



WS EdiwinWS EDICOM
Guía de usuario

Título del documento:	WS EdiwinWS EDICOM
Nombre del fichero:	ES WebService EdiwinWS Guía de usuario.odt
Versión:	1.1
Estado:	REVISADO
Fecha:	08/10/2013
Autor:	Oscar Albert Arcas

Revisión, Aprobación

Revisado por:	Oscar Albert Arcas	Fecha: 08/10/2013
Aprobado por:	Depto. Investigación y Desarrollo	Fecha: 08/10/2013

Historial de cambios

Versión	Fecha	Descripción de la acción	Páginas
1.0	21/01/2011	Versión inicial EdiwinWS	Todas

Índice de contenido

1.Introducción.....	5
1.1.Acceso al servicio.....	5
1.2.Funciones.....	5
2.Funciones.....	7
2.1.Registro de sesión.....	7
Llamada:.....	7
Parámetros:.....	7
Respuesta:.....	7
Ejemplos.....	7
2.2.Finalización de la sesión.....	9
Llamada:.....	9
Parámetros:.....	9
Respuesta:.....	9
Ejemplos.....	9
2.3.Obtención de documentos.....	11
Llamada:.....	11
Parámetros:.....	11
Respuesta:.....	11
Ejemplos.....	11
2.4.Obtención de documentos en Base64.....	14
Llamada:.....	14
Parámetros:.....	14
Respuesta:.....	14
Ejemplos.....	14
2.5.Publicación de documentos.....	16
Llamada:.....	16
Parámetros:.....	16
Respuesta:.....	16
Ejemplos.....	17
2.6.Publicación de documentos en Base64.....	18
Llamada:.....	18
Parámetros:.....	19
Respuesta:.....	19
Ejemplos.....	19
2.7.Ejecutar complemento.....	21
Llamada:.....	21
Parámetros:.....	21
Respuesta:.....	21
Ejemplos.....	21
2.8.Exportar documentos.....	22
Llamada:.....	23
Parámetros:.....	23
Respuesta:.....	23
Ejemplos.....	24

2.9.Exportar documentos en Base64.....	25
Llamada:.....	26
Parámetros:.....	26
Respuesta:.....	26
Ejemplos.....	26
2.10.Cambiar documentos de situación.....	28
Llamada:.....	28
Parámetros:.....	28
Respuesta:.....	29
Ejemplos.....	29
2.11.Cambio de los campos extendidos de los documentos.....	31
Llamada:.....	31
Parámetros:.....	31
Respuesta:.....	31
Ejemplos.....	31
2.12.Obtención de datos estadísticos.....	33
Llamada:.....	33
Parámetros:.....	33
Respuesta:.....	34
Ejemplos.....	34
2.13.Obtención de interfaces.....	36
Llamada:.....	36
Parámetros:.....	36
Respuesta:.....	36
Ejemplos.....	37
2.14.Obtención de interlocutores.....	38
Llamada:.....	38
Parámetros:.....	38
Respuesta:.....	38
Ejemplos.....	39
3.Lista de Códigos de Error.....	41
3.1.Introducción.....	41
Ejemplo de una respuesta de error.....	42
4.Anexo: WSDL.....	43
4.1.WSDL WEBSERVICE EdiwinWS.....	43

1. INTRODUCCIÓN

EdiwinWS es una plataforma de acceso a Ediwin mediante Webservices.

A través de esta plataforma, los clientes tienen un acceso externo a Ediwin, mediante el cual pueden realizar operaciones entre su sistema de gestión y el servidor Ediwin. De esta forma, pueden generar o extraer información de Ediwin e integrarla conforme la necesiten.

1.1. ACCESO AL SERVICIO

La URL de acceso al webservice de Ediwin es la siguiente:

<https://web.sedeb2b.com/EdiwinWS/services/EdiwinWS?wsdl>

1.2. FUNCIONES

El WSDL define las funciones que se muestran a continuación:

función	Descripción	ver.
registerSession	Registro de sesión. Permite autenticarse en el webservice para poder acceder al resto de peticiones del webservice.	1.0
endSessionendSession	Finalización de la sesión. Permite finalizar la sesión activa de Ediwin	1.0
getDocument	Obtención de documentos. Permite obtener documentos e información de los documentos de Ediwin, tanto de la estación como del volumen.	1.0
getDocumentB64	Obtención de documentos en Base 64. Permite obtener documentos e información de los documentos de Ediwin, tanto de la estación como del volumen. El resultado será un fichero en base 64 para sistemas que no permiten utilizar archivos adjuntos.	1.0
publishDocument	Publicación de documentos. Permite publicar documentos en la estación Ediwin.	1.0
publishDocumentB64	Publicación de documentos en Base 64. Permite publicar documentos en la estación Ediwin utilizando ficheros en base 64 para sistemas .que no permiten utilizar ficheros adjuntos.	1.0
complementFunction	Ejecutar complemento. Permite ejecutar una función de complemento de Ediwin.	1.0
exportDocuments	Exportar documentos. Permite obtener los documentos y la información asociada a los documentos de Ediwin.	1.0
exportDocumentsB64	Exportar documentos en Base 64. Permite obtener los documentos y la información asociada a los documentos de Ediwin. El resultado será un fichero en base 64 para sistemas que no permiten utilizar archivos adjuntos.	1.0
changeDocSituation	Cambiar documentos de situación. Permite al usuario cambiar la situación de los documentos de la estación Ediwin.	1.0
changeExtendedFieldValue	Cambio de los campos extendidos de los documentos. Permite al usuario cambiar los valores de los campos extendidos. de los documentos de la estación Ediwin.	1.0
getStatistics	Obtención de datos estadísticos. Permite al usuario acceder a las estadísticas de la estación Ediwin.	1.0

<u>getInterfaces</u>	Obtención de interfaces. Permite al usuario acceder a las guías configuradas en el dominio de la estación Ediwin.	1.0
<u>getInterlocutors</u>	Obtención de interlocutores. Permite al usuario acceder a los interlocutores configurados en el dominio de la estación Ediwin.	1.0

2. FUNCIONES

El Webservice define una serie de funciones que se pueden ejecutar contra el servidor Ediwin. Estas funciones están definidas por la especificación SOAP 1.2, en la que se utiliza “SOAP Attachments” para la transmisión de ficheros a través del webservice. Además, el webservice implementa sesión para poder utilizar la propia sesión del Ediwin.

El WSDL define las funciones que se muestran a continuación:

2.1. REGISTRO DE SESIÓN.

registerSession

Permite registrarse en la sesión del Ediwin, y así poder empezar a operar. Puesto que la sesión se mantiene en Ediwin, el webservice mantiene la sesión. De esta forma, sólo es necesaria la autenticación del cliente una vez mientras el cliente webservice se mantenga activo.

Esta función es obligatoria que se ejecute al inicio de las operaciones.

LLAMADA:

```
public void registerSession(String user, String password, String domain, int clientType);
```

PARÁMETROS:

- **user:** Usuario que va realizar la conexión
- **password:** Contraseña del usuario que va realizar la conexión
- **domain:** Dominio del usuario que realiza la conexión.
- **group:** Grupo del dominio al que se realiza la conexión. Es un parámetro opcional
- **clientType:** Tipo de cliente.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que la función se ejecute correctamente, devolverá el identificador de la sesión del webservice que está accediendo a Ediwin. Este identificador se mantiene durante toda la sesión del webservice para poder seguir interactuando con Ediwin.
- **Respuesta errónea:** En caso de que haya algún error en el proceso de autenticación, saltará una excepción que contendrá un mensaje de texto con el error producido.

EJEMPLOS

Código del cliente:

```
public void registerSession(String user, String pass, String domain, String group, int clientType){
    try {
        service.registerSession(user, pass, domain, group, clientType);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException Re) {
        Re.printStackTrace();
        System.out.println("RemoteExcepcion: " + Re.toString());
    } catch (Exception e) {
```

```

        e.printStackTrace();
        System.out.println("Exception: " + e.toString());
    }
}

```

Petición SOAP:

```

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:registerSession
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <user xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">XXXXXX</user>
      <password xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">XXXXXX</password>
      <domain xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">XXXXXX</domain>
      <group xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">XXXXXX</group>
      <clientType xsi:type="xsd:int">1</clientType>
    </urn:registerSession>
  </soapenv:Body>
</soapenv:Envelope>

```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de una **autenticación satisfactoria**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:registerSessionResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS"/>
  </soapenv:Body>
</soapenv:Envelope>

```

El siguiente ejemplo es una respuesta de **error**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --> Cod: 1; Text: Los datos de autenticación
introducidos son incorrectos.</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">1</cod>
          <text xsi:type="xsd:string">Los datos de autenticación
introducidos son incorrectos.</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```


Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.2. FINALIZACIÓN DE LA SESIÓN.

endSession

Permite finalizar la sesión del Ediwin.

Esta función es obligatoria que se ejecute al final de las operaciones.

LLAMADA:

```
public void endSession();
```

PARÁMETROS:

No tiene. La propia sesión iniciada con la petición registerSession autentifica la sesión que tiene que finalizar.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que la función se ejecute correctamente, no devolverá nada. La propia respuesta sin error indica que la sesión que se mantiene en la sesión se ha finalizado correctamente.
- **Respuesta errónea:** En caso de que haya algún error en el proceso de finalización de la sesión, saltará una excepción que contendrá un mensaje de texto con el error producido.

EJEMPLOS

Código del cliente:

```
public void endSession() {
    try {
        service.endSession();
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEException: " + EWSe.getText());
    } catch (RemoteException Re) {
        Re.printStackTrace();
        System.out.println("RemoteException: " + Re.toString());
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("Exception: " + e.toString());
    }
}
```

Petición SOAP:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:registerSession
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <user xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">XXXXXX</user>
      <password xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">XXXXXX</password>
```

```

        <domain xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">XXXXXX</domain>
        <group xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">XXXXXX</group>
        <clientType xsi:type="xsd:int">1</clientType>
    </urn:registerSession>
</soapenv:Body>
</soapenv:Envelope>

```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de una **finalización de la sesión satisfactoria**.

```

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:endSession
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    </soapenv:Body>
</soapenv:Envelope>

```

El siguiente ejemplo es una respuesta de **error**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>soapenv:Server.userException</faultcode>
            <faultstring>Error --> Cod: -1; Text: Error desconocido.</faultstring>
            <detail>
                <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
                    <cod xsi:type="xsd:int">-1</cod>
                    <text xsi:type="xsd:string">Error desconocido.</text>
                </ns1:fault>
                <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
            </detail>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>

```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.3. OBTENCIÓN DE DOCUMENTOS

getDocuments

Permite al usuario autenticado a través del webservice obtener datos de los documentos que tiene disponibles en la estación Ediwin utilizando SoapAttachments.

Datos disponibles a través de este webservice:

- Información extendida del documento (Campos de la tabla de documentos separador por limitadores)
- Contenido del documento original en su formato de origen (EDIFACT, XML, etc.)
- Contenido del sobre del documento incluyendo su firma, si la tiene. (Fuente).
- Informe del documento en el formato que tenga especificado el cliente.

LLAMADA:

```
public DataHandler getDocuments (String filter, int type, String
documentTypeList, Integer idVolume);
```

PARÁMETROS:

- **filter:** Filtro SQL para identificar los documentos. Campos para utilizar en los filtros:
 1. REFERENCIA: Campo REFERENCIA
 2. ORIGEN: Campo ENV_ORIGEN
 3. DESTINO: Campo ENV_DESTINO
 4. ESTADO: Campo ESTADOEXT
 5. FECHA CAMBIO DE ESTADO: Campo FECHACAMBIOESTADO
 Ejemplo: REFERENCIA='FACTURA001'
- **type:** Tipo de documento(Info/Doc/Int/Informe)
 1. Información extendida
 2. Documentos
 3. Intercambio
 4. Informes
- **documentTypeList:** Listado de interfaces de documento a utilizar para el caso de que hayan distintos tipos de informes a generar.
- **idVolume:** Identificador del volumen del que se quiere obtener los datos del documento. Por defecto, la estación es el id 0.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya obtenido correctamente el/los documentos, devolverá un fichero adjunto en formato ZIP con los documentos extraídos.
- **Respuesta con errores:** En caso de que haya algún error en el proceso de recuperación, saltará una excepción que contendrá un mensaje de texto con el error producido. Ejemplos de este tipo de excepciones sería que el documento solicitado no exista en la estación.

EJEMPLOS

Código del cliente:

```

public DataHandler getDocuments(String filter, int type, String listado, int
idVolume) {
    DataHandler dato = null;
    try {
        dato = service.getDocuments(filter, type, listado, idVolume);
    } catch (EdiwinWSException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSExcepcion: " + EWSe.getText());
    } catch (RemoteException rE) {
        rE.printStackTrace();
        System.out.println("RemoteExcepcion: " + rE.toString());
    }
    return dato;
}

```

Petición SOAP:

```

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:getDocuments
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
            <filter xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">REFERENCIA='FACTURA001'
</filter>
            <type xsi:type="xsd:int">2</type>
            <documentTypeList xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></documentTypeList>
            <idVolume></idVolume>
        </urn:getDocuments>
    </soapenv:Body>
</soapenv:Envelope>

```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de **obtención de documentos satisfactoria**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <ns1:getDocumentsResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
            <getDocumentsReturn href="cid:CC1D2AC8DFE1FAE69087EBD2B4460370"
xsi:type="ns2:DataHandler" xmlns:ns2="http://xml.apache.org/xml-soap"/>
        </ns1:getDocumentsResponse>
    </soapenv:Body>
</soapenv:Envelope>

```

Objetos adjuntados dentro de la petición:

```

-----_Part_0_1621661368.1381245077840
Content-Type: application/zip
Content-Transfer-Encoding: binary
Content-Id: <CC1D2AC8DFE1FAE69087EBD2B4460370>

PK#####EDI_INVOIC_D_01B_UN_EAN010.EDI_n6#U2ZPaI#h#
) $im0X#{Pvv$-
[...]
#=#zPK#gX##l###PK#####gX##l#####EDI_INVOIC_D_
01B_UN_EAN010.EDI PK#####L###

```

```
-----=_Part_0_1621661368.1381245077840
```

Nota: Los datos que aparecen en este ejemplo no son reales. El código del fichero binario embebido en la petición SOAP ha sido resumido [...]

El siguiente ejemplo es una respuesta de **error**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --> Cod: 7; Text: No se ha exportado ningún
documento. Por favor, revise el filtro de búsqueda de documentos.</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">7</cod>
          <text xsi:type="xsd:string">No se ha exportado ningún documento.
Por favor, revise el filtro de búsqueda de documentos.</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.4. OBTENCIÓN DE DOCUMENTOS EN BASE64

getDocumentsB64

Permite al usuario autenticado a través del webservice obtener datos de los documentos que tiene disponibles en la estación Ediwin sin utilizar SoapAttachments.

Datos disponibles a través de este webservice:

- Información extendida del documento (Campos de la tabla de documentos separador por limitadores)
- Contenido del documento original en su formato de origen (EDIFACT, XML, etc.)
- Contenido del sobre del documento incluyendo su firma, si la tiene. (Fuente).
- Informe del documento en el formato que tenga especificado el cliente.

LLAMADA:

```
public byte[] getDocumentsB64 (String filter, int type, String documentTypeList,
Integer idVolume);
```

PARÁMETROS:

- **filter:** Filtro SQL para identificar los documentos. Campos para utilizar en los filtros:
 1. REFERENCIA: Campo REFERENCIA
 2. ORIGEN: Campo ENV_ORIGEN
 3. DESTINO: Campo ENV_DESTINO
 4. ESTADO: Campo ESTADOEXT
 5. FECHA CAMBIO DE ESTADO: Campo FECHACAMBIOESTADO
 Ejemplo: REFERENCIA='FACTURA001'
- **type:** Tipo de documento(Info/Doc/Int/Informe)
 1. Información extendida
 2. Documentos
 3. Intercambio
 4. Informes
- **documentTypeList:** Listado de interfaces de documento a utilizar para el caso de que hayan distintos tipos de informes a generar.
- **idVolume:** Identificador del volumen del que se quiere obtener los datos del documento. Por defecto, la estación es el id 0.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya obtenido correctamente el/los documentos, devolverá un fichero adjunto en formato ZIP con los documentos extraídos.
- **Respuesta con errores:** En caso de que haya algún error en el proceso de recuperación, saltará una excepción que contendrá un mensaje de texto con el error producido. Ejemplos de este tipo de excepciones sería que el documento solicitado no exista en la estación.

EJEMPLOS

Código del cliente:

```

public byte[] getDocumentsB64(String filter, int type, String listado, int
idVolume) {
    byte[] dato = null;
    try {
        dato = service.getDocumentsB64(filter, type, listado, idVolume);
    } catch (EdiwinWSException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSExcepcion: " + EWSe.getText());
    } catch (RemoteException rE) {
        rE.printStackTrace();
        System.out.println("RemoteExcepcion: " + rE.toString());
    }
    return dato;
}

```

Petición SOAP:

```

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:getDocumentsB64
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
            <filter xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">REFERENCIA='FACTURA001'
</filter>
            <type xsi:type="xsd:int">2</type>
            <documentTypeList xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></documentTypeList>
            <idVolume></idVolume>
        </urn:getDocumentsB64>
    </soapenv:Body>
</soapenv:Envelope>

```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de **obtención de documentos satisfactoria**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <ns1:getDocumentsB64Response
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
            <getDocumentsB64Return xsi:type="soapenc:base64Binary"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">UESDBBQACAAIAC6x9/8AAAA
AAAAAAAAAAAAAeAAAARURJX01OVk9JQ19EXzAxQl9VTl9FQU4wMTAuRURJrZVfbuM2EMavMsiLWlBhSYrU
[...]
wHAABQSwECFAAUAAgACAAusff/xxdn0VgDAABsBwAAHgAAAAAAAAAAAAAAAAAAAAAAAAARURJX01OVk9JQ19
EXzAxQl9VTl9FQU4wMTAuRURJUESFBgAAAAABAAEATAAAAKQDAAAAAA==</getDocumentsB64Return>
        </ns1:getDocumentsB64Response>
    </soapenv:Body>
</soapenv:Envelope>

```

Nota: Los datos que aparecen en este ejemplo no son reales. El código del fichero en base 64 ha sido resumido [...]

El siguiente ejemplo es una respuesta de **error**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

```

```

<soapenv:Body>
  <soapenv:Fault>
    <faultcode>soapenv:Server.userException</faultcode>
    <faultstring>Error --> Cod: 7; Text: No se ha exportado ningún
documento. Por favor, revise el filtro de búsqueda de documentos.</faultstring>
    <detail>
      <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
        <cod xsi:type="xsd:int">7</cod>
        <text xsi:type="xsd:string">No se ha exportado ningún documento.
Por favor, revise el filtro de búsqueda de documentos.</text>
      </ns1:fault>
      <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
    </detail>
  </soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>

```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.5. PUBLICACIÓN DE DOCUMENTOS

publishDocument

Permite al usuario autenticado a través del webservice importar documentos en salida en la estación Ediwin utilizando SoapAttachments.

Los documentos se pasan al webservice mediante un único fichero comprimido en formato ZIP. De esta forma, se pueden publicar en Ediwin todos los documentos que sean necesarios.

LLAMADA:

```
public String publishDocument(DataHandler message, int publishType, String process);
```

PARÁMETROS:

- **message:** Documento en formato zip con los documentos a importar. Se envía en la petición como un SoapAttachment.
- **publishType:** Tipo de publicación:
 1. Publicación de los documentos en entrada.
 2. Publicación de los documentos en salida.
- **process:** Nombre del proceso que se quiere que se ejecute con los ficheros que se suben al servidor. Si se especifica este parámetro, lo que se importará en la estación Ediwin serán los ficheros generados por este proceso, y no los documentos enviados.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya publicado correctamente el documento, devolverá un mensaje de texto con los ids de los documentos importados como resultado de la ejecución.

- **Respuesta con errores:** En caso de que haya algún error en el proceso de publicación, saltará una excepción que contendrá un mensaje de texto con el error producido. Ejemplos de este tipo de excepciones sería que el documento solicitado no exista en la estación.

EJEMPLOS

Código del cliente:

```
public String publishDocument(File file, int publishType, String process) {
    String retorno = "";
    try {
        DataSource dataSource = new FileDataSource(file);
        DataHandler data = new DataHandler(dataSource);
        retorno = service.publishDocument(data, publishType, process);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException Re) {
        Re.printStackTrace();
        System.out.println("RemoteExcepcion: " + Re.toString());
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("Exception: " + e.toString());
    }
    return retorno;
}
```

Petición SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:publishDocument
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
      <dataHandler href="cid:F646291D6CC366075B6ECC671874008E"
xsi:type="ns2:DataHandler" xmlns:ns2="http://xml.apache.org/xml-soap"/>
      <publishType href="#id0"/>
      <process xsi:type="soapenc:string" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" />
    </ns1:publishDocument>
    <multiRef id="id0" soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="xsd:int"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">1</multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

Objetos adjuntados dentro de la petición:

```
-----_Part_0_1239943370.1381246725541
Content-Type: application/zip
Content-Transfer-Encoding: binary
Content-Id: <F646291D6CC366075B6ECC671874008E>

PK#####|#####ECI PESOS EDI
# %7#saQ#t5M A: d7 { ## #m #R #& V #EJ+ ! #e
[...]
```

Nota: Los datos que aparecen en este ejemplo no son reales. El código del fichero binario embebido en la petición SOAP ha sido resumido [...]

Respuesta SOAP:

Ejemplo de una respuesta de **publicación satisfactoria**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:publishDocumentResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
      <publishDocumentReturn xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">148418</publishDocument
Return>
    </ns1:publishDocumentResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Ejemplo de una respuesta de **error**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --&gt; Cod: 23; Text: Errores del documento
&lt;Id:115067&gt; &lt;Referencia:NOBORRAR&gt; &lt;Origen:8438001303004&gt;
&lt;Destino:8422416000504&gt; &lt;Fichero:D115067.DOE&gt;~POS=No existe el Punto
Operacional Origen &lt;8438001303004&gt;~|</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">23</cod>
          <text xsi:type="xsd:string">Errores del documento
&lt;Id:115067&gt; &lt;Referencia:NOBORRAR&gt; &lt;Origen:8438001303004&gt;
&lt;Destino:8422416000504&gt; &lt;Fichero:D115067.DOE&gt;~POS=No existe el Punto
Operacional Origen &lt;8438001303004&gt;~|</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.6. PUBLICACIÓN DE DOCUMENTOS EN BASE64

publishDocumentB64

Permite al usuario autenticado a través del webservice importar documentos en salida en la estación Ediwin sin utilizar SoapAttachments.

Los documentos se pasan al webservice mediante un único fichero comprimido en formato ZIP. De esta forma, se pueden publicar en Ediwin todos los documentos que sean necesarios.

LLAMADA:

```
public String publishDocumentB64(byte[] message, int publishType, String process);
```

PARÁMETROS:

- **message:** Documento en formato zip con los documentos a importar. Se envía en la petición como un SoapAttachment.
- **publishType:** Tipo de publicación:
 1. Publicación de los documentos en entrada.
 2. Publicación de los documentos en salida.
- **process:** Nombre del proceso que se quiere que se ejecute con los ficheros que se suben al servidor. Si se especifica este parámetro, lo que se importará en la estación Ediwin serán los ficheros generados por este proceso, y no los documentos enviados.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya publicado correctamente el documento, devolverá un mensaje de texto con los ids de los documentos importados como resultado de la ejecución.
- **Respuesta con errores:** En caso de que haya algún error en el proceso de publicación, saltará una excepción que contendrá un mensaje de texto con el error producido. Ejemplos de este tipo de excepciones sería que el documento solicitado no exista en la estación.

EJEMPLOS

Código del cliente:

```
public String publishDocumentB64(File file, int publishType, String process) {
    String retorno = "";
    try {
        FileInputStream fis = new FileInputStream(file);
        byte[] data = fis.read();
        retorno = service.publishDocumentB64(data, publishType, process);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException Re) {
        Re.printStackTrace();
        System.out.println("RemoteExcepcion: " + Re.toString());
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("Exception: " + e.toString());
    }
    return retorno;
}
```

Petición SOAP:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:publishDocumentB64
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <message xsi:type="soapenc:base64Binary"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">UESDBBQACAAIAFVMHgAAAAA
AAAAAAAAAAAAeAAAAARURJX0RFU0FEVl9EXzk2QV9VT19FQU4wMDUuRURJlZ3djlXHcgYfZy7651T+Vs0d
JXLXmLZceSV5vXr/B/EZni8JAWsYCKIiwZGKTJIXMx3B7Orfv/7H68s/ty3fZnVen7/8+unzf79/fj/16
f33r+9fPn29rnz74a8/vzzXa5lHVr/O2+fffn4t73e7lq117fdl2dunv/30er3+p97+8Ze/vP7+9f
[...]
```

```

vt5+//j2Y0Xtyz+3fVxnbfeD/f8FUEsHCKZ9fhGjDgAAeH8AAFBLAQIUABQACAAIAFVMHgCmfX4Row4AA
Hh/AAAEAAAAAAAAAAAAAAAAABFRELfREVTQURWX0RfOTZBX1VOX0VBTjAwNS5FRElQSUGAAAAAA
EAAQBMAAAA7w4AAAAA</message>
    <publishType xsi:type="xsd:int">2</publishType>
    <proccess xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    </urn:publishDocumentB64>
  </soapenv:Body>
</soapenv:Envelope>

```

Nota: Los datos que aparecen en este ejemplo no son reales. El código del fichero en base64 ha sido resumido [...]

Respuesta SOAP:

Ejemplo de una respuesta de **publicación satisfactoria**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:publishDocumentB64Response
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
      <publishDocumentB64Return xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">148419</publishDocument
B64Return>
    </ns1:publishDocumentB64Response>
  </soapenv:Body>
</soapenv:Envelope>

```

Ejemplo de una respuesta de **error**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --&gt; Cod: 23; Text: Errores del documento
&lt;Id:115067&gt; &lt;Referencia:NOBORRAR&gt; &lt;Origen:8438001303004&gt;
&lt;Destino:8422416000504&gt; &lt;Fichero:D115067.DOE&gt;~POS=No existe el Punto
Operacional Origen &lt;8438001303004&gt;~|</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">23</cod>
          <text xsi:type="xsd:string">Errores del documento
&lt;Id:115067&gt; &lt;Referencia:NOBORRAR&gt; &lt;Origen:8438001303004&gt;
&lt;Destino:8422416000504&gt; &lt;Fichero:D115067.DOE&gt;~POS=No existe el Punto
Operacional Origen &lt;8438001303004&gt;~|</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.7. EJECUTAR COMPLEMENTO

complementFunction

Permite al usuario autenticado a través del webservice invocar la ejecución de las funciones de un complemento que tenga configurado el usuario.

LLAMADA:

```
public String complementFunction(String complementName, String
complementFunction, String complementParamEnt);
```

PARÁMETROS:

- **complementName:** Nombre del complemento a invocar
- **complementFunction:** Función del complemento que hay que ejecutar
- **complementParamEnt:** Listado de parámetros de entrada para el complemento.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya ejecutado correctamente el complemento, devolverá el texto del resultado de la ejecución de la ejecución del complemento.
- **Respuesta con errores:** En caso de que haya algún error en el proceso de publicación, saltará una excepción que contendrá un mensaje de texto con el error producido en la ejecución del complemento.

EJEMPLOS

Código del cliente:

```
public String complementFunction(String complementName, String
complementFunction, String complementParamEnt) {
    String retorno = "";
    try {
        retorno = service.complementFunction(complementName,
complementFunction, complementParamEnt);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException Re) {
        Re.printStackTrace();
        System.out.println("RemoteExcepcion: " + Re.toString());
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("Exception: " + e.toString());
    }
    return retorno;
}
```

Petición SOAP:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
```

```

    <soapenv:Header/>
    <soapenv:Body>
      <urn:complementFunction
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <complementName xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">IECTE_XXX</complementName>
        <complementFunction xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">FUNCTION_XXX</complementFunction>
        <complementParamEnt xsi:type="soapenc:string"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">PARAM1=VALUE1;PARAM2=VALUE2</complementParamEnt>
        </urn:complementFunction>
      </soapenv:Body>
    </soapenv:Envelope>

```

Respuesta SOAP:

Ejemplo de una respuesta de ejecución de complemento **satisfactoria**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:complementFunctionResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:EdiwinWS">
      <complementFunctionReturn xsi:type="soapenc:string"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">RESULT_COMPLEMENT</complementFunctionReturn>
    </ns1:complementFunctionResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Ejemplo de una respuesta de **error**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --> Cod: 9; Text: No Existe la función</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
          xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">9</cod>
          <text xsi:type="xsd:string">No Existe la función</text>
        </ns1:fault>
        <ns3:hostname
          xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
        </detail>
      </soapenv:Fault>
    </soapenv:Body>
  </soapenv:Envelope>

```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.8. EXPORTAR DOCUMENTOS

exportDocuments

Permite al usuario autenticado a través del webservice exportar los documentos que tiene disponibles en la estación Ediwin utilizando SoapAttachments.

Tipos de exportaciones disponibles a través de este webservice:

- Exportación de los documentos originales en su formato de origen (EDIFACT, XML, etc.)
- Exportación de los documentos en formato LOT.
- Exportación de los documentos según estén configurados en la estación Ediwin.
- Exportación de los documentos utilizando un mapa de usuario para obtener los documentos finales.

LLAMADA:

```
public DataHandler exportDocuments (String filter, int type, String process, int
idVolume, String fileName, Boolean showDate, Boolean oneFileDoc, Boolean genUnb,
Boolean genControl);
```

PARÁMETROS:

- **filter:** Filtro SQL para identificar los documentos. Campos para utilizar en los filtros:
 1. REFERENCIA: Campo REFERENCIA
 2. ORIGEN: Campo ENV_ORIGEN
 3. DESTINO: Campo ENV_DESTINO
 4. ESTADO: Campo ESTADOEXT
 5. FECHA CAMBIO DE ESTADO: Campo FECHACAMBIOESTADO
 Ejemplo: REFERENCIA='FACTURA001'
- **type:** Tipo de exportación (DOC/LOT/CONF/MAP)
 1. Formato origen.
 2. Formato Lot.
 3. Según configuración.
 4. Mapeando.
- **process:** Nombre del mapa del usuario a través del cual hay que transformar los documentos originales. Este parámetro sólo se tiene en cuenta si se utiliza el tipo de exportación MAP (exportar mapeando).
- **IdVolume:** Identificador del volumen del que se quiere obtener los datos del documento. Por defecto, la estación es el id 0.
- **fileName:** Nombre del fichero que se quiere exportar.
- **showDate:** Si es true, se añade en el nombre del fichero la fecha del documento. Por defecto es false.
- **oneFileDoc:** Si es true, se genera un fichero por documento. Por defecto es false.
- **genUnb:** Si es true, genera el segmento UNB.
- **genControl:** Si es true genera también el registro de control de los ficheros exportados. Por defecto es false.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya obtenido correctamente el/los documentos, devolverá un fichero adjunto en formato ZIP con los documentos extraídos.

- **Respuesta de error:** En caso de que haya algún error en el proceso de recuperación, saltará una excepción que contendrá un mensaje de texto con el error producido. Ejemplos de este tipo de excepciones sería que el documento solicitado no exista en la estación.

EJEMPLOS

Código del cliente:

```
public DataHandler exportDocuments(String filter, int type, String process, int
idVolume, String fileName, Boolean showDate, Boolean oneFileDoc, Boolean genUnb,
Boolean genControl) {
    DataHandler dato = null;
    try {
        dato = service.exportDocuments(filter, type, process, idVolume,
fileName, showDate, oneFileDoc, genUnb, genControl);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException rE) {
        rE.printStackTrace();
        System.out.println("RemoteExcepcion: " + rE.toString());
    }
    return dato;
}
```

Petición SOAP:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:exportDocuments
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <filter xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">REFERENCIA='FACTURA01'<
/filter>
      <type xsi:type="xsd:int">1</type>
      <process xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></process>
      <idVolume xsi:type="xsd:int">0</idVolume>
      <fileName xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></fileName>
      <showDate xsi:type="soapenc:boolean" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></showDate>
      <oneFileDoc xsi:type="soapenc:boolean" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></oneFileDoc>
      <genUnb xsi:type="soapenc:boolean" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></genUnb>
      <genControl xsi:type="soapenc:boolean" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></genControl>
    </urn:exportDocuments>
  </soapenv:Body>
</soapenv:Envelope>
```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de **exportación de documentos satisfactoria**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:exportDocumentsResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
```



```

    <exportDocumentsReturn href="cid:45EF6AB92FCD5C9FCDBFFD13C95A1699"
xsi:type="ns2:DataHandler" xmlns:ns2="http://xml.apache.org/xml-soap"/>
  </ns1:exportDocumentsResponse>
</soapenv:Body>
</soapenv:Envelope>

```

Objeto con los datos obtenidos:

```

-----_Part_0_2145549774.1381398336690
Content-Type: application/zip
Content-Transfer-Encoding: binary
Content-Id: <45EF6AB92FCD5C9FCDBFFD13C95A1699>

PK#####@/#####XML_EDIWINSYSTEM.XML[#####~#f#7a##`###/##
7####Kc#g#N####9{K3##fE####WE]W##u,h####?####jVf####B##
[.]
$CNT$PK#####@/##{5V###m#####XML_MESSAGESTATUS.XMLPK#####
##@/##H##{#####XML_MESSAGESTATUS.$CNT$PK#####
-----_Part_0_2145549774.1381398336690

```

Nota: Los datos que aparecen en este ejemplo no son reales. El código del fichero binario embebido en la petición SOAP ha sido resumido [...]

El siguiente ejemplo es una respuesta de error.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --&gt; Cod: 7; Text: No se ha exportado ning&#xFA;n
documento. Por favor, revise el filtro de b&#xFA;squeda de
documentos.</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">7</cod>
          <text xsi:type="xsd:string">No se ha exportado ning&#xFA;n
documento. Por favor, revise el filtro de b&#xFA;squeda de documentos.</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.9. EXPORTAR DOCUMENTOS EN BASE64

exportDocumentsB64

Permite al usuario autenticado a través del webservice exportar los documentos que tiene disponibles en la estación Ediwin sin utilizar SoapAttachments.

Tipos de exportaciones disponibles a través de este webservice:

- Exportación de los documentos originales en su formato de origen (EDIFACT, XML, etc.)

- Exportación de los documentos en formato LOT.
- Exportación de los documentos según estén configurados en la estación Ediwin.
- Exportación de los documentos utilizando un mapa de usuario para obtener los documentos finales.

LLAMADA:

```
public DataHandler exportDocumentsB64 (String filter, int type, String process,
int idVolume, String fileName, Boolean showDate, Boolean oneFileDoc, Boolean
genUnb, Boolean genControl);
```

PARÁMETROS:

- **filter:** Filtro SQL para identificar los documentos. Campos para utilizar en los filtros:
 1. REFERENCIA: Campo REFERENCIA
 2. ORIGEN: Campo ENV_ORIGEN
 3. DESTINO: Campo ENV_DESTINO
 4. ESTADO: Campo ESTADOEXT
 5. FECHA CAMBIO DE ESTADO: Campo FECHACAMBIOESTADO

Ejemplo: REFERENCIA='FACTURA001'
- **type:** Tipo de exportación (DOC/LOT/CONF/MAP)
 1. Formato origen.
 2. Formato Lot.
 3. Según configuración.
 4. Mapeando.
- **process:** Nombre del mapa del usuario a través del cual hay que transformar los documentos originales. Este parámetro sólo se tiene en cuenta si se utiliza el tipo de exportación MAP (exportar mapeando).
- **idVolume:** Identificador del volumen del que se quiere obtener los datos del documento. Por defecto, la estación es el id 0.
- **fileName:** Nombre del fichero que se quiere exportar.
- **showDate:** Si es true, se añade en el nombre del fichero la fecha del documento. Por defecto es false.
- **oneFileDoc:** Si es true, se genera un fichero por documento. Por defecto es false.
- **genUnb:** Si es true, genera el segmento UNB.
- **genControl:** Si es true genera también el registro de control de los ficheros exportados. Por defecto es false.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya obtenido correctamente el/los documentos, devolverá un fichero adjunto en formato ZIP con los documentos extraídos.
- **Respuesta de error:** En caso de que haya algún error en el proceso de recuperación, saltará una excepción que contendrá un mensaje de texto con el error producido. Ejemplos de este tipo de excepciones sería que el documento solicitado no exista en la estación.

EJEMPLOS

Código del cliente:

```

public DataHandler exportDocumentsB64(String filter, int type, String process,
int idVolume, String fileName, Boolean showDate, Boolean oneFileDoc, Boolean
genUnb, Boolean genControl) {
    byte[] dato = null;
    try {
        dato = service.exportDocumentsB64(filter, type, process, idVolume,
fileName, showDate, oneFileDoc, genUnb, genControl);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException rE) {
        rE.printStackTrace();
        System.out.println("RemoteExcepcion: " + rE.toString());
    }
    return dato;
}

```

Petición SOAP:

```

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:exportDocumentsB64
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
            <filter xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">TIPO='E' AND
ESTADO='COR'</filter>
            <type xsi:type="xsd:int">1</type>
            <process xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></process>
            <idVolume xsi:type="xsd:int">0</idVolume>
            <fileName xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></fileName>
            <showDate xsi:type="soapenc:boolean" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></showDate>
            <oneFileDoc xsi:type="soapenc:boolean" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></oneFileDoc>
            <genUnb xsi:type="soapenc:boolean" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></genUnb>
            <genControl xsi:type="soapenc:boolean" xsi:nil="true"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"></genControl>
        </urn:exportDocumentsB64>
    </soapenv:Body>
</soapenv:Envelope>

```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de **exportación de documentos satisfactoria**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <ns1:exportDocumentsB64Response
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
            <exportDocumentsB64Return xsi:type="soapenc:base64Binary"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">UESDBBQACAAIAF40+v8AAAA
AAAAAAAAAAAAeAAAARURJX0RFU0FEVl9EXzk2QV9VTl9FQU4wMDUuRURJ7ZbBbqMwEIYfx4eRks/YBjM3
wE6abSAEQrs57fu/xRoDaUJYbSJVPtWRiD3z2/L/jcahr1/Bf1hLWZpIC853uXtnx1mSc1+zz2spj
[...]
```

```

    </ns1:exportDocumentsB64Response>
  </soapenv:Body>
</soapenv:Envelope>

```

Nota: Los datos que aparecen en este ejemplo no son reales. El código del fichero en base64 ha sido resumido [...]

El siguiente ejemplo es una respuesta de **error**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --> Cod: 7; Text: No se ha exportado ning&#xFA;n
documento. Por favor, revise el filtro de b&#xFA;squeda de
documentos.</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">7</cod>
          <text xsi:type="xsd:string">No se ha exportado ning&#xFA;n
documento. Por favor, revise el filtro de b&#xFA;squeda de documentos.</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.10. CAMBIAR DOCUMENTOS DE SITUACIÓN

changeDocSituation

Permite al usuario autenticado a través del webservice cambiar de situación los documentos que tiene disponibles en la estación Ediwin.

LLAMADA:

```

public String changeDocSituation(String filter, String sourceSit, String
destinationSit);

```

PARÁMETROS:

- **filter:** Filtro SQL para identificar los documentos. Campos para utilizar en los filtros:

1. REFERENCIA: Campo REFERENCIA
2. ORIGEN: Campo ENV_ORIGEN
3. DESTINO: Campo ENV_DESTINO
4. ESTADO: Campo ESTADOEXT
5. FECHA CAMBIO DE ESTADO: Campo FECHACAMBIOESTADO

Ejemplo: REFERENCIA='FACTURA001'

- **sourceSit:** Situación inicial de los documentos seleccionados:
 - Tipo salida:
 - PEN: Pendientes.
 - DEP: Depositados.
 - ENV: Enviados.
 - RCP: Recuperados.
 - NTF: Notificado.
 - RH1: Rechazados tipo 1: No han llegado al servidor.
 - RH2: Rechazados tipo 2: Rechazados por el servidor.
 - RH3: Rechazados tipo 3: Enviados al buzón del destinatario, pero no recibidos por éste en el tiempo configurado.
 - Tipo entrada:
 - REC: Recibidos.
 - TRA: Tratados.
 - DestinationSit: Situación final de los documentos seleccionados. Son los mismos estados que en el parámetro sourceSit.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya procesado correctamente el cambio de situación de los documentos seleccionados, devolverá el mensaje "OK". Ediwin intentará cambiar de situación aquellos documentos del filtro que cumplan las condiciones de que están en el estado inicial y pueden ser cambiados a la situación final. El hecho de no poder mover ningún documento no significa que la ejecución del proceso sea erróneo.
- **Respuesta de error:** En caso de que haya algún error en el proceso de recuperación, saltará una excepción que contendrá un mensaje de texto con el error producido.

EJEMPLOS

Código del cliente:

```
public String changeDocSituation(String filter, String sourceSit, String
destinationSit) {
    String retorno = "";
    try {
        retorno = service.changeDocSituation(filter, sourceSit,
destinationSit);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEException: " + EWSe.getText());
    } catch (RemoteException Re) {
        Re.printStackTrace();
        System.out.println("RemoteExcepcion: " + Re.toString());
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("Exception: " + e.toString());
    }
    return retorno;
}
```

Petición SOAP:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:changeDocSituation
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    <filter xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">REFERENCIA='FACTURA01'<
  /filter>
    <sourceSit xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">PEN</sourceSit>
    <destinationSit xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">DEP</destinationSit>
    </urn:changeDocSituation>
  </soapenv:Body>
</soapenv:Envelope>
```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de **cambio de situación satisfactoria**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:changeDocSituationResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
      <changeDocSituationReturn xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">OK</changeDocSituationR
    eturn>
    </ns1:changeDocSituationResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

El siguiente ejemplo es una respuesta de **error**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --> Cod: 13; Text: No se ha especificado ningún
documento para cambiar su situación.</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">13</cod>
          <text xsi:type="xsd:string">No se ha especificado ningún documento
para cambiar su situación.</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.11. CAMBIO DE LOS CAMPOS EXTENDIDOS DE LOS DOCUMENTOS

changeExtendedFieldValue

Permite al usuario autenticado a través del webservice cambiar los valores de los campos extendidos de los documentos que tiene disponibles en la estación Ediwin.

LLAMADA:

```
public String changeExtendedFieldValue (String filter, String[] fieldNames,
String[] fieldValues);
```

PARÁMETROS:

- **filter:** Filtro SQL para identificar los documentos. Campos para utilizar en los filtros:
 1. REFERENCIA: Campo REFERENCIA
 2. ORIGEN: Campo ENV_ORIGEN
 3. DESTINO: Campo ENV_DESTINO
 4. ESTADO: Campo ESTADOEXT
 5. FECHA CAMBIO DE ESTADO: Campo FECHACAMBIOESTADO

Ejemplo: REFERENCIA='FACTURA001'
- **fieldNames:** Es un listado de los nombres de los campos que se quieren modificar.
- **fieldValues:** Es un listado de los valores que hay que insertar en cada uno de los campos que se quieren modificar. Ha de tener un valor por cada campo introducido en el parámetro fieldNames, en el mismo orden, y se insertan siempre como cadenas de texto.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya procesado correctamente el cambio de situación de los documentos seleccionados, devolverá el mensaje "OK".
- **Respuesta con error:** En caso de que haya algún error en el proceso de recuperación, saltará una excepción que contendrá un mensaje de texto con el error producido.

EJEMPLOS

Código del cliente:

```
public String changeExtendedFieldValue (String filter, String[] fieldNames,
String[] fieldValues) {
    String retorno = "";
    try {
        retorno = service.changeExtendedFieldValue (filter, fieldNames,
fieldValues);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException Re) {
        Re.printStackTrace();
        System.out.println("RemoteExcepcion: " + Re.toString());
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("Exception: " + e.toString());
    }
    return retorno;
}
```

}

Petición SOAP:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:changeExtendedFieldValue
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <filter xsi:type="soapenc:string">REFERENCIA='FACTURA01'</filter>
      <fieldNames xsi:type="urn:ArrayOf_soapenc_string"
soapenc:arrayType="xsd:string[]">ESTADOEXT</fieldNames>
      <fieldValues xsi:type="urn:ArrayOf_soapenc_string"
soapenc:arrayType="xsd:string[]">1</fieldValues>
    </urn:changeExtendedFieldValue>
  </soapenv:Body>
</soapenv:Envelope>
```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de **modificación de campos extendidos satisfactoria**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:changeExtendedFieldValueResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
      <changeDocSituationReturn xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">OK</changeDocSituationR
eturn>
    </ns1:changeExtendedFieldValueResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

El siguiente ejemplo es una respuesta de **error**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error--> Cod: 14; Text: No se ha especificado ningún
documento para cambiar los campos extendidos.</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">14</cod>
          <text xsi:type="xsd:string">No se ha especificado ningún documento
para cambiar los campos extendidos.</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.12. OBTENCIÓN DE DATOS ESTADÍSTICOS

`getStatistics`

Permite al usuario autenticado a través del webservice obtener datos de las estadísticas de los documentos que tiene disponibles en la estación Ediwin utilizando SoapAttachments.

Datos disponibles a través de este webservice:

- Gráfica de documentos en formato jpg.
- Gráfica de protocolos de comunicaciones en formato jpg.
- Datos de los documentos enviados en formato CSV.
- Listado de datos de los documentos enviados en formato PDF.

LLAMADA:

```
public DataHandler getStatistics(Statistic statistic);
```

PARÁMETROS:

- **statistic:** Objeto que contiene la información de la solicitud de estadísticas. Este objeto tiene los siguientes campos:
 1. **type:** Indica el tipo de estadística que se quiere generar. Por defecto asume tipo documentos. Posibles valores:
 - 1. DOCUMENTOS: Genera una gráfica en formato jpg de documentos.
 - 2. PROTOCOLOS DE COMUNICACIONES: Genera una gráfica en formato jpg de protocolos de comunicaciones.
 - 3. PDF: Genera un listado en formato pdf de documentos enviados.
 - 4. CSV: Genera un csv con los documentos agrupados.
 2. **from:** Indica la fecha de inicio de las estadísticas. Si no se indica es desde el primer documento del dominio.
 3. **until:** Indica la fecha de fin de las estadísticas. Si no se indica es hasta el momento actual.
 4. **groupBy:** Indica el tipo de agrupación temporal que hay que hacer con los datos. Por defecto se agrupa por meses. Posible valores:
 - 0. MESES: Agrupa los resultados por meses.
 - 1. AÑOS: Agrupa los resultados por años.
 - 2. TOTALES: Agrupa los resultados por totales (documentos o protocolos).
 - 3. INTERFACES: Agrupa los resultados por interfaces.
 - 4. PROTOCOLOS: Agrupa los resultados por protocolos.
 - 5. CRIPTO: Agrupa los resultados por cripto.
 - 6. INTERLOCUTORES: Agrupa los resultados por interlocutores.
 5. **height:** Indica el alto de la imagen jpg a crear. Sólo tiene sentido en los tipos documentos y protocolos.
 6. **width:** Indica el ancho de la imagen jpg a crear. Sólo tiene sentido en los tipos documentos y protocolos.

7. **dataAt:** Indica el tipo de datos a mostrar. Por defecto muestra los datos como números. Posibles valores:
 - 0. KBYTES: Muestra los datos por tamaño de los documentos.
 - 1. NUMEROS: Muestra los datos por cantidades de documentos.
8. **scale:** Muestra la escala de los datos generados en las gráficas. Por defecto es escala normal. Sólo tiene sentido en los tipos documentos y protocolos. Posibles valores:
 - 0. LOGARITMICA: Crea una imagen con escala logarítmica.
 - 1. NORMAL: Crea una imagen con escala decimal.
9. **neutralColors:** Indica si utiliza colores neutros (para daltónicos). Por defecto utiliza colores normales. Posibles valores:
 - false: Utiliza colores normales.
 - true: Utiliza colores neutros.
10. **source:** Indica el interlocutor origen de los documentos que se tratan en la estadística.
11. **destination:** Indica el interlocutor destino de los documentos que se tratan en la estadística.
12. **logicSource:** Indica si se tienen en cuenta los interlocutores lógicos del interlocutor origen indicado. Por defecto no se tienen en cuenta. Posibles valores:
 - 0: No se tienen en cuenta.
 - 1: Se tienen en cuenta.
13. **logicDestination:** Indica si se tienen en cuenta los interlocutores lógicos del interlocutor destino indicado. Por defecto no se tienen en cuenta. Posibles valores:
 - 0: No se tienen en cuenta.
 - 1: Se tienen en cuenta.
14. **crypto:** Indica el tipo de firma por el que se quiere filtrar en la estadística.
15. **interfaz:** Indica el tipo de interfaz por el que se quiere filtrar en la estadística.
16. **state:** Indica el estado por el que se quiere filtrar en la estadística.
17. **msgType:** Indica el tipo de mensaje por el que se quiere filtrar en la estadística.
18. **dateInUTC:** Indica si se quiere utilizar fechas en formato UTC. Por defecto no se utilizan fechas en formato UTC.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya obtenido correctamente la estadística, devolverá un fichero adjunto en formato ZIP con el documento correspondiente (jpg, pdf o csv).
- **Respuesta con errores:** En caso de que haya algún error en el proceso de generación de la estadística, saltará una excepción que contendrá un mensaje de texto con el error producido. Ejemplos de este tipo de excepciones sería por ejemplo que el tipo de estadística no es un tipo válido.

EJEMPLOS

Código del cliente:

```
public DataHandler getStatistics(Statistic statistic) {
    DataHandler dato = null;
    try {
        dato = service.getStatistics(statistic);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
    }
}
```

```

        System.out.println("EdiwinWSExcepcion: " + EWSe.getText());
    } catch (RemoteException rE) {
        rE.printStackTrace();
        System.out.println("RemoteExcepcion: " + rE.toString());
    }
    return dato;
}

```

Petición SOAP:

```

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getStatistics
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    <statistic xsi:type="bean:Statistic"
xmlns:bean="http://beans.service.ediwinws.edicom.com">
      <crypto xsi:type="xsd:string"></crypto>
      <dataAt xsi:type="xsd:int">0</dataAt>
      <dateInUTC xsi:type="xsd:boolean">false</dateInUTC>
      <destination xsi:type="xsd:string">AAAAAAA</destination>
      <from xsi:type="xsd:dateTime">2013-01-01T00:00:00.000Z</from>
      <groupBy xsi:type="xsd:int">2</groupBy>
      <height xsi:type="xsd:int">0</height>
      <interfaz xsi:type="xsd:string"></interfaz>
      <logicDestination xsi:type="xsd:string"></logicDestination>
      <logicSource xsi:type="xsd:string"></logicSource>
      <msgType xsi:type="xsd:string"></msgType>
      <neutralColors xsi:type="xsd:boolean">false</neutralColors>
      <scale xsi:type="xsd:int">1</scale>
      <source xsi:type="xsd:string"></source>
      <state xsi:type="xsd:string"></state>
      <type xsi:type="xsd:int">1</type>
      <until xsi:type="xsd:dateTime">2013-09-01T00:00:00.000Z</until>
      <width xsi:type="xsd:int">0</width>
    </statistic>
  </urn:getStatistics>
</soapenv:Body>
</soapenv:Envelope>

```

Nota: Los datos que aparecen en este ejemplo no son reales. El código del fichero en base64 ha sido resumido [...]

Respuesta SOAP:

El siguiente ejemplo es una respuesta satisfactoria de la obtención de un UUID

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getStatisticsResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
      <getStatisticsReturn href="cid:DA2B2B7C99B55E4BB284946A3E2BE00A"
xsi:type="ns2:DataHandler" xmlns:ns2="http://xml.apache.org/xml-soap"/>
    </ns1:getStatisticsResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Objeto con los datos obtenidos:

```
-----=_Part_4_2129362951.1381402305494
```

```
Content-Type: application/zip
Content-Transfer-Encoding: binary
Content-Id: <DA2B2B7C99B55E4BB284946A3E2BE00A>

PK#####@/#####DATA.CSV[#####~f#7a#####`#####/#####Kc#g#
N#####9{K3#####WE]#####h#####j#####v#####B
[.]
#####@/#####H#####(#####DATA.CSVPK##########
-----=_Part_4_2129362951.1381402305494
```

Nota: Los datos que aparecen en este ejemplo no son reales. El código del fichero binario embebido en la petición SOAP ha sido resumido [...]

El siguiente ejemplo es una respuesta de **error**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --> Cod: 37; Text: Se han producido errores al
obtener datos de las estadísticas del dominio.</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">37</cod>
          <text xsi:type="xsd:string">Se han producido errores al obtener
datos de las estadísticas del dominio.</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">ed267</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.13. OBTENCIÓN DE INTERFACES.

getInterfaces

Permite al usuario autenticado a través del webservice obtener la lista de interfaces disponibles para el usuario.

LLAMADA:

```
public String[] getInterfaces();
```

PARÁMETROS:

No tiene

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya ejecutado correctamente la llamada, nos devolverá un listado en modo texto de los interfaces disponibles por el usuario.

- **Respuesta con errores:** En caso de que haya algún error en el proceso, saltará una excepción que contendrá un mensaje de texto con el error producido.

EJEMPLOS

Código del cliente:

```
public String[] getInterfaces() {
    String[] interfaces = null;
    try {
        interfaces = service.getInterfaces();
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException rE) {
        rE.printStackTrace();
        System.out.println("RemoteExcepcion: " + rE.toString());
    }
    return interfaces;
}
```

Petición SOAP:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getInterfaces
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </soapenv:Body>
</soapenv:Envelope>
```

Respuesta SOAP:

El siguiente ejemplo es una respuesta **satisfactoria**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getInterfacesResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
      <getInterfacesReturn soapenc:arrayType="soapenc:string[86]"
xsi:type="soapenc:Array"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <getInterfacesReturn
xsi:type="soapenc:string">EDI_DESADV_D_96A_UN_EAN005</getInterfacesReturn>
xsi:type="soapenc:string">EDI_INVOIC_D_93A_UN_EAN007</getInterfacesReturn>
        <getInterfacesReturn
xsi:type="soapenc:string">EDI_INVOIC_D_96A_UN_EAN008</getInterfacesReturn>
        <getInterfacesReturn
xsi:type="soapenc:string">EDI_ORDERS_D_96A_UN_EAN008</getInterfacesReturn>
      </getInterfacesReturn>
    </ns1:getInterfacesResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

El siguiente ejemplo es una respuesta de **error**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
```

```

<faultcode>soapenv:Server.userException</faultcode>
<faultstring>Error --> Cod: 0; Text: Error desconocido.</faultstring>
<detail>
  <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
    <cod xsi:type="xsd:int">0</cod>
    <text xsi:type="xsd:string">Error desconocido.</text>
  </ns1:fault>
  <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">ed267</ns3:hostname>
</detail>
</soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>

```

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

2.14. OBTENCIÓN DE INTERLOCUTORES

getInterlocutors

Permite al usuario autenticado a través del webservice obtener los interlocutores configurados en la libreta de direcciones.

Se pueden obtener los interlocutores propios o externos.

LLAMADA:

```
public Interlocutor[] getInterlocutors(int interlocutorType, boolean
onlyPhysics);
```

PARÁMETROS:

- **interlocutorType:** Indica si se obtienen los interlocutores propios o externos. Posibles valores:
 1. EXTERNO: Obtiene los interlocutores externos.
 2. PROPIO: Obtiene los interlocutores propios.
- **onlyPhysics:** Indica si se obtienen sólo los interlocutores físicos, o también los lógicos.

RESPUESTA:

- **Respuesta satisfactoria:** En caso de que se haya ejecutado correctamente el proceso, se obtiene un listado de interlocutores. Estos objetos interlocutor tienen los siguientes campos:
 - **id:** Identificador del interlocutor.
 - **parentId:** Identificador del interlocutor padre.
 - **name:** Nombre del interlocutor.
 - **own:** Si es propio o externo.
 - **type:** Tipo de interlocutor (físico o lógico).
 - **logics:** Listado de interlocutores lógicos. Sólo tendrá datos si es un interlocutor físico que tiene interlocutores lógicos y se ha indicado en la petición que se quieren obtener también los lógicos.

- **Respuesta con errores:** En caso de que haya algún error en el proceso de obtención de los interlocutores, saltará una excepción que contendrá un mensaje de texto con el error producido. Ejemplos de este tipo de excepciones sería por ejemplo que el tipo de interlocutor no es un tipo válido.

Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

EJEMPLOS

Código del cliente:

```
public Interlocutor[] getInterlocutors(int interlocutorType, boolean onlyPhysics)
{
    Interlocutor[] interlocutors = null;
    try {
        interlocutors = service.getInterlocutors(interlocutorType,
onlyPhysics);
    } catch (EdiwinWSEException EWSe) {
        EWSe.printStackTrace();
        System.out.println("EdiwinWSEExcepcion: " + EWSe.getText());
    } catch (RemoteException rE) {
        rE.printStackTrace();
        System.out.println("RemoteExcepcion: " + rE.toString());
    }
    return interlocutors;
}
```

Petición SOAP:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:EdiwinWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getInterlocutors
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <codInterlocutor xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">111111111116</codInter
locutor>
      <interlocutorType xsi:type="xsd:int">0</interlocutorType>
      <onlyPhysics xsi:type="xsd:boolean">true</onlyPhysics>
    </urn:getInterlocutors>
  </soapenv:Body>
</soapenv:Envelope>
```

Respuesta SOAP:

Ejemplo de una respuesta de **cancelación de CFDI firmada satisfactoria**.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getInterlocutorsResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:EdiwinWS">
      <getInterlocutorsReturn soapenc:arrayType="ns2:Interlocutor[1]"
xsi:type="soapenc:Array" xmlns:ns2="http://beans.service.ediwinws.edicom.com"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <getInterlocutorsReturn soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns3:Interlocutor"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns3="http://beans.service.ediwinws.edicom.com">
          <id xsi:type="xsd:string">111111111116</id>
```

```

        <logics soapenc:arrayType="ns3:Interlocutor[0]"
xsi:type="soapenc:Array"/>
        <name xsi:type="xsd:string">"1QQQQ"</name>
        <own xsi:type="xsd:boolean">false</own>
        <parentId xsi:type="xsd:string">1111111111116</parentId>
        <type xsi:type="xsd:int">2</type>
    </getInterlocutorsReturn>
</getInterlocutorsReturn>
</ns1:getInterlocutorsResponse>
</soapenv:Body>
</soapenv:Envelope>

```

Ejemplo de una respuesta de **error**.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>soapenv:Server.userException</faultcode>
            <faultstring>Error --> Cod: 6; Text: El tipo de interlocutor enviado no
es válido. Los tipos válidos son Propio: 1 y Externo: 0.</faultstring>
            <detail>
                <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
                    <cod xsi:type="xsd:int">6</cod>
                    <text xsi:type="xsd:string">El tipo de interlocutor enviado no es
válido. Los tipos válidos son Propio: 1 y Externo: 0.</text>
                </ns1:fault>
                <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
            </detail>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>

```

Nota: Para ver una lista completa de los posibles códigos de error y respuestas asociadas consultar el punto [Lista de Códigos de Error](#).

3. LISTA DE CÓDIGOS DE ERROR

3.1. INTRODUCCIÓN

Durante la ejecución de las funciones del webservice, se pueden producir varios tipos de errores. El webservice devuelve estos errores en forma de excepción, con un código de error determinado en función del tipo de operación que esté realizando. Además, y en función del error en concreto, se devuelve también un texto explicativo del error que se ha producido.

Código	Error
-1	Error desconocido: <i>Se ha producido un error interno al ejecutar el proceso.</i>
1	Login incorrecto: <i>La autenticación del usuario para acceder al sistema no es correcta.</i>
2	Error de denegación de permisos: No dispones de permiso para realizar la operación solicitada.
3	Error por múltiples informes: Hay distintas posibilidades para mostrar el informe seleccionado.
4	Error por fichero inexistente: No se ha recibido ningún fichero.
5	Error al importar ficheros. No se ha podido importar el fichero al servidor.
6	Error por parámetros erróneos. Los parámetros enviados al servicio son incorrectos.
7	Error al exportar documentos: No se ha podido exportar los documentos.
8	Error en la gestión de usuarios: Error en la gestión de usuarios.
9	Error en la ejecución del complementos: Error en la ejecución del complemento.
10	Error de sesión: Usuario no autenticado o sesión caducada.
12	Error al mapear documentos: No se ha podido mapear el documento.
13	Error en cambio de situación: No se ha podido cambiar la situación del documento.
14	Error en la modificación de documentos: No se ha podido modificar los datos del documento.
18	Error en la gestión de libretas: Error en la gestión de la libreta de interlocutores.
19	Error en la gestión de documentos: Error en la gestión de documentos.
20	Error al iniciar sesión: Debes iniciar sesión en Ediwin para cambiar el password en el primer inicio de sesión.
21	Error al subir ficheros al servidor: No se ha podido subir el fichero al servidor para hacer la operación.
22	Error al obtener documentos importados: Los documentos han sido importados pero no se ha podido obtener sus ids para identificarlos.
23	Error al validar ficheros importados: Los ficheros importados tienen errores de validación.
24	Error de acceso a la base de datos: Error al acceder a la base de datos.
29	Error al gestionar certificados: Se han producido errores al gestionar los certificados.
32	Error en alta de usuarios: Se han producido errores al dar de alta al usuario.
34	Error al editar usuarios: Se han producido errores al editar los datos del usuario.
35	Error al editar la libreta de interlocutores: Se han producido errores al editar los datos del punto operacional.
36	Error en la gestión de complementos: Se han producido errores al editar los complementos.
37	Error en la gestión de estadísticas: Se han producido errores al obtener estadísticas del dominio.

EJEMPLO DE UNA RESPUESTA DE ERROR

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>Error --> Cod: 1; Text: Los datos de autenticación
introducidos son incorrectos.</faultstring>
      <detail>
        <ns1:fault xsi:type="ns2:EdiwinWSEException" xmlns:ns1="urn:EdiwinWS"
xmlns:ns2="http://exception.service.ediwinws.edicom.com">
          <cod xsi:type="xsd:int">1</cod>
          <text xsi:type="xsd:string">Los datos de autenticación
introducidos son incorrectos.</text>
        </ns1:fault>
        <ns3:hostname
xmlns:ns3="http://xml.apache.org/axis/">web.sedeb2b.com</ns3:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```

4. ANEXO: WSDL

4.1. WSDL WEBSERVICE EDIWINWS

El contenido del WSDL del servicio es el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:EdiwinWS"
  xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="urn:EdiwinWS"
  xmlns:intf="urn:EdiwinWS"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns2="http://exception.service.ediwinws.edicom.com"
  xmlns:tns3="http://beans.service.ediwinws.edicom.com"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <schema targetNamespace="urn:EdiwinWS"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://beans.service.ediwinws.edicom.com"/>
      <import namespace="http://exception.service.ediwinws.edicom.com"/>
      <import namespace="http://xml.apache.org/xml-soap"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="ArrayOf_soapenc_string">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="soapenc:string[]" />
          </restriction>
        </complexContent>
      </complexType>
      <complexType name="ArrayOf_tns3_Interlocutor">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="tns3:Interlocutor[]" />
          </restriction>
        </complexContent>
      </complexType>
    </schema>
    <schema targetNamespace="http://exception.service.ediwinws.edicom.com"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://beans.service.ediwinws.edicom.com"/>
      <import namespace="urn:EdiwinWS"/>
      <import namespace="http://xml.apache.org/xml-soap"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="EdiwinWSException">
        <sequence>
          <element name="cod" type="xsd:int"/>
          <element name="text" nillable="true" type="soapenc:string"/>
        </sequence>
      </complexType>
    </schema>
    <schema targetNamespace="http://beans.service.ediwinws.edicom.com"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="urn:EdiwinWS"/>
      <import namespace="http://exception.service.ediwinws.edicom.com"/>
      <import namespace="http://xml.apache.org/xml-soap"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="Statistic">
        <sequence>
          <element name="crypto" nillable="true" type="soapenc:string"/>
          <element name="dataAt" type="xsd:int"/>
          <element name="dateInUTC" type="xsd:boolean"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>

```

```

    <element name="destination" nillable="true" type="soapenc:string"/>
    <element name="from" nillable="true" type="xsd:dateTime"/>
    <element name="groupBy" type="xsd:int"/>
    <element name="height" type="xsd:int"/>
    <element name="interfaz" nillable="true" type="soapenc:string"/>
    <element name="logicDestination" nillable="true" type="soapenc:string"/>
    <element name="logicSource" nillable="true" type="soapenc:string"/>
    <element name="msgType" nillable="true" type="soapenc:string"/>
    <element name="neutralColors" type="xsd:boolean"/>
    <element name="scale" type="xsd:int"/>
    <element name="source" nillable="true" type="soapenc:string"/>
    <element name="state" nillable="true" type="soapenc:string"/>
    <element name="type" type="xsd:int"/>
    <element name="until" nillable="true" type="xsd:dateTime"/>
    <element name="width" type="xsd:int"/>
  </sequence>
</complexType>
<complexType name="Interlocutor">
  <sequence>
    <element name="id" nillable="true" type="soapenc:string"/>
    <element name="logics" nillable="true"
type="impl:ArrayOf_tns3_Interlocutor"/>
    <element name="name" nillable="true" type="soapenc:string"/>
    <element name="own" type="xsd:boolean"/>
    <element name="parentId" nillable="true" type="soapenc:string"/>
    <element name="type" type="xsd:int"/>
  </sequence>
</complexType>
</schema>
</wsdl:types>
<wsdl:message name="publishDocumentB64Response">
  <wsdl:part name="publishDocumentB64Return" type="soapenc:string"/>
</wsdl:message>
<wsdl:message name="exportDocumentsResponse">
  <wsdl:part name="exportDocumentsReturn" type="apachesoap:DataHandler"/>
</wsdl:message>
<wsdl:message name="getStatisticsResponse">
  <wsdl:part name="getStatisticsReturn" type="apachesoap:DataHandler"/>
</wsdl:message>
<wsdl:message name="getInterfacesRequest">
</wsdl:message>
<wsdl:message name="changeExtendedFieldValueRequest">
  <wsdl:part name="filter" type="soapenc:string"/>
  <wsdl:part name="fieldNames" type="impl:ArrayOf_soapenc_string"/>
  <wsdl:part name="fieldValues" type="impl:ArrayOf_soapenc_string"/>
</wsdl:message>
<wsdl:message name="getInterfacesResponse">
  <wsdl:part name="getInterfacesReturn" type="impl:ArrayOf_soapenc_string"/>
</wsdl:message>
<wsdl:message name="endSessionResponse">
</wsdl:message>
<wsdl:message name="publishDocumentB64Request">
  <wsdl:part name="message" type="soapenc:base64Binary"/>
  <wsdl:part name="publishType" type="xsd:int"/>
  <wsdl:part name="proccess" type="soapenc:string"/>
</wsdl:message>
<wsdl:message name="exportDocumentsRequest">
  <wsdl:part name="filter" type="soapenc:string"/>
  <wsdl:part name="type" type="xsd:int"/>
  <wsdl:part name="process" type="soapenc:string"/>
  <wsdl:part name="idVolume" type="soapenc:int"/>
  <wsdl:part name="fileName" type="soapenc:string"/>
  <wsdl:part name="showDate" type="soapenc:boolean"/>
  <wsdl:part name="oneFileDoc" type="soapenc:boolean"/>
  <wsdl:part name="genUnb" type="soapenc:boolean"/>
  <wsdl:part name="genControl" type="soapenc:boolean"/>

```

```

</wsdl:message>
<wsdl:message name="changeExtendedFieldValueResponse">
  <wsdl:part name="changeExtendedFieldValueReturn" type="soapenc:string"/>
</wsdl:message>
<wsdl:message name="getInterlocutorsRequest">
  <wsdl:part name="codInterlocutor" type="soapenc:string"/>
  <wsdl:part name="interlocutorType" type="xsd:int"/>
  <wsdl:part name="onlyPhysics" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="publishDocumentRequest">
  <wsdl:part name="dataHandler" type="apachesoap:DataHandler"/>
  <wsdl:part name="publishType" type="xsd:int"/>
  <wsdl:part name="process" type="soapenc:string"/>
</wsdl:message>
<wsdl:message name="complementFunctionRequest">
  <wsdl:part name="complementName" type="soapenc:string"/>
  <wsdl:part name="complementFunction" type="soapenc:string"/>
  <wsdl:part name="complementParamEnt" type="soapenc:string"/>
</wsdl:message>
<wsdl:message name="changeDocSituationRequest">
  <wsdl:part name="filter" type="soapenc:string"/>
  <wsdl:part name="sourceSit" type="soapenc:string"/>
  <wsdl:part name="destinationSit" type="soapenc:string"/>
</wsdl:message>
<wsdl:message name="getDocumentsB64Response">
  <wsdl:part name="getDocumentsB64Return" type="soapenc:base64Binary"/>
</wsdl:message>
<wsdl:message name="getStatisticsRequest">
  <wsdl:part name="statistic" type="tns3:Statistic"/>
</wsdl:message>
<wsdl:message name="getDocumentsRequest">
  <wsdl:part name="filter" type="soapenc:string"/>
  <wsdl:part name="type" type="xsd:int"/>
  <wsdl:part name="documentTypeList" type="soapenc:string"/>
  <wsdl:part name="idVolume" type="soapenc:int"/>
</wsdl:message>
<wsdl:message name="registerSessionRequest">
  <wsdl:part name="user" type="soapenc:string"/>
  <wsdl:part name="password" type="soapenc:string"/>
  <wsdl:part name="domain" type="soapenc:string"/>
  <wsdl:part name="clientType" type="xsd:int"/>
  <wsdl:part name="group" type="soapenc:string"/>
</wsdl:message>
<wsdl:message name="getDocumentsB64Request">
  <wsdl:part name="filter" type="soapenc:string"/>
  <wsdl:part name="type" type="xsd:int"/>
  <wsdl:part name="documentTypeList" type="soapenc:string"/>
  <wsdl:part name="idVolume" type="soapenc:int"/>
</wsdl:message>
<wsdl:message name="registerSessionResponse">
</wsdl:message>
<wsdl:message name="exportDocumentsB64Request">
  <wsdl:part name="filter" type="soapenc:string"/>
  <wsdl:part name="type" type="xsd:int"/>
  <wsdl:part name="process" type="soapenc:string"/>
  <wsdl:part name="idVolume" type="soapenc:int"/>
  <wsdl:part name="fileName" type="soapenc:string"/>
  <wsdl:part name="showDate" type="soapenc:boolean"/>
  <wsdl:part name="oneFileDoc" type="soapenc:boolean"/>
  <wsdl:part name="genUnb" type="soapenc:boolean"/>
  <wsdl:part name="genControl" type="soapenc:boolean"/>
</wsdl:message>
<wsdl:message name="publishDocumentResponse">
  <wsdl:part name="publishDocumentReturn" type="soapenc:string"/>
</wsdl:message>
<wsdl:message name="complementFunctionResponse">

```

```

    <wsdl:part name="complementFunctionReturn" type="soapenc:string"/>
  </wsdl:message>
  <wsdl:message name="getInterlocutorsResponse">
    <wsdl:part name="getInterlocutorsReturn"
type="impl:ArrayOf_tns3_Interlocutor"/>
  </wsdl:message>
  <wsdl:message name="changeDocSituationResponse">
    <wsdl:part name="changeDocSituationReturn" type="soapenc:string"/>
  </wsdl:message>
  <wsdl:message name="getDocumentsResponse">
    <wsdl:part name="getDocumentsReturn" type="apachesoap:DataHandler"/>
  </wsdl:message>
  <wsdl:message name="EdiwinWSEException">
    <wsdl:part name="fault" type="tns2:EdiwinWSEException"/>
  </wsdl:message>
  <wsdl:message name="endSessionRequest">
  </wsdl:message>
  <wsdl:message name="exportDocumentsB64Response">
    <wsdl:part name="exportDocumentsB64Return" type="soapenc:base64Binary"/>
  </wsdl:message>
  <wsdl:message name="createDomainRequest">
    <wsdl:part name="domain" type="soapenc:string"/>
    <wsdl:part name="scriptFile" type="soapenc:string"/>
    <wsdl:part name="userPassword" type="soapenc:string"/>
    <wsdl:part name="lang" type="soapenc:string"/>
  </wsdl:message>
  <wsdl:message name="createDomainResponse">
  </wsdl:message>
  <wsdl:portType name="EdiwinWS">
    <wsdl:operation name="getInterfaces">
      <wsdl:input message="impl:getInterfacesRequest"
name="getInterfacesRequest"/>
      <wsdl:output message="impl:getInterfacesResponse"
name="getInterfacesResponse"/>
      <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="registerSession" parameterOrder="user password domain
clientType group">
      <wsdl:input message="impl:registerSessionRequest"
name="registerSessionRequest"/>
      <wsdl:output message="impl:registerSessionResponse"
name="registerSessionResponse"/>
      <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="endSession">
      <wsdl:input message="impl:endSessionRequest" name="endSessionRequest"/>
      <wsdl:output message="impl:endSessionResponse"
name="endSessionResponse"/>
      <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="getDocuments" parameterOrder="filter type
documentTypeList idVolume">
      <wsdl:input message="impl:getDocumentsRequest"
name="getDocumentsRequest"/>
      <wsdl:output message="impl:getDocumentsResponse"
name="getDocumentsResponse"/>
      <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="getDocumentsB64" parameterOrder="filter type
documentTypeList idVolume">
      <wsdl:input message="impl:getDocumentsB64Request"
name="getDocumentsB64Request"/>
      <wsdl:output message="impl:getDocumentsB64Response"
name="getDocumentsB64Response"/>
      <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
  </wsdl:portType>

```

```

    <wsdl:operation name="publishDocument" parameterOrder="dataHandler
publishType process">
    <wsdl:input message="impl:publishDocumentRequest"
name="publishDocumentRequest"/>
    <wsdl:output message="impl:publishDocumentResponse"
name="publishDocumentResponse"/>
    <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="publishDocumentB64" parameterOrder="message
publishType process">
    <wsdl:input message="impl:publishDocumentB64Request"
name="publishDocumentB64Request"/>
    <wsdl:output message="impl:publishDocumentB64Response"
name="publishDocumentB64Response"/>
    <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="complementFunction" parameterOrder="complementName
complementFunction complementParamEnt">
    <wsdl:input message="impl:complementFunctionRequest"
name="complementFunctionRequest"/>
    <wsdl:output message="impl:complementFunctionResponse"
name="complementFunctionResponse"/>
    <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="exportDocuments" parameterOrder="filter type process
idVolume fileName showDate oneFileDoc genUnb genControl">
    <wsdl:input message="impl:exportDocumentsRequest"
name="exportDocumentsRequest"/>
    <wsdl:output message="impl:exportDocumentsResponse"
name="exportDocumentsResponse"/>
    <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="exportDocumentsB64" parameterOrder="filter type
process idVolume fileName showDate oneFileDoc genUnb genControl">
    <wsdl:input message="impl:exportDocumentsB64Request"
name="exportDocumentsB64Request"/>
    <wsdl:output message="impl:exportDocumentsB64Response"
name="exportDocumentsB64Response"/>
    <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="changeDocSituation" parameterOrder="filter sourceSit
destinationSit">
    <wsdl:input message="impl:changeDocSituationRequest"
name="changeDocSituationRequest"/>
    <wsdl:output message="impl:changeDocSituationResponse"
name="changeDocSituationResponse"/>
    <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="changeExtendedFieldValue" parameterOrder="filter
fieldNames fieldValues">
    <wsdl:input message="impl:changeExtendedFieldValueRequest"
name="changeExtendedFieldValueRequest"/>
    <wsdl:output message="impl:changeExtendedFieldValueResponse"
name="changeExtendedFieldValueResponse"/>
    <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="getStatistics" parameterOrder="statistic">
    <wsdl:input message="impl:getStatisticsRequest"
name="getStatisticsRequest"/>
    <wsdl:output message="impl:getStatisticsResponse"
name="getStatisticsResponse"/>
    <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="getInterlocutors" parameterOrder="codInterlocutor
interlocutorType onlyPhysics">

```



```

        <wsdl:input message="impl:getInterlocutorsRequest"
name="getInterlocutorsRequest"/>
        <wsdl:output message="impl:getInterlocutorsResponse"
name="getInterlocutorsResponse"/>
        <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
    <wsdl:operation name="createDomain" parameterOrder="domain scriptFile
userPassword lang">
        <wsdl:input message="impl:createDomainRequest"
name="createDomainRequest"/>
        <wsdl:output message="impl:createDomainResponse"
name="createDomainResponse"/>
        <wsdl:fault message="impl:EdiwinWSEException" name="EdiwinWSEException"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="EdiwinWSSoapBinding" type="impl:EdiwinWS">
    <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getInterfaces">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getInterfacesRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getInterfacesResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="registerSession">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="registerSessionRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="registerSessionResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="endSession">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="endSessionRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="endSessionResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>

```



```

        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getDocuments">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getDocumentsRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getDocumentsResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getDocumentsB64">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getDocumentsB64Request">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getDocumentsB64Response">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="publishDocument">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="publishDocumentRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="publishDocumentResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="publishDocumentB64">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="publishDocumentB64Request">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>

```

```

        <wsdl:output name="publishDocumentB64Response">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="complementFunction">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="complementFunctionRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="complementFunctionResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="exportDocuments">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="exportDocumentsRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="exportDocumentsResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="exportDocumentsB64">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="exportDocumentsB64Request">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="exportDocumentsB64Response">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="changeDocSituation">
        <wsdlsoap:operation soapAction=""/>

```

```

        <wsdl:input name="changeDocSituationRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="changeDocSituationResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="changeExtendedFieldValue">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="changeExtendedFieldValueRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="changeExtendedFieldValueResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getStatistics">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getStatisticsRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getStatisticsResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getInterlocutors">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getInterlocutorsRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getInterlocutorsResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">

```

```

        <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
    </wsdl:operation>
    <wsdl:operation name="createDomain">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="createDomainRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="createDomainResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinWSEException">
            <wsdlsoap:fault
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="EdiwinWSEException" namespace="urn:EdiwinWS" use="encoded"/>
        </wsdl:fault>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="EdiwinWSService">
    <wsdl:port binding="impl:EdiwinWSSoapBinding" name="EdiwinWS">
        <wsdlsoap:address
location="http://localhost:8081/EdiwinWS/services/EdiwinWS"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Para acceder a la última versión de la WSDL a través de Internet:

<https://web.sedeb2b.com/EdiwinWS/services/EdiwinWS?wsdl>