la custodia de los documentos que digitalizan sus usuarios.

 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	<p>Consejería de Hacienda y Administración Pública</p>	<p>eniDoc Web Services</p> <p>Manual de Integración</p>
---	--	---

	OPERACIÓN DE ACCESO	OPERACIÓN DE SOLICITUD DE CUSTODIA
REPOSITORIO DE DOCUMENTOS	Deberá ofrecer esta operación si debe permitir el acceso desde otros sistemas. Ejemplo; permitir el acceso requerido desde HCV. La HCV también ofrece esta operación a otros sistemas.	Deberá ofrecer esta operación solo si debe admitir documentos recibidos desde otro sistema. Ejemplo: los repositorios que acepten documentos capturados en la herramienta Compuls@.
SISTEMA CLIENTE DE UN REPOSITORIO DE DOCUMENTOS:	Deberá ser cliente de esta operación si necesita acceder a documentos custodiados en otros sistemas. La HCV implementa esta operación como cliente para acceder a los repositorios dados de alta que custodian los documentos.	Deberá implementar esta operación como cliente si debe solicitar la custodia de un nuevo repositorio a un repositorio de documentos. Ejemplo: la herramienta Compulsa implementa un cliente de esta operación.

 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

2 DESCRIPCIÓN DE LAS OPERACIONES ENIDOCWS

Los sistemas que deban ofrecer como servicio las operaciones “enidocWS” tienen a su disposición una librería de servicios web (archivo jar). Esta librería implementa las operaciones para permitir el acceso a un documento ENI o para atender solicitudes de custodia de nuevos documentos. Como alternativa a la librería, existe también una distribución en formato de aplicación web (archivo war) para que sea desplegada en un contenedor web JEE, según las necesidades de cada sistema/repositorio.

Los servicios web donde se implementan las operaciones enidocWS han sido desarrollados con tecnología REST (Representational State Transfer, técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web), haciendo uso del framework Jersey para implementación de Servicios Web RESTful en Java.

2.1 Operación de acceso a documento electrónico ENI (verificarFirma)

2.1.1 Estructura de la petición

A continuación se muestra el diagrama UML con la relación entre las entidades que conforman la petición a la operación de integración:

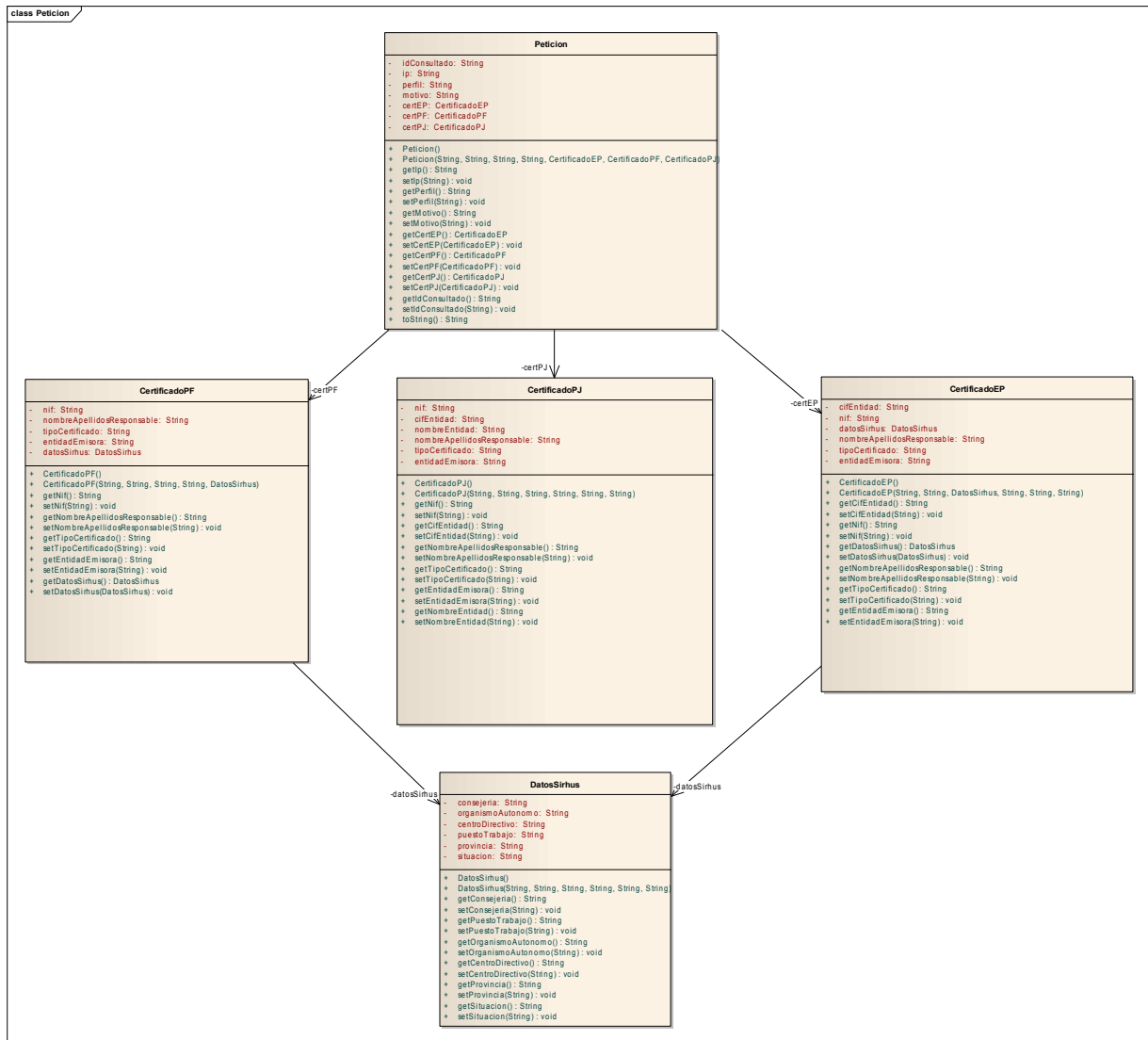


Ilustración 2. Diagrama de clases Petición

El detalle de los campos que componen el objeto **Peticion** que construirá el sistema cliente de la operación de acceso a un documento electrónico ENI y que recibirá como parámetro de entrada el repositorio destino “enidocWS” donde éste reside es el siguiente:

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
peticion	Solicitud de acceso a documento electrónico ENI	Peticion	Obligatorio

El tipo de objeto **Peticion** se compone de los siguientes objetos Java:

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
idConsultado	Identificador de la petición. Será el código seguro de verificación (CSV) del documento solicitado.	String	Obligatorio
ip	Identificador del host cliente que realiza la petición.	String	Obligatorio
perfil	<p>Cadena con uno de los siguientes valores, que indicará el contexto que indica el tipo de actor que motiva la consulta realizada desde el sistema cliente.</p> <ul style="list-style-type: none"> • 1: Ciudadanía • 2: Personal empleado público de la Junta de Andalucía en el ejercicio de sus funciones • 3: Otro personal empleado público en el ejercicio de sus funciones • 4: Consulta automatizada procedente de repositorio o sistema de Consjería o entidad de la Junta de 	String	Obligatorio

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
	Andalucía.		
motivo	Motivo por el cual el usuario desea acceder al documento.	String	Opcional
certEP	Certificado de Empleado Público	CertificadoEP	Opcional
certPF	Certificado de Persona Física	CertificadoPF	Opcional
certPJ	Certificado de Persona Jurídica	CertificadoPJ	Opcional
dir3	Código DIR3 del organismo que genera la solicitud.	String	Opcional
dir3Vigente	Determinará si el DIR3 vinculado al sistema origen se encuentra vigente en el momento de realizar la consulta.	boolean	Opcional
sistemaOrigen	Identificador del sistema que desencadena la consulta.	String	Opcional
nombreCliente	Nombre descriptivo del sistema que desencadena la consulta.	String	Opcional

En los casos de implementar la operación de acceso como servicio ofrecido a la Herramienta Centralizada de Verificación, es necesario que el campo “**idConsultado**” contenedor del CSV consultado **coincida en sus primeros 5 caracteres con los del identificador único del sistema/repositorio proporcionado en el alta**. Para el resto de caracteres, se recomienda usar el código seguro de verificación propuesto en el Anexo I del presente documento. No obstante, cada administración puede diseñar el proceso de generación según sus necesidades, asegurando en cualquier caso los principios de impredecibilidad, uniformidad, resistencia a colisiones e irreversibilidad.

Detalle del objeto **CertificadoEP**:

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
cifEntidad	CIF de la entidad	String	Obligatorio
nif	NIF del empleado	String	Obligatorio
nombreApellidosResponsable	Nombre y apellidos	String	Obligatorio
tipoCertificado	Tipo de certificado	String	Obligatorio
entidadEmisora	Nombre de la entidad	String	Obligatorio

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
	emisora del certificado		
datosSirhus	Datos devueltos por el sistema SIRHUS	DatosSirhus	Opcional

Detalle del objeto **CertificadoPF**:

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
nif	NIF de la persona	String	Obligatorio
nombreApellidosResponsable	Nombre y apellidos	String	Obligatorio
tipoCertificado	Tipo de certificado	String	Obligatorio
entidadEmisora	Nombre de la entidad emisora del certificado	String	Obligatorio
datosSirhus	Datos devueltos por el sistema SIRHUS	DatosSirhus	Opcional

Detalle del objeto **CertificadoPJ**:

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
nif	NIF de la persona	String	Obligatorio
cifEntidad	CIF de la entidad representada	String	Obligatorio
nombreEntidad	Nombre de la entidad representada	String	Obligatorio
nombreApellidosResponsable	Nombre y apellidos	String	Obligatorio
tipoCertificado	Tipo de certificado	String	Obligatorio
entidadEmisora	Nombre de la entidad emisora del certificado	String	Obligatorio

Detalle del objeto **DatosSirhus**:

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
consejeria	Nombre de la Consejería	String	Obligatorio
organismoAutonomo	Entidad a la que pertenece	String	Obligatorio

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
centroDirectivo	Centro directivo	String	Obligatorio
puestoTrabajo	Puesto de trabajo	String	Obligatorio
provincia	Provincia	String	Obligatorio
situacion	Situación	String	Obligatorio

Se incluye en la petición una cabecera con las credenciales de autorización (usuario y clave) en la petición de la operación de acceso a documento. Estas credenciales serán las proporcionadas por el sistema/repositorio tercero para su inclusión en la herramienta que consuma el servicio.

2.1.2 Flujo del servicio de integración

Se muestra el diagrama UML que representa el flujo de integración y la relación entre las distintas clases que conforman la implementación de esta operación en modo de servicio web:

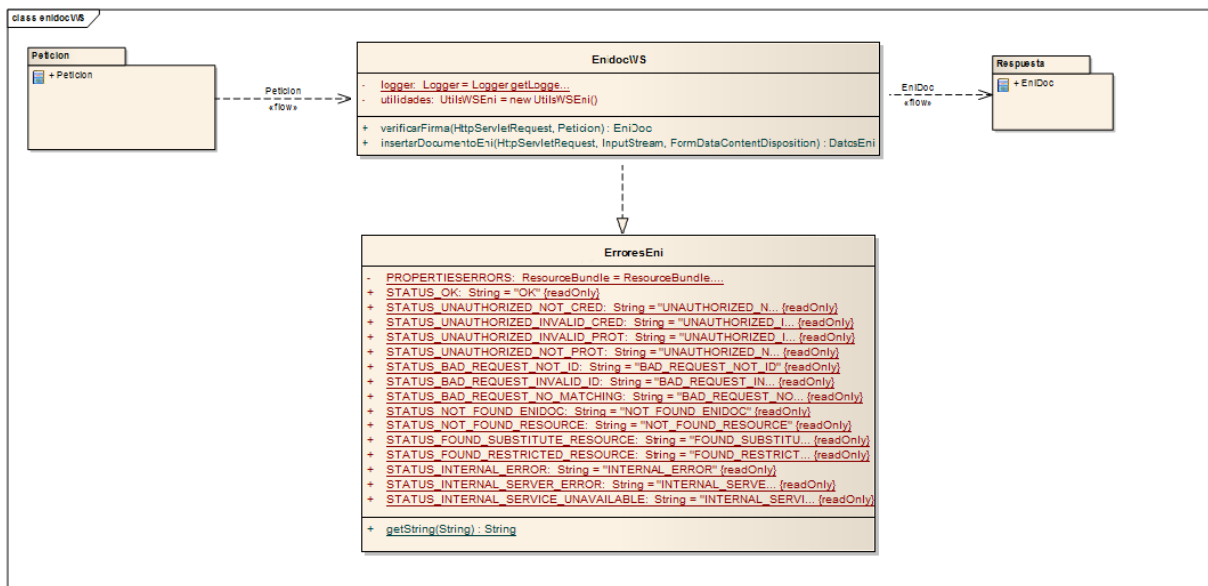


Ilustración 1. Diagrama de clases EniDocWS (verificarFirma)

Los sistemas/repositorios que ofrezcan el servicio deberán implementar un método (véase apartado 3 del presente documento) que recibe como parámetro de entrada un objeto de tipo **Petición** y devuelve como respuesta un objeto de tipo **EniDoc**, cuyo contenido será un documento ENI acompañado de un conjunto de metadatos adicionales e información de contexto.

El detalle de los campos del objeto de respuesta **EniDoc** que deben devolver los sistemas/repositorios tras la operación de consulta por parte del sistema cliente es el siguiente:

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
documentoENI	<p>Documento electrónico ENI conforme al Anexo II de la NTI de Documento Electrónico¹.</p> <p>En aquellos casos en los que el repositorio/sistema que custodia el documento no esté en disposición de asignar identificadores normalizados de documentos según lo dispuesto en la Norma Técnica de Interoperabilidad de Documento Electrónico, como por ejemplo en el ámbito de las entidades no incorporadas al Directorio Común de Unidades Orgánicas, el metadato "Órgano" deberá adoptar el valor "XXXXXXXX".</p>	Fichero [cadena de bytes cuyo contenido se corresponde con lo definido en la Norma Técnica de Interoperabilidad del Documento Electrónico].	Obligatorio
informeFirma	Documento PDF representante del informe de firma del documento consultado. (<i>Copia papel auténtica de documento electrónico</i>).	Fichero [cadena de bytes]	Opcional
textoLibre	Texto libre de respuesta para ser mostrado al usuario final. En HCV, si este campo se completa por parte por el sistema/repositorio consultado, se mostrará al usuario única y exclusivamente dicho texto, ignorando el resto de la información.	String	Opcional

¹ Si se asigna un valor al metadato mínimo obligatorio **Identificador** del Documento Electrónico, deberá cumplir con los requisitos definidos por la Norma Técnica de Interoperabilidad del Documento Electrónico (<Idioma>_<Órgano>_<AAAA>_<ID_especifico>), donde:

1. <Idioma>: código del idioma o lengua cooficial del documento: - ES: español
2. <Órgano>: Código alfanumérico único para cada órgano/unidad/oficina extraído del Directorio Común gestionado por el Ministerio de Hacienda y Administraciones Públicas (Longitud: 9 caracteres).
3. <AAAA>: Año de la fecha de captura del documento. (Longitud: 4 caracteres).
4. <ID_especifico>: Código alfanumérico que identifica de forma única al documento dentro de los generados por la administración responsable. (Longitud: 30 caracteres).

Es indispensable que los 5 primeros caracteres sean coincidentes con los del identificador único del sistema/repositorio proporcionado por la herramienta destino. Para el resto de caracteres, cada administración puede diseñar el proceso de generación según sus necesidades, asegurando en cualquier caso su unicidad y que **no coincidan con el valor del código seguro de verificación**.

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
textoAdicional	Texto libre adicional de respuesta para ser mostrado al usuario final, junto con los demás resultados del proceso.	String	Opcional
generadorFirma	En caso de firma electrónica basada en certificado, componente o herramienta utilizada para su generación.	String	Opcional
marcaTiempo	Información de utilidad para las firmas electrónicas que no incorporen sello de tiempo. Marca de tiempo de generación / incorporación del documento al sistema/repositorio y conforme al artículo 15 del Real Decreto 4/2010, de 8 de enero, por el que se regula el Esquema Nacional de Interoperabilidad en el ámbito de la Administración Electrónica.	Date	Opcional
validacionCertificadosFirma	Validación de certificados en el momento de realizar la firma basada en certificado electrónico.	boolean	Opcional
selloTiempoRequerido	Indica si el sello de tiempo es o no requerido.	boolean	Opcional
identificadorDocumentoSustituto	Identificador normalizado que, en su caso, hubiera sustituido al documento al cual se intenta acceder.	String	Opcional
depositoSeguro	Indica si la firma electrónica está almacenada en un depósito seguro. ²	boolean	Opcional
descargarFirmaydocumentoENI	Indica si se desea habilitar que la persona usuaria de la herramienta pueda descargar el correspondiente fichero de firma electrónica y el documento ENI. ³	boolean	Opcional
tipoFirmaNoNormalizada	Se considera este campo, en el caso de que el tipo de firma no esté contemplado en la adaptación al NTI ⁴	TipoFirmaNoNormalizada [Tipificado:	Opcional

² Según epígrafes 2.b y 5.b del apartado II.7 de la Norma Técnica de Interoperabilidad de Política de Firma Electrónica y de certificados de la Administración.

³ Ello, siempre que el sistema/repositorio se haya configurado previamente para habilitar esta posibilidad.

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
		- TF97 - TF98 - TF99]	
campoAdicional	Campo adicional que se utilizará para posibles necesidades futuras.	String	Opcional
metadatosComplementarios	Campo adicional donde se podrán asignar metadatos complementarios sobre el documento no previstos en la NTI de Interoperabilidad de Documento Electrónico. (Ver apartado 2,1,4)	Properties	Opcional

2.1.4 Incorporación de metadatos complementarios

En el objeto de respuesta es posible incorporar metadatos complementarios vinculados al documento electrónico que se incorpora en la respuesta. Estos metadatos complementarios se incorporarán en el campo opcional “metadatosComplementarios” en la forma de listado de pares clave-valor. El listado de metadatos complementarios se empaquetará en la forma de un objeto Java de tipo `java.util.Properties`.


En el caso de la HCV como cliente de un repositorio, los metadatos complementarios solo se mostrarán en la interface de usuario cuando así expresamente lo indique expresamente el repositorio. Para ello, existirá un metadato complementario adicional “mostrarhcv” cuyo valor determinará qué metadatos complementarios serán visibles en HCV. A modo de ejemplo, considérese el siguiente contenido de metadatos:

⁴ Identificar que es necesario incluir nuevos valores de forma que se puedan informar formatos de firma no incluidos en la Norma Técnica de Interoperabilidad de Política de Firma Electrónica y de certificados de la Administración, sin perjuicio de que no se deba continuar con la utilización de los mismos. Se han incluido los tipos de firmas:

- **TF97:** para formatos de firma CMS
- **TF98:** para formatos de firma XMLDSig
- **TF99:** para otros formatos de firma (por ejemplo PKCS#7).

En el caso de que este campo venga informado, la herramienta de verificación informará a los usuarios el tipo de firma indicada por el valor de este campo. Es decir, el valor de este campo prevalece sobre el valor del campo documentoENI en el que se indica el tipo de firma, el cuál es obligatorio. Se establece la siguiente tabla de mapeos:

Tipo de Firma	Valor en el campo documentoENI	Valor campo otroTipoFirma
NoNormalizada	CMSTF02TF97XMLDSigTF03TF98IndefinidoTF02TF99	

	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

MOSTRARHCV=FACTURA,IMPORTE

FACTURA=2015/123/22211144

PROCEDIMIENTO=Sistema de registro de XXXX

IMPORTE=25000

En base a estos metadatos complementarios, la HCV mostrará en su interface de usuario los valores de los metadatos "FACTURA" e "IMPORTE" y ocultará el metadato "PROCEDIMIENTO".

En ausencia del metadato "mostrarhcv" no se mostrará ningún metadato complementario en la interface gráfica de usuario de HCV. No obstante, si existe el metadato "mostrarhcv" con valor vacío se mostrarán todos los metadatos incorporados. Por otra parte, si se requiere mostrar los metadatos en un orden definido, es necesario indicar uno a uno el en el orden requerido los metadatos en el valor del metadato "mostrarhcv".

2.1.5 Gestión de respuestas y errores en HCV como cliente

Los mensajes de respuesta y de error se producirán en base a las siguientes situaciones:

- Si el sistema/repositorio es consultado por un código seguro de verificación que no se vincula a ninguno de los documentos electrónicos que custodia, HCV informará a sus usuarios de que el documento consultado no se ha encontrado.
- Si la HCV no puede determinar el sistema/repositorio candidato a la custodia del CSV consultado, se indicará al usuario que el documento no se ha encontrado.
- Si la HCV trata de comunicarse con el sistema/repositorio que custodia el documento y no es posible o éste devuelve un error, se indicará al usuario que en ese momento debido a problemas técnicos no es posible el acceso al documento consultado.
- Si la HCV se comunica con el sistema/repositorio correctamente y éste responde indicando el identificador normalizado del documento que sustituye al documento al cual se intenta acceder, cumplimentando el parámetro "identificadorDocumentoSustituto", se indicará al usuario únicamente el identificador normalizado del documento sustitutivo, ignorando el resto de la información.
- Si la HCV se comunica con el sistema/repositorio correctamente y éste determina que no procede otorgar el acceso al documento para la persona/sistema usuaria de la herramienta, cumplimenta el parámetro "textoLibre" expresando un texto informativo que se le mostrará al usuario. Ha HCV ignorará el resto de la información.

Adicionalmente se recibe un objeto Petición, similar al descrito en el apartado 2.1.2, que dota de contexto a la solicitud de custodia de un documento electrónico.

2.2.3 Estructura de la respuesta

El siguiente diagrama representa la clase de Respuesta a devolver por los sistemas/repositorios:

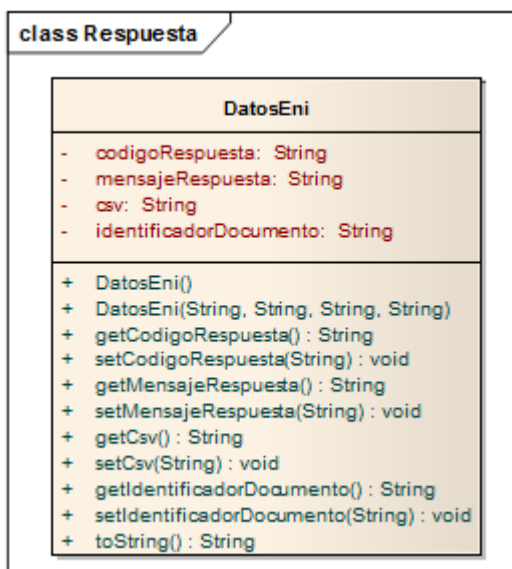



Ilustración 5. Diagrama de clases Respuesta DatosEni

El detalle de los campos del objeto de respuesta **DatosEni** que debe devolver los sistemas/repositorios que implementan el servicio con la operación de solicitud de custodia de documento es el siguiente:

NOMBRE	DESCRIPCIÓN	TIPO DATO	CARÁCTER
csv	Código de verificación generado por la aplicación que realiza la custodia del documento, que permitirá un posterior acceso al documento ENI.	String	Obligatorio
identificadorDocumento	Valor que el repositorio que acepta al documento asigna a su metadato mínimo obligatorio "Identificador".	String	Opcional

 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

2.2.4 Gestión de respuestas y errores

Los mensajes de respuesta y de error se producirán en base a las siguientes situaciones:

- Si el sistema que custodiará el documento no reconoce como autorizado al sistema petionario de la solicitud, se indicará al éste que el documento no se ha almacenado.
- Si el sistema cliente creador de la solicitud de custodia trata de comunicarse con el sistema/repositorio que custodiará el documento y no es posible o éste devuelve un error, se informará al usuario de que en ese momento debido a problemas técnicos no es posible almacenar el documento.
- El sistema que ofrece el servicio con la operación de solicitud de custodia de nuevos documentos podrá rechazar una petición de custodia en base a lógica propia que tenga implementada. Se indicará al usuario que el documento no se ha almacenado.

3 IMPLEMENTACIÓN DE LAS OPERACIONES ENIDOCWS COMO SERVICIO OFRECIDO

NOTA: Los repositorios o sistemas de documentos electrónicos integrados con HCV deberán implementar al menos como servicio la operación de acceso a documento electrónico.


Un sistema/repositorio que deba ofrecer como servicio las operaciones "enidocWS" dispone de las siguientes alternativas:

- 1) Incorporar una librería que ofrecerá el servicio en el sistema/repositorio (véase 3.1).
- 2) Desplegar una nueva aplicación que actúa de fachada de interconexión entre la herramienta destino y la fuente de los datos (véase 3.2).
- 3) Implementación propia conforme a las estructuras XML de intercambio de información petición y respuesta (véase 3.3).



Ilustración 6. Diagrama de integración con el Servicio Web

NOTA: La obtención de los recursos necesarios para la integración se materializará mediante la zona de "descargas privadas" de la web de soporte de administración electrónica de la Junta de Andalucía. Para ello, se deberá realizar la correspondiente solicitud en el apartado "descargas de software" en la herramienta de "gestión de incidencias, consultas y solicitudes" disponible en la citada web.

	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

3.1 Incorporación de la librería “enidocWS”

Si se desea utilizar esta alternativa de integración hay que tener en cuenta, como requisito técnico, que el sistema/repositorio debe ejecutarse sobre una **JDK 6 o superior**.

Para la integración del servicio web será necesario seguir los siguientes pasos:

3.1.1 Dependencias software

A continuación se exponen las dependencias software para la correcta integración con la librería, identificando las actuaciones a realizar según se disponga de una aplicación mavenizada o no.

A. Establecer dependencias sin maven: dirigido a aplicaciones no mavenizadas.

Son necesarias una serie de dependencias (ficheros jar) que deben estar en el classpath de ejecución de la librería principal para su correcto funcionamiento. La librería que implementa el servicio se denomina **enidocWS-x.x.jar** y las dependencias mencionadas son las siguientes:

- com.google.guava:guava:jar:14.0.1
- com.sun.xml.wsit:xmlsec:jar:1.1
- commons-lang:commons-lang:jar:2.3
- commons-logging:commons-logging:jar:1.1.1
- es.juntadeandalucia.nti:conversor-eni:jar:1.3.0
- javax.annotation:javax.annotation-api:jar:1.2
- javax.servlet:javax.servlet-api:jar:3.1.0
- javax.validation:validation-api:jar:1.1.0.Final
- javax.ws.rs:javax.ws.rs-api:jar:2.0
- javax.xml.stream:stax-api:jar:1.0-2
- log4j:log4j:jar:1.2.17
- org.codehaus.castor:castor-core:jar:1.3.2
- org.codehaus.castor:castor-xml:jar:1.3.2
- org.glassfish.hk2.external:asm-all-repackaged:jar:2.2.0-b21
- org.glassfish.hk2.external:cglib:jar:2.2.0-b21
- org.glassfish.hk2.external:javax.inject:jar:2.2.0-b21
- org.glassfish.hk2:hk2-api:jar:2.2.0-b21
- org.glassfish.hk2:hk2-locator:jar:2.2.0-b21
- org.glassfish.hk2:hk2-utils:jar:2.2.0-b21
- org.glassfish.hk2:osgi-resource-locator:jar:1.0.1
- org.glassfish.jersey.containers:jersey-container-servlet-core:jar:2.4.1
- org.glassfish.jersey.core:jersey-client:jar:2.4.1
- org.glassfish.jersey.core:jersey-common:jar:2.4.1
- org.glassfish.jersey.media:jersey-media-multipart:jar:2.4.1
- org.glassfish.jersey.core:jersey-server:jar:2.4.1
- org.hamcrest:hamcrest-core:jar:1.3
- org.jvnet.mimepull:mimepull:jar:1.9.3
- stax:stax:jar:1.2.0
- stax:stax-api:jar:1.0.1

NOTA: En el caso de utilizar un servidor Tomcat 6 o superior, no se deberá incluir la dependencia “**javax.servlet:javax.servlet-api:jar:3.1.0**”.

B. Establecer dependencias con Maven: dirigido a aplicaciones mavenizadas.

El componente “enidocWS” puede ser incluido en aplicaciones desarrolladas en Maven añadiendo la siguiente dependencia al fichero **pom.xml** de la aplicación:

```
<dependency>
  <groupId>es.juntadeandalucia.nti</groupId>
  <artifactId>enidocWS</artifactId>
  <version>x.x</version>
</dependency>
```

Este componente puede encontrarse en el Repositorio Maven del Servicio de Coordinación de Administración Electrónica (<https://ws024.juntadeandalucia.es/ae/adminelec/repositoriomaven>).

Para que Maven obtenga el artefacto que contiene el componente y resuelva automáticamente sus dependencias, es necesario incluir del mismo modo, la referencia al repositorio de software donde se encuentra el componente:

```
<repository>
  <id>RepositorioMavenSCAE</id>
  <name>Repositorio de la Junta de Andalucía</name>
  <url>https://ws024.juntadeandalucia.es/maven/repository/internal</url>
</repository>
```

NOTA: El Repositorio Maven del Servicio de Coordinación de Administración Electrónica de la Consejería de Hacienda y Administración Pública está accesible a través del protocolo HTTPS utilizando un certificado SSL emitido por la Fábrica Nacional de Moneda y Timbre. Para que el cliente Maven confíe en el dominio ws024.juntadeandalucia.es debe incorporarse al fichero “cacerts” de la JVM cliente el certificado raíz “AC Componentes” de FNMT.

3.1.2 Fichero “web.xml”

Es requisito indispensable para la llevar a cabo la integración que se defina el siguiente servlet, como cualquier otro integrante de la aplicación web cliente, en el fichero “web.xml”. Se muestra a continuación un ejemplo del fichero con la definición del servlet mencionado:


```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns=http://java.sun.com/xml/ns/j2ee
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>Web Service Enidoc</display-name>
  <servlet>
    <servlet-name>EnidocService</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
      <param-name>jersey.config.server.provider.classnames</param-name>
      <param-value>org.glassfish.jersey.media.multipart.MultiPartFeature;
es.juntadeandalucia.nti.ws.eni.services.EnidocWS</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>EnidocService</servlet-name>
    <url-pattern>/rest/eni/*</url-pattern>
  </servlet-mapping>
</web-app>
```


NOTA: El *url-pattern* especificado en el *servlet-mapping* debe formar parte de la url donde despliegue el servicio. Es decir:

```
<servlet-mapping>  
  <servlet-name>EnidocService</servlet-name>  
  <url-pattern>/rest/eni/*</url-pattern>  
</servlet-mapping>
```

En el formulario de solicitud de alta en la hcv el campo *Dirección de acceso al servicio* debe concluir con dicho patrón:

Por ejemplo: `http://RUTAAPLICACION/rest/eni/`

	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

3.1.3 Parámetros de configuración

Para la correcta configuración del componente, se presentan dos modalidades:

- Mediante un fichero de propiedades.
- Mediante parámetros de contexto definidos en el fichero "web.xml".

3.1.3.1 Fichero de propiedades

Con el objetivo de facilitar la integración se establece un fichero de propiedades, denominado "enidocWS.properties" que contiene los argumentos de las peticiones a realizar.

A continuación se muestra un ejemplo del fichero que debe incluirse en el classpath donde se ejecute el servicio, donde se observan los parámetros a cumplimentar:

```
# Información de correspondencia entre la herramienta y el repositorio/sistema integrado
# usuario registrado y autorizado
usuarioWS = usuario
# password registrado y autorizado
passwordWS = pass
# clase (ruta del paquete separado por puntos '.') donde se implementa el método de recuperación
claseWS = paquete.clase.MiClase
# nombre del método que recibe como parámetro el objeto Petición y devuelve Enidoc
metodoWS = nombreMetodo
# clase (ruta del paquete separado por puntos '.') donde se implementa el método de inserción
claseWSInsertar = paquete.clase.MiClaseInsertar
# nombre del método que recibe como parámetro el objeto InputStream y devuelve DatosEni
metodoWSInsertar = nombreMetodoInsertar
# Valor true/false para obviar o no la limitación de 30 caracteres del CSV
obviarCaracteresCSV = valor
```

3.1.3.2 Parámetros de contexto

En caso de optar por no realizar la configuración directamente mediante el fichero de propiedades, se ofrece la posibilidad de realizar la configuración necesaria a través de parámetros de contexto. Para ello, será necesario especificar dos parámetros de contexto en el fichero "web.xml" del sistema que identifiquen, la clase y el método que instanciarán las propiedades necesarias para el funcionamiento del servicio web. Los parámetros deben ser:

1. **claseConfiguracionEnidocWS:** cuyo valor se corresponde con la ruta de la clase donde se instanciará el método de recuperación de las propiedades.
2. **metodoObtencionPropiedadesEnidocWS:** cuyo valor se corresponde con el nombre del propio método encargado de obtener las propiedades.

```
<context-param>
  <description>Indica la clase que contiene el método de obtención de las propiedades necesarias
para EnidocWS</description>
  <param-name>claseConfiguracionEnidocWS</param-name>
  <param-value>paquete.clase.MiClaseConfiguracion</param-value>
</context-param>

<context-param>
  <description>Indica el método que obtiene las propiedades necesarias para EnidocWS</description>
  <param-name>metodoObtencionPropiedadesEnidocWS</param-name>
  <param-value>nombreMetodoObtenerParametros</param-value>
</context-param>
```

La clase deberá llamarse como se indica en la etiqueta param-value definida anteriormente, y estar ubicada en el paquete indicado. Es estrictamente necesario que el método de obtención de las propiedades que se defina y se haya establecido en el parámetro de contexto sea **estático**. Dicho método será el encargado de dar valor a un atributo de tipo **java.util.Properties**, cumplimentando las propiedades indicadas en el apartado anterior.

Se muestra un ejemplo básico de construcción de la clase y el método necesarios:

```
package paquete.clase;

import java.util.Properties;

public class MiClaseConfiguracion{

    public static Properties properties = null;

    public static Properties nombreMetodoObtenerParametros() {
        properties = new Properties();
        // usuario registrado y autorizado
        properties.setProperty("usuarioWS", "usuario");
        // password registrado y autorizado
        properties.setProperty("passwordWS", "pass");
        // clase (ruta del paquete separado por puntos '.') donde se implementa el método de
recuperación
        properties.setProperty("claseWS", "paquete.clase.MiClase");
        // nombre del método que recibe como parámetro el objeto Peticion y devuelve Enidoc
        properties.setProperty("metodoWS", "nombreMetodo");
        // clase (ruta del paquete separado por puntos '.') donde se implementa el método de
inserción
        properties.setProperty("claseWSInsertar", "paquete.clase.MiClaseInsertar");
        // nombre del método que recibe como parámetro el InputStream y devuelve DatosEni
        properties.setProperty("metodoWSInsertar", "nombreMetodoInsertar");
        // Para obviar la limitación de 30 caracteres del CSV
        properties.setProperty("obviarCaracteresCSV", "valor");
        return properties;
    }
}
```

NOTA: Se permite la configuración múltiple de usuarios de acceso al Servicio Web. Esto es, múltiples usuarioWS-passwordWS, con el fin de que un sistema/repositorio que despliegue el componente enidocWS pueda dar servicio a varios sistemas clientes autorizados.

En ambos modos de configuración, es suficiente con separar por comas ',' los valores en los campos usuarioWS y passwordWS. Deben indicarse los mismos números de valores en estos campos y en el orden adecuado, es imprescindible que no haya usuarios repetidos:

Por ejemplo:

```
usuarioWS = usuarioA,usuarioB
passwordWS = passA,passB
```

Se corresponderá con el sistemaA (usuarioA, passA) y con el sistemaB (usuarioB, passB).

NOTA2: Si establece como “true” el parámetro de configuración “obviarCaracteresCSV”, la librería no hará ninguna verificación preliminar sobre la longitud y estructura de los códigos CSV consultados.

3.1.4 Implementación de la clase invocada por el WS

Es necesario que esta clase contenga un método, que necesariamente debe ser **estático**, con la siguiente especificación, según las operaciones del servicio “enidocWS” a ofrecer:

A) public static EniDoc nombreMetodo (Petición petición)
B) public static DatosEni nombreMetodoIn (InputStream in, Petición petición)

3.1.4.1 A) Acceso al documento ENI

Sobre este método recae la responsabilidad de obtener el documento ENI del sistema/repositorio cliente para devolver el objeto esperado.

A continuación se muestra un ejemplo de implementación de este método con el objetivo de guiar a los desarrolladores en la secuencia de pasos a seguir para obtener el documento ENI.

```

public static EniDoc nombreMetodo (Petición petición) {

    // Creamos el objeto EniDoc que debemos rellenar
    EniDoc eniDoc = new EniDoc();

    // TODO Tratamiento opcional de los datos de la Petición para definir e implantar
    // criterios y políticas para la gestión de las peticiones de acceso al documento
    String idConsultado = petición.getIdConsultado();
    String ip = petición.getIp();
    String perfil = petición.getPerfil();
    String motivo = petición.getMotivo();
    CertificadoEP certEP = petición.getCertEP();
    CertificadoPF certPF = petición.getCertPF();
    CertificadoPJ certPJ = petición.getCertPJ();

    // TODO Obtener documento de BBDD mediante el csv consultado
    byte[] documentoBytes = obtenerDocumentoENIPorId(csv);

    // Si el documento vinculado al CSV no existe en el repositorio
    if (documentoBytes == null) {
        eniDoc = new EniDoc();
        eniDoc.setCodigoRespuesta(ErroresEni.STATUS_NOT_FOUND_RESOURCE);
    }
    // Si el documento vinculado al CSV existe en el repositorio
    else {

        // TODO Rellenar el objeto EniDoc
        // 1.- Array de bytes que contiene el documento electrónico ENI.
        // Obligatorio.
        eniDoc.setDocumentoENI(documentoBytes);

        // 2.- Array de bytes que contiene el documento original con el cajetín
        // de firma incorporado. Opcional.
        byte[] informeFirma = null;
    }
}

```

```
eniDoc.setInformeFirma(informeFirma);

// 3.- Texto libre de respuesta para ser mostrado al usuario final. Si
// este campo se complementa se dispondrá al usuario única y
// exclusivamente dicho texto, ignorando el resto de la información.
// Opcional.
String textoLibre = null;
eniDoc.setTextoLibre(textoLibre);

// 4.- Texto libre adicional de respuesta para ser mostrado al usuario
// final, junto con los demás resultados del proceso. Opcional.
String textoAdicional = null;
eniDoc.setTextoAdicional(textoAdicional);

// 5.- En caso de firma electrónica basada en certificado, componente o
// herramienta utilizada para su generación. Opcional.
String generadorFirma = null;
eniDoc.setGeneradorFirma(generadorFirma);

// 6.- Información de utilidad para las firmas electrónicas que no
// incorporen sello de tiempo. Opcional.
Date marcaTiempo = null;
eniDoc.setMarcaTiempo(marcaTiempo);

// 7.- Campo que recoge, en el caso de firma electrónica basada en
// certificado, si el sello de tiempo es requerido. Este dato resultará
// de utilidad en el caso de que la firma
// electrónica no disponga de sello de tiempo, o el mismo hubiera
// caducado, en el proceso de su validación. Opcional.
boolean selloTiempoRequerido = true;
eniDoc.setSelloTiempoRequerido(selloTiempoRequerido);

// 8.- Campo que recoge expresamente, en el caso de firma electrónica
// basada en certificado, si en su momento se validaron los certificados
// electrónicos utilizados para la firma. Tomará valores 'true' o
// 'false'. Este dato resultará de utilidad en el caso de que la firma
// electrónica no disponga de sello de tiempo, en el proceso de su validación.
// Opcional.
boolean validacionCertificadosFirma = true;
eniDoc.setValidacionCertificadosFirma(validacionCertificadosFirma);

// 9.- Campo con el identificador normalizado del documento que, en su
// caso, hubiera sustituido al documento al cual se intenta acceder.
// Opcional.
String identificadorDocumentoSustituto = null;
eniDoc.setIdentificadorDocumentoSustituto(identificadorDocumentoSustituto);

// 10.- Campo que indica si la firma electrónica está almacenada en un
// depósito seguro. Opcional
boolean depositoSeguro = false;
eniDoc.setDepositoSeguro(depositoSeguro);

// 11.- Campo que indica si se desea habilitar que la persona usuaria de
// la herramienta pueda descargar el correspondiente fichero de firma
// electrónica y el fichero en formato ENI. Opcional
boolean descargarFirmaydocumentoENI = true;
eniDoc.setDescargarFirmaydocumentoENI(descargarFirmaydocumentoENI);

// 12.- Se considera este campo, en el caso de que el tipo de firma no
// esté contemplado en la adaptación al NTI. Opcional
// - TF97 - CMS.
// - TF98 - XMLDSig.
// - TF99 - Indefinido.
TipoFirmaNoNormalizada tipoFirmaNoNormalizada = null;
eniDoc.setTipoFirmaNoNormalizada(tipoFirmaNoNormalizada);

// 13.- Campo adicional que se utilizará para posibles necesidades
// futuras, en el caso que se requiera alguna información adicional.
// Opcional.
String campoAdicional = null;
eniDoc.setCampoAdicional(campoAdicional);
```

```

    }

    // Devolver el objeto EniDoc relleno.
    return eniDoc;
}

```

3.1.4.2 B) Inserción del Documento ENI

Sobre este método recae la responsabilidad de generar el CSV y el identificador normalizado del documento a almacenar en el sistema/repositorio cliente, así como la actualización de los metadatos mínimos obligatorios.


A continuación se muestra un ejemplo de implementación de este método con el objetivo de guiar a los desarrolladores en la secuencia de pasos a seguir para obtener el objeto DatosEni.

```

public static DatosEni nombreMetodo (InputStream in, Peticion peticion) {
    tratamientoPeticion(peticion);
    DatosEni datosEni = null;
    try {
        // Creamos el objeto DatosEni que debemos rellenar
        datosEni = new DatosEni();
        // SISTEMA, se corresponde con los 5 caracteres que identifican el sistema
        // repositorio
        String csv = UtilidadesEni.generarCSV("SISTEMA",
            RandomStringUtils.randomNumeric(6)).getValor();
        // Indicar el identificador
        String identificador = "IDENTIFICADOR";
        // Se crea un DocumentoEni del inputStream recibido
        DocumentoEni conversorENI = new DocumentoEni(in);
        // Crear un DocumentoEni temporal para setear los valores
        DocumentoEni docEniAux = new DocumentoEni();
        docEniAux.setContenido(conversorENI.getContenido());
        docEniAux.setFirmas(conversorENI.getFirmas());
        docEniAux.setOrganos(conversorENI.getOrganos());
        docEniAux.setFechaCaptura(conversorENI.getFechaCaptura());
        docEniAux.setNombreFormato(conversorENI.getNombreFormato());
        docEniAux.setTipoDocumental(conversorENI.getTipoDocumental());
        docEniAux.setEstadoElaboracion(conversorENI.getEstadoElaboracion());
        docEniAux.establecerIdentificador(identificador);
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        docEniAux.storeToXML(out);
        byte[] bytesEni = out.toByteArray();
        // Método encargado grabar el documento eni
        escribirArchivo(bytesEni, identificador);
        datosEni.setCsv(csv);
        String idEspecifico = UtilidadesEni.generaIdentificadorEni(
            ORGANOS, identificador, new Date());
        datosEni.setIdentificadorDocumento(idEspecifico);
    } catch (ConversionException e) {
        datosEni = null;
    }
    return datosEni;
}

```

Debe quedar claro que es responsabilidad del sistema que que recibe la solicitud de custodia decidir si lo acepta, pudiendo en su caso devolver algún error en caso de rechazo.

	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

El sistema que captura y remite el documento electrónico hacia el repositorio podrá establecer valores a los metadatos mínimos obligatorios del documento, si bien el sistema receptor puede aceptar estos valores o sustituirlos por otros. Así, por ejemplo, la asignación definitiva del metadato Identificador recae sobre el repositorio donde el documento se custodiará.

Un ejemplo de rechazar la petición podría ser similar a:

```

/**
 * Tratar petición.
 * @param petición the petición
 * @return the datos eni
 */
private static DatosEni tratarPetición(Petición petición) {
    DatosEni datosEni = null;
    if (petición == null) {
        datosEni = new DatosEni();
        datosEni.setCodigoRespuesta(ErroresEni.STATUS_REFUSE_REQUEST);
        datosEni.setMensajeRespuesta(ErroresEni.getString(datosEni
            .getCodigoRespuesta()));
    }
    return datosEni;
}

```

3.2 Despliegue de la aplicación “enidocWS”

En caso de optar por esta alternativa, se pone a disposición una aplicación intermedia entre la herramienta destino y el sistema/repositorio donde se deberá configurar el fichero de propiedades e implementar la clase contenedora del método/s a invocar desde el servicio web, de forma análoga a la explicada respectivamente en los puntos 3.1.3 y 3.1.4 del presente documento.

Para acceder al modelo de datos del sistema/repositorio, es posible que sea necesario hacer uso de jdbc. Para ello habría que definir el DataSource correspondiente en el fichero “context.xml”.

Una vez configurada la aplicación e implementado el código necesario, se debe generar el war de la aplicación y desplegarlo en un servidor Tomcat con JDK 1.6.

3.3 Cumpliendo los xml de intercambio de información. Petición y Respuesta


En caso de optar por esta alternativa, se deben realizar una implementación propia conforme a las estructuras XML de intercambio, tanto para la de petición como la de respuesta, siendo ambas equivalentes a las que se muestra a modo de ejemplo:

Petición:

```


<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <petición perfil="" motivo="" ip="" idConsultado="">
    <certPF tipoCertificado="" nombreApellidosResponsable="" nif=""
    entidadEmisora=""/>
  </petición>

```

 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

Respuesta:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <eniDoc>
    <codigoRespuesta/>
    <mensajeRespuesta/>
    <documentoENI/>
    <despositoSeguro/>
    <descargarFirmaydocumentoENI/>
    <selloTiempoRequerido/>
    <validacionCertificadosFirma/>
    ...
  </eniDoc>
```


 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	<p>Consejería de Hacienda y Administración Pública</p>	<p>eniDoc Web Services</p> <p>Manual de Integración</p>
---	--	---

4 IMPLEMENTACIÓN DE LAS OPERACIONES ENIDOCWS COMO CLIENTE CONSUMIDOR

NOTA: El desarrollo del cliente deberá ser implementado por los sistemas o herramientas de tramitación que deban acceder a documentos electrónicos ENI o bien solicitar la custodia de los nuevos documentos que capturan.


Para facilitar la integración cliente de enidocWS, se proporciona una librería enidocClient mediante la cual podemos consumir los servicios de intercambio de documentos electrónicos, simplemente añadiendo las siguiente líneas de código:

```
IEniDocClient eniDocClient = new EniDocClientImpl();
EniDoc eniDoc = eniDocClient.obtenerDocumentoENI(peticion, usuario, pass, url);
```

Donde peticion será el objeto descrito para enidocWS y el resto de parámetros, usuario, pass y url, son los correspondientes parámetros de conexión al servicio en cuestión.

```
IEniDocClient eniDocClient = new EniDocClientImpl();
DatosEni datosENI = eniDocClient.insertarDocumentoENI(peticion, insercion, usuario, pass, url);
```

Donde peticion será el objeto descrito para enidocWS, insercion un objeto tipo File que contiene el xml del documento electrónico y el resto de parámetros, usuario, pass y url, son los correspondientes parámetros de conexión al servicio en cuestión.

	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

5 ANEXO I. GENERACIÓN DE CÓDIGO SEGURO DE VERIFICACIÓN

Este anexo presenta, a modo de ejemplo, una propuesta para calcular los 25 caracteres últimos para formar un código seguro de verificación.

- a. Los 25 caracteres serán alfanuméricos, de entre aquellos que forman parte del alfabeto ABCDEFGHJKLMNPQRSTUVWXYZ23456789.
- b. Se generarán 60 bytes aleatorios satisfaciendo el IETF RFC 1750, utilizando para ello la clase Java "SecureRandom" o cualquier otro sistema equivalente.
- c. Se calcula el hash SHA1 de estos bytes, con un resultado final de 20 bytes.
- d. A los 20 bytes calculados en el paso "c" se les unirán los situados en las posiciones 2, 17, 19, 21 y 57 de los calculados en el paso "b".
- e. Para cada uno de 25 bytes disponibles en el paso "d":
 1. Se tomarán los 5 bits menos significativos, con lo que se tendrá un valor numérico decimal entre 0 y 31.
 2. Del alfabeto expuesto en el paso "a", se selecciona el carácter cuya posición coincida con la marcada por el valor obtenido en el punto "e.1" ("0" es la primera posición).
 3. El carácter obtenido en el punto "e.2" se concatena a la derecha del CSV en construcción.
- f. Una vez generado el CSV, el sistema lo vinculará al documento, asegurando en todo caso su unicidad y que por tanto no colisiona con uno ya existente, en cuyo caso se deberá proceder a generar uno nuevo hasta que la colisión no se produzca.

Cada Consejería, entidad, sistema/repositorio, puede utilizar otros medios para la generación de los códigos, atendiendo a sus prácticas, políticas y necesidades en materia de gestión documental.

6 ANEXO II. UTILIZACIÓN DE OTRO FRAMEWORK REST. RESTEASY

Por necesidades y/o limitaciones de la arquitectura sobre la que esté montado/desplegado el sistema/repositorio es posible que no sea compatible con el framework de rest jersey, que es el usado en la implementación de estas librerías, es por tanto, que se indican unas anotaciones que pueden ser de utilidad para usar otro framework, manteniendo la compatibilidad, resteasy.

- 1.- Dependencias a añadir en el *pom.xml*:

```
<dependency>
  <groupId>es.juntadeandalucia.nti</groupId>
  <artifactId>enidocObject</artifactId>
  <version>1.0.0</version>
</dependency>
<dependency>
  <groupId>org.jboss.resteasy</groupId>
  <artifactId>resteasy-client</artifactId>
  <version>3.0.12.Final</version>
</dependency>
<dependency>
  <groupId>org.jboss.resteasy</groupId>
  <artifactId>resteasy-jettison-provider</artifactId>
  <version>3.0.12.Final</version>
</dependency>
<dependency>
  <groupId>org.apache.httpcomponents</groupId>
  <artifactId>httpcore</artifactId>
  <version>4.4</version>
</dependency>
```

- 2.- Ejemplo de cliente:

```
...
// 1) Se crea la credencial
final CredentialsProvider credsProvider = new BasicCredentialsProvider();
credsProvider.setCredentials(new AuthScope(ip, puertoInt), new UsernamePasswordCredentials(user, pass));
// 2) Se crea la caché necesaria para la autenticación preemptively y se establecen los datos de conexión
final AuthCache authCache = new BasicAuthCache();
authCache.put(new HttpHost(ip, puertoInt, protocolo), new BasicScheme());
// 3) Se añade la caché al contexto
final BasicHttpContext localContext = new BasicHttpContext();
localContext.setAttribute(HttpClientContext.AUTH_CACHE, authCache);
// 4) Se crea el cliente ejecutor con las credenciales y el contexto
final HttpClient httpClient = HttpClientBuilder.create().setDefaultCredentialsProvider(credsProvider)
.build();
ApacheHttpClient4Engine engine = new ApacheHttpClient4Engine(httpClient, localContext);
ResteasyClient client = new ResteasyClientBuilder().httpEngine(engine).build();
// 5) Se realiza la llamada al enidocWS junto con el objeto petición
ResteasyWebTarget target = client.target(url).path("verificarFirma");
try{
    eniDoc = target.queryParam("peticion", peticion.toString()).request().get(EniDoc.class);
} catch (Exception e){
    Response response = target.queryParam("peticion", peticion.toString()).request().get();
    eniDoc = parseErrorEnidoc(response.getStatus());
}
...
```

7 ANEXO III. COMPROBACIÓN DE LA INTEGRACIÓN CON ENIDOCWS

7.1 Postman: Cliente REST para Chrome

Para verificar que se ha integrado correctamente la librería con la aplicación deseada y poder comprobar posibles errores o problemas se plantea la alternativa del uso de un cliente REST mediante la instalación de una aplicación complemento del navegador Chrome: *Postman*.

1. Instalar el complemento de Chrome:
<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop>
2. Configurar los parámetros de autenticación en la opción "Authorization > Basic Authorization". Especificar el usuario y la clave de conexión al servicio ENIDOCWS facilitados.

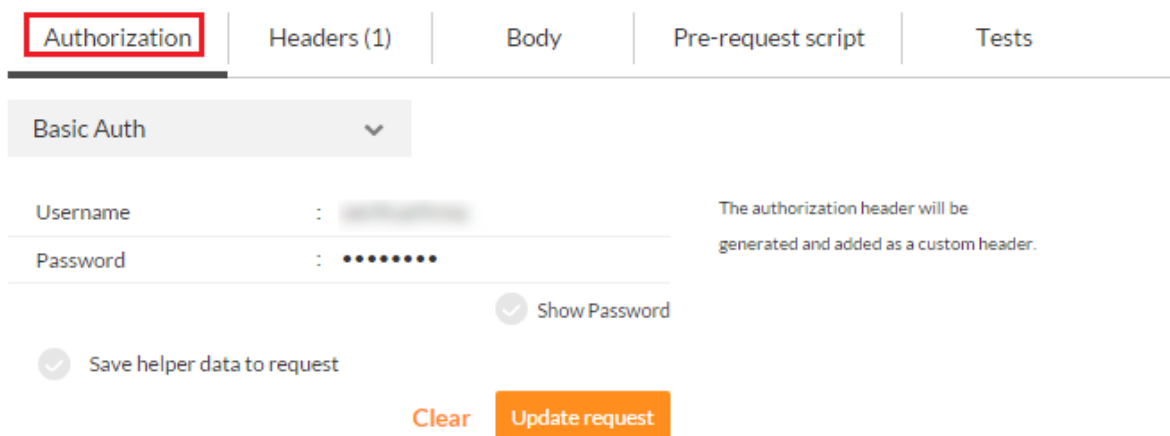


Ilustración 7. Pantalla del complemento de Chrome

3. Especificar el usuario y la clave de conexión al servicio ENIDOCWS facilitados.

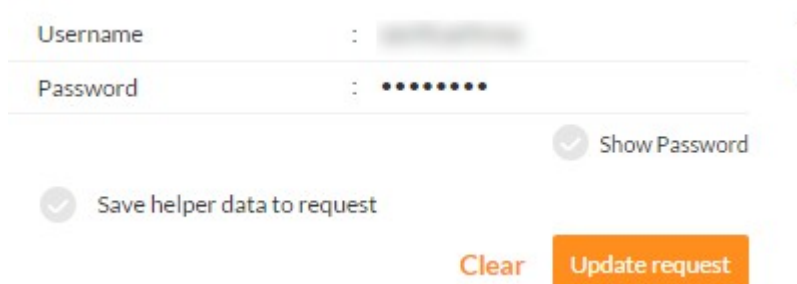


Ilustración 8. Formulario de autenticación

4. En el campo *Method* junto a la URL:

a. *verificarFirma*:

Indicar valor **GET**. En el campo *URL* se pondrá la siguiente:

`http://RUTAAPLICACION/rest/eni/verificarFirma?peticion=<?xml version="1.0" encoding="UTF-8" standalone="yes"?><peticion perfil="1" motivo="MOTIVO" ip="IP" idConsultado="CSV"><certPF tipoCertificado="FNMT PF" nombreApellidosResponsable="NOMBRE APELLIDO1 APELLIDO2" nif="NIF" entidadEmisora="FNMT"/></peticion>`

b. *insertarDocumentoEni*:

Indicar valor **POST**. En el campo URL se pondrá la siguiente,

<http://RUTAAPLICACION/rest/eni/insertarDocumentoEni>

En la pestaña *Body* indicar *form-data* indicar "insercion" como key del parámetro de tipo *File* e indicar el fichero:

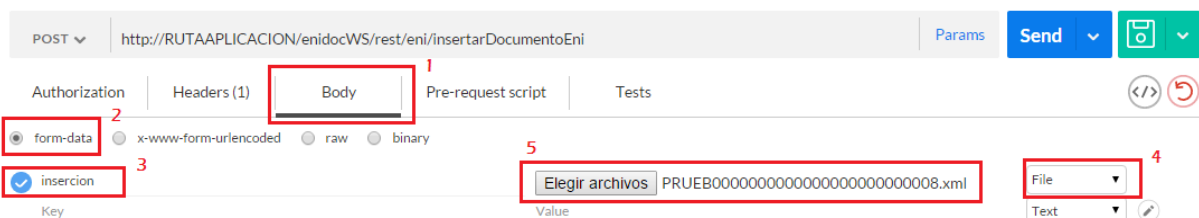


Ilustración 9. Especificar los parámetros de inserción

De la misma forma indicar el xml del objeto petición en un atributo denominado petición:

XML petición=`<?xml version="1.0" encoding="UTF-8" standalone="yes"?><peticion perfil="1" motivo="MOTIVO" ip="IP" idConsultado="CSV"><certPF tipoCertificado="FNMT PF" nombreApellidosResponsable="NOMBRE APELLIDO1 APELLIDO2" nif="NIF" entidadEmisora="FNMT"/></peticion>`

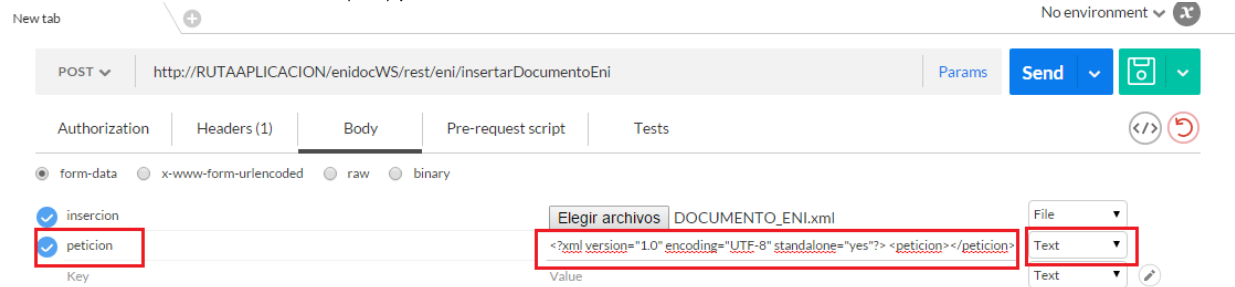
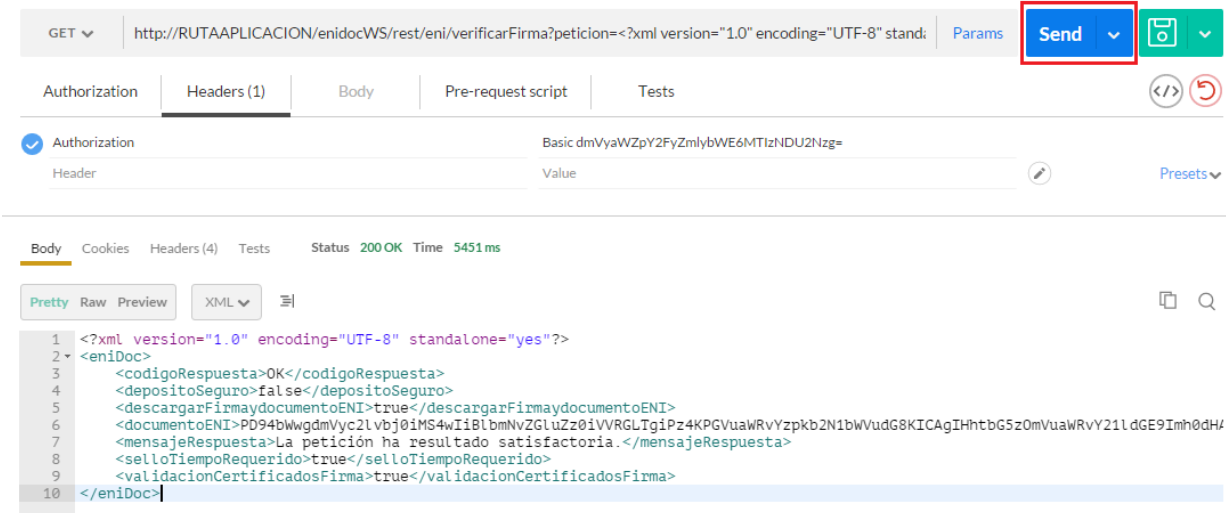


Ilustración 10. Especificar los parámetros de inserción

NOTA: En ambos casos hay que reemplazar previamente las partes resaltadas en negrita y cursiva con la información correspondiente al sistema/repositorio.

5. Una vez establecido los valores indicados, se procede a pulsar el botón *SEND*.
 - 1) Si el resultado es correcto, se obtiene algo similar a lo siguiente:

a. *verificarFirma*:



GET http://RUTAAPLICACION/enidocWS/rest/eni/verificarFirma?peticion=<?xml version="1.0" encoding="UTF-8" stand: Params Send

Authorization Headers (1) Body Pre-request script Tests

Authorization Basic dmVyaWZpY2FyZmlybWE6MTIzNDU2Nzg=

Header Value Presets

Body Cookies Headers (4) Tests Status 200 OK Time 5451 ms

Pretty Raw Preview XML

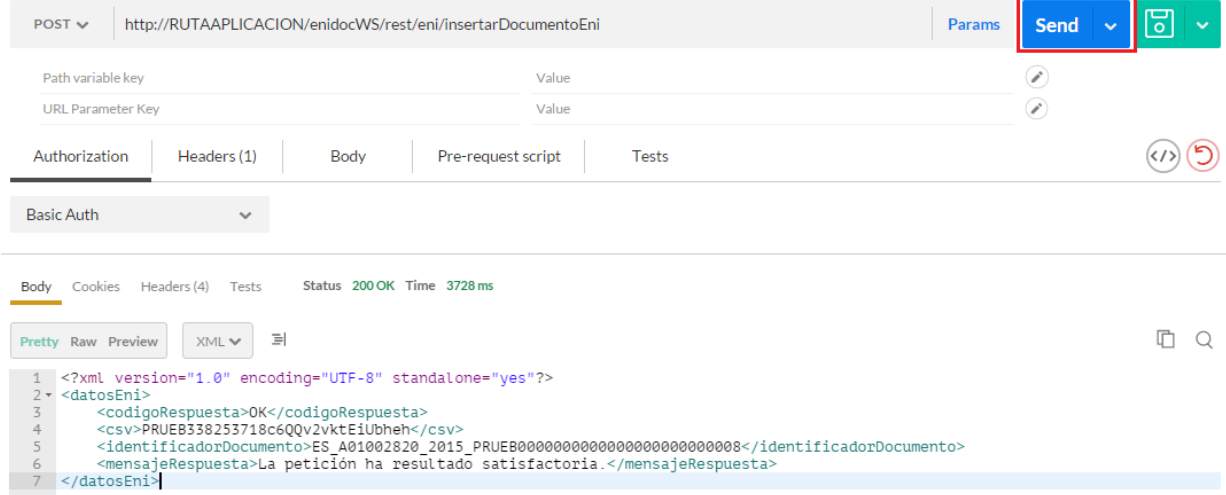
```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <eniDoc>
3   <codigoRespuesta>OK</codigoRespuesta>
4   <depositoSeguro>>false</depositoSeguro>
5   <descargarFirmaydocumentoENI>true</descargarFirmaydocumentoENI>
6   <documentoENI>PD94bWwgdmVyc2lvcj0iMS4wIi8lbnNvZGlucz0iVVRGLTgiPz4KPGVuaWRvYzpkb2N1bWVudG8KICAgIHhtbG5zOmVuaWRvY211dGE9Imh0dH
7   <mensajeRespuesta>La petición ha resultado satisfactoria.</mensajeRespuesta>
8   <selloTiempoRequerido>true</selloTiempoRequerido>
9   <validacionCertificadosFirma>true</validacionCertificadosFirma>
10 </eniDoc>

```

Ilustración 11. Respuesta correcta del servicio verificarFirma

b. *insertarDocumentoEni*:



POST http://RUTAAPLICACION/enidocWS/rest/eni/insertarDocumentoEni Params Send

Path variable key Value

URL Parameter Key Value

Authorization Headers (1) Body Pre-request script Tests

Basic Auth

Body Cookies Headers (4) Tests Status 200 OK Time 3728 ms

Pretty Raw Preview XML


```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <datosEni>
3   <codigoRespuesta>OK</codigoRespuesta>
4   <csv>PRUEB338253718c6QQv2vktEiUbbhe</csv>
5   <identificadorDocumento>ES_A01002820_2015_PRUEB00000000000000000000000008</identificadorDocumento>
6   <mensajeRespuesta>La petición ha resultado satisfactoria.</mensajeRespuesta>
7 </datosEni>

```

Ilustración 12. Respuesta correcta del servicio insertarDocumentoEni

- 1) Si por el contrario se produjera un error, se obtendrá información más detallada que permitirá identificarlo.


 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

8 ANEXO IV. VALIDACIÓN DOCUMENTO ENI

Para validar un documento electrónico (ENI) se recomienda hacer uso de la funcionalidad de validación que dispone la Herramienta Centralizada de Verificación que pone a disposición la Junta de Andalucía. A continuación se indican las URLs de acceso a los entornos de pruebas y producción de esta herramienta:

Entorno de pruebas: https://cancanaprun1.chap.junta-andalucia.es/verificarFirma_des/


Entorno de producción: <https://ws050.juntadeandalucia.es/verificarFirma>

	Consejería de Hacienda y Administración Pública	eniDoc Web Services Manual de Integración
---	--	--

9 ANEXO V. TIPIFICACIÓN DEL CÓDIGO DE RESPUESTA

Los códigos/mensajes de respuesta tipificados para el protocolo eniWS se detallan a continuación:

Código respuesta	Descripción
Respuesta Correcta	
OK	La petición ha resultado satisfactoria.
Errores de autenticación	
UNAUTHORIZED_NOT_CRED	No incluido el usuario y/o la clave en las credenciales de autorización.
UNAUTHORIZED_INVALID_CRED	Credenciales de autorización (usuario y/o clave) incorrectas.
UNAUTHORIZED_INVALID_PROT	La petición al servicio de integración no cumple con el protocolo de autenticación especificado.
UNAUTHORIZED_NOT_PROT	La petición al servicio de integración no contiene el protocolo de autenticación.
Errores con los datos proporcionados	
BAD_REQUEST	Petición incorrecta.
REFUSE_REQUEST	Petición rechazada.
BAD_REQUEST_NOT_ID	El parámetro "idConsultado" no ha sido proporcionado.
BAD_REQUEST_INVALID_ID	El identificador normalizado del documento consultado contiene sintaxis errónea.
BAD_REQUEST_NO_MATCHING	El identificador normalizado del documento consultado no coincide con el identificador del documento recuperado.
Errores con los datos obtenidos	
NOT_FOUND_RESOURCE	Recurso solicitado no encontrado en base a los parámetros proporcionados.
NOT_FOUND_ENIDOC	Error al obtener el Documento ENI.

 <p>JUNTA DE ANDALUCÍA CONSEJERÍA DE HACIENDA Y ADMINISTRACIÓN PÚBLICA</p>	<p>Consejería de Hacienda y Administración Pública</p>	<p>eniDoc Web Services</p> <p>Manual de Integración</p>
---	--	---

FOUND_SUBSTITUTE_RESOURCE	El documento solicitado se encuentra disponible con otro identificador.
FOUND_RESTRICTED_RESOURCE	El documento solicitado ha sido restringido por el sistema/repositorio.
FOUND_ARCHIVED_RESOURCE	El documento solicitado ha sido archivado por el sistema/repositorio.
Errores del servidor	
INTERNAL_ERROR	Error técnico de acceso a datos.
INTERNAL_SERVER_ERROR	Error del servidor.
INTERNAL_SERVICE_UNAVAILABLE	Conexión rechazada. Servidor no disponible temporalmente.

10 REFERENCIAS

Objeto	Referencia
NTI de Documento Electrónico	http://www.boe.es/diario_boe/txt.php?id=BOE-A-2011-13169
JVM	http://en.wikipedia.org/wiki/Java_virtual_machine
REST	http://en.wikipedia.org/wiki/REST
Jersey	https://jersey.java.net/
Complemento Firefox	https://addons.mozilla.org/es/firefox/addon/restclient/