



Web Services EdiwinIntegration

Guía de usuario

Título del documento:	Web Services EdiwinIntegration
Nombre del fichero:	ES WebService EdiwinIntegration Guía de usuario.odt
Versión:	1.1
Estado:	VIGENTE
Fecha:	02/06/2015
Autor:	Oscar Albert Arcas

Revisión, Aprobación

Revisado por:	EDICOM R+D Department.	Fecha: 08/06/15
Aprobado por:	EDICOM R+D Department.	Fecha: 08/06/15

Historial de cambios

Versión	Fecha	Descripción de la acción	Páginas
1.0	02/06/15	Versión inicial WS EdiwinIntegration	Todas
1.1	19/06/2015	Añado parámetro grupo a todas las funciones del webservice. Añado el parámetro operación al objeto Param. Actualizo wsdl con los nuevos valores.	5, 7-8, 10-12, 14-16, 18-24

Índice de contenido

1	WEBSERVICE EdiwinIntegration	4
1.1	Introducción	4
1.1.1	Acceso al servicio Webservice EdiwinIntegration de Edicom	4
1.1.2	Funciones Del Webservice EdiwinIntegration de Edicom	4
2	Funciones	5
2.1	Publicación de documentos	5
2.1.1	Llamada:	5
2.1.2	Parámetros:	5
2.1.3	Requerimientos	6
2.1.4	Respuesta:	7
2.1.5	Ejemplos	7
2.2	Obtención del documento	10
2.2.1	Llamada:	10
2.2.2	Parámetros:	10
2.2.3	Respuesta	11
2.2.4	Ejemplos	11
2.3	Obtención de datos personalizados	14
2.3.1	Llamada:	14
2.3.2	Parámetros:	14
2.3.3	Requerimientos	14
2.3.4	Respuesta:	14
3	Lista de Códigos de Error	18
3.1	Introducción	18
3.1.1	Códigos genéricos definidos por EDICOM	18
3.1.2	Códigos específicos para el dominio	18
3.1.3	Ejemplo de una respuesta de error	18
3.2	WSDL WEBSERVICE EdiwinIntegration	19

1 WEBSERVICE EDIWININTEGRATION

1.1 INTRODUCCIÓN

EdiwinIntegration es la plataforma webservice de EDICOM que pueden utilizar los clientes para integración de documentos con sus dominios. Esta plataforma implementa las funcionalidades más comunes que se suelen hacer en integraciones de documentos.

Este documento ofrece la WSDL del servicio y la información necesaria para que el cliente pueda interactuar con el servicio EdiwinIntegration y desarrollar aplicaciones compatibles con el servicio de integración.

1.1.1 ACCESO AL SERVICIO WEBSERVICE EDIWININTEGRATION DE EDICOM

La URL de acceso al webservice EdiwinIntegration de EDICOM es la siguiente:

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

La implementación del servicio se realiza a través del protocolo https. De esta forma, se utiliza un canal de comunicaciones seguro con el servicio de integración de documentos de EDICOM para poder operar de forma segura.

1.1.2 FUNCIONES DEL WEBSERVICE EDIWININTEGRATION DE EDICOM

El Webservice EdiwinIntegration de Edicom permite acceder a los dominios de Edicom. Las funciones disponibles son:

Función	Descripción	ver.
publishDocument	Publicación de documentos. Permite publicar documentos dentro del dominio.	1.0
getDocument	Obtención de documentos. Permite obtener el o los documentos buscados dentro del dominio en distintos formatos.	1.0
getCustomData	Obtención de datos personalizados. Permite ejecutar funciones personalizadas dentro del dominio para obtener los resultados personalizados en función del mapa utilizado.	1.0

2 FUNCIONES

2.1 PUBLICACIÓN DE DOCUMENTOS.

`publishDocument`

Permite publicar documentos en el dominio.

Para poder implementar la llamada, el usuario debe autenticarse como usuario del servicio del dominio y especificar el documento que quiere importar.

2.1.1 LLAMADA:

```
public byte[] publishDocument (String user, String password,
String domain, String group, int publishType, byte[] message,
String process, boolean saveOriginalMessage, boolean
sendDocument, boolean returnData, int returnDataType, String
returnDataProcess) throws EdiwinIntegrationException
```

2.1.2 PARÁMETROS:

- **user:** Usuario que va realizar la conexión.
- **password:** Contraseña del usuario que va realizar la conexión.
- **domain:** Dominio de Ediwin en el que se va a realizar la conexión.
- **group:** Grupo donde se encuentra el dominio de Ediwin. Este parámetro es opcional.
- **publishType:** Tipo de publicación del documento.
 - 1: Documento de entrada.
 - 2: Documento de salida.
- **message:** Fichero con los datos del documentos a importar. Se trata del propio fichero que se quiere importar al dominio o los datos iniciales con los que definitivamente se generará el documento final que se importará en el dominio. Este fichero estará comprimido en formato ZIP. Dentro del mensaje SOAP, se enviarán los datos del fichero ZIP codificados en Base64 dentro del sobre de la llamada SOAP.
- **process:** Proceso de transformación que se aplicará al documento de entrada para generar el documento final que se importará.
- **saveOriginalMessage:** Permite guardar el documento original en entrada en caso de que se haga una transformación del documento original especificada por el parámetro process.
- **sendDocument:** Indica si el documento hay que enviarlo o no una vez publicado.
- **returnData:** Indica si hay que devolver un fichero como resultado de la ejecución de la publicación. Por defecto, si el parámetro es true, devolverá el listado del documento.
- **returnDataType:** Indica el tipo de fichero que se devuelve por defecto.:

- 1: Información del documento
- 2: Documento
- 3: Intercambio del documento
- 4: Listado del documento
- 5: Información auxiliar del documento
- **returnDataProcess:** En caso de que lo que se quiera devolver sea una transformación del documento original, indicaremos aquí el proceso que hay que ejecutar para hacer la transformación. Se utiliza para especificar el proceso de transformación que hay que hacer para devolver los ficheros deseados.

Proceso que contiene el servicio

- Verifica que los datos enviados al servicio son correctos.
- Verifica que los datos de acceso al dominio indicado son correctos.
- Si se ha solicitado una transformación de los datos, se permite la publicación en entrada el documento original para guardar traza del documento original.
- Publica el documento en entrada o salida según especifique el parámetro publishType. Se publica el documento si process es nulo o blanco, y si no, se publica la transformación del documento según ese proceso.
- Si el parámetro sendDocument es true, se deposita el documento.
- Si el parámetro returnData es false, el proceso termina devolviendo una respuesta vacía.
- Si el parámetro returnData es true, y el parámetro returnDataProcess es nulo o blanco, el proceso devolverá la parte del documento especificada en la variable returnDataType.
- Si el parámetro returnData es true, y el parámetro returnDataProcess es un proceso, el proceso devolverá la transformación del documento importado según el proceso indicado.

2.1.3 REQUERIMIENTOS

La ejecución de las publicaciones permite recuperar una serie de errores personalizados. Estos errores tienen que devolverse a partir de validadores personalizados que se ejecutarán a partir de funciones de la guía que ejecutarán un validador.

Para que este validador se pueda ejecutar, el interfaz destino ha de ser un XML obligatoriamente.

Si se quiere hacer una validación del fichero original antes de hacer la transformación para obtener el comprobante final, el interfaz del documento origen tiene que ser también un XML para poder ejecutar el validador desde las funciones de la guía del interfaz origen. Si se tiene cualquier otro tipo de interfaz intermedio, no se podrá utilizar un validador para el fichero origen y las validaciones se harán exclusivamente sobre el comprobante final.

Para recuperar estos errores, las respuestas tendrán que venir codificadas por un código interno y un text para que se puedan formatear como un error de la forma "codigo: texto". Por ejemplo el error del validador tendría que ser "1001: Error del documento".

2.1.4 RESPUESTA:

El resultado de la función dependerá de lo que contenga el parámetro `returnData` que se envía al webservice. En caso de que el parámetro `returnData` sea `false`, no se devolverá nada, simplemente una respuesta sin datos. En caso de que el parámetro `returnData` sea `true`, devolverá un fichero zip que dentro contendrá el documento especificado. Este documento puede ser el propio documento, el listado, información del propio documento, o lo que se necesite en función del mapa que se ejecute sobre el documento.

- **Respuesta satisfactoria.** Una vez acabado todo el proceso, el webservice devolverá una respuesta que podrá estar vacía o contener un fichero en formato zip con el resultado de la respuesta seleccionada.
- **Respuesta con error.** En caso de que se produzca algún error durante el proceso de publicación, el webservice devolverá una excepción con información acerca del error que se haya producido. El error tendrá un código de error y un texto genérico del servicio, y a continuación una serie de errores específicos del dominio en concreto. Ejemplos de este tipo de excepciones genéricas sería por ejemplo que el usuario no se haya autenticado correctamente en el sistema, o que no se haya enviado el fichero ZIP con el documento.

2.1.5 EJEMPLOS

Código del cliente:

```
public byte[] publishDocument(String user, String password,
String domain, String group, int publishType, byte[] message,
String process, boolean saveOriginalMessage, boolean
sendDocument, boolean returnData, int returnDataType, String
returnDataProcess) {
    byte[] data = null;
    try {
        data = ediwinIntegrationService.publishDocument(user,
password, domain, group, publishType, message, process,
saveOriginalMessage, sendDocument, returnData, returnDataType,
returnDataProcess);
    } catch (EdiwinIntegrationException eie) {
        eie.printStackTrace();
        System.out.println("EdiwinIntegrationException: " +
eie.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:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ediw="http://ediwinIntegration.service.ediwinws.edicom.com"
">
  <soapenv:Header/>
  <soapenv:Body>
    <ediw:publishDocument>
      <ediw:user>xxxxxxx</ediw:user>
      <ediw:password>_xxxxxxx</ediw:password>
      <ediw:domain>xxxxxxx</ediw:domain>
      <ediw:group>xxxxxxx</ediw:group>
      <ediw:publishType>2</ediw:publishType>

      <ediw:message>UESDBBQAAAAIAOJkmD0CMccqxQcAAFUSAAAFAAAAc2luX3RpbW
      JyYXJfX1hN+NAGoBzSJWgV4cbALEKkAC6Q/8Kf6O+2MupJ4ZzWI7/OoYRaCSKrOy
      Th50VeZvfwxFjp2jpk2r8m1CToky4Ti037j1s2nVxDhFECTuyJIRJFHhffVN/7vz
      p7RsO68MogkmV2UUem8T2VEmmJnWFeMVflqhZCZYEDV
      [...]
      C2Qy59xh/9sq4/ZB2RfEvjSuv3+L1BLAQIUCxQAAAAIAOJkmD0CMccqxQcAAFUSA
      AAFAAAAAAAAAAEIAAAAAAAAAABzaW5fdGltYnJhc19fWE1MSWlwXzQ3MzIwOS54b
      Wx+UESFBgAAAAABAAEATQAAAAIIAAAAAAAA==</ediw:message>
      <ediw:process></ediw:process>
    <ediw:saveOriginalMessage>true</ediw:saveOriginalMessage>
      <ediw:sendDocument>false</ediw:sendDocument>
      <ediw:returnData>false</ediw:returnData>
      <ediw:returnDataType>4</ediw:returnDataType>
      <ediw:returnDataProcess></ediw:returnDataProcess>
    </ediw:publishDocument>
  </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 de un **envío satisfactorio** sin datos de respuesta.

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
    <publishDocumentResponse
      xmlns="http://ediwinIntegration.service.ediwinws.edicom.com">
      <publishDocumentReturn xsi:nil="true"/>
    </publishDocumentResponse>
```



```

    </soapenv:Body>
  </soapenv:Envelope>

```

El siguiente ejemplo es una respuesta de un **envío satisfactorio** con datos de respuesta.

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
    <publishDocumentResponse
      xmlns="http://ediwinIntegration.service.ediwinws.edicom.com">

      <publishDocumentReturn>UESDBBQACAAIABaGZkAAAAAAAAAAAAAAAAAAAAAY
      29tcG9zZXIucGRm3LvZttrrakjd4zxjnHZaCIiAqrYKKIp0tNtj3iCAISqOezPNdf
      K9YF/UGVS9SrP9eZ+99qnJnZdb4rmpdAL0JmL+IGREzpiNWbiKrOFGi/wbk/o//8
      3/73/8G/A0gvipf4eX+N+Dr66vR+Pr1Tv/KUmKb7zD
      [...]
      dMd8lN8wvT+iPpF/D4wDodI1Y/eIVqBe7j/8eK9yFz2i905Xb5rTAGSBW8Yjj/11
      DvkT1BLBwgT+YmIPjsAADVUAABQSwECFAAUAAgACAAWhmZAE/mJiD47AAA1VAAAD
      AAAAAAAAAAAAAAAAAAAAAAY29tcG9zZXIucGRmUESFBgAAAAABAAEAOgAAAHg7A
      AAAAA=
    </publishDocumentReturn>
  </publishDocumentResponse>
</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 un **envío con errores**.

```

<?xml version="1.0" encoding="UTF-8"?>

<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>Errores del documento <Id:140085>
      <Referencia:BGM_1004> <Origen:> <Destino:>
      <Fichero:>~Tipo de error: Error en los interlocutores~POS=No
      existe el Punto Operacional Origen [3791000100001]~POS=No existe
      el Punto Operacional Destino [3791000100162]</faultstring>
      <detail>
        <ns1:fault
          xmlns:ns1="http://ediwinIntegration.service.ediwinws.edicom.com"
          >
          <ns1:cod>131</ns1:cod>
          <ns1:errors>
            <ns1:errors>
              <ns1:cod>100</ns1:cod>

```

```

        <ns1:text>LA VERSIÓN DEL COMPROBANTE NO ES
VÁLIDA [3.0]</ns1:text>
    </ns1:errors>
</ns1:errors>
    <ns1:text>Errores del documento <Id:140085>
<Referencia:BGM_1004> <Origen:> <Destino:>
<Fichero:>~Tipo de error: Error en los interlocutores~POS=No
existe el Punto Operacional Origen [3791000100001]~POS=No existe
el Punto Operacional Destino [3791000100162]</ns1:text>
</ns1:fault>
    <ns2:hostname
xmlns:ns2="http://xml.apache.org/axis/">web.sedeb2b.com</ns2:hos
tname>
</detail>
</soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>

```

Nota: En este ejemplo concreto se presenta un error 131 indicando que el documento tiene errores de puntos operacionales desconocidos.

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

2.2 OBTENCIÓN DEL DOCUMENTO.

```
getDocument
```

Permite obtener el documento que se especifique con los parámetros especificados en la llamada a la función del webservice.

Para poder implementar la llamada, el usuario debe autenticarse como usuario del dominio y especificar el documento que quiere exportar y el tipo de exportación.

2.2.1 LLAMADA:

```

public byte[] getDocument(String user, String password, String
domain, int exportType, int idVolume, Param[] parameters) throws
EdiwinIntegrationException

```

2.2.2 PARÁMETROS:

- **user:** Usuario que va realizar la conexión.
- **password:** Contraseña del usuario que va realizar la conexión.
- **domain:** Dominio de Ediwin en el que se va a realizar la conexión.
- **group:** Grupo donde se encuentra el dominio de Ediwin. Este parámetro es opcional.
- **exportType:** Tipo de exportación del documento.
 - 1: Información extendida
 - 2: Documento

- 3: Intercambio
- 4: Informe
- 5: Información auxiliar
- **parameters:** Son un conjunto de datos con los que se hará referencia al documento que se quiere exportar. Consisten en una secuencia de datos con formato campo/operador/valor, a través del cual se harán el filtro de búsqueda de los documentos. Los posibles operadores son los siguientes:
 - = (Igual a)
 - <> (Distinto a, se debe indicar "<>" en la petición)
 - > (Mayor qué, se debe indicar ">" en la petición)
 - < (Menor qué, se debe indicar "<" en la petición)
 - >= (Mayor o igual qué, se debe indicar ">=" en la petición)
 - <= (Menor o igual qué, se debe indicar "<=" en la petición)
 - IN (El valor es alguno de una lista de valores separada por comas.)
 - NOT IN (El valor no está en la lista de valores separada por comas.)
 - IS NULL (El valor es Null, No es necesario indicar "valor")
 - IS NOT NULL (El valor no es Null, No es necesario indicar "valor")
 - LIKE (Busca un patrón determinado. Se debe usar %.)
 - NOT LIKE (Excluye un patrón determinado. Se debe usar %.)

2.2.3 RESPUESTA

El resultado de la función será un fichero comprimido en formato ZIP con el resultado de la exportación del documento especificado por la llamada del webservice.

- **Respuesta satisfactoria:** En caso de que se haya realizado la operación, el servicio devolverá un documento comprimido en formato ZIP con el resultado de la exportación del documento.
- **Respuesta con errores:** En caso de que se produzca algún error durante el proceso de obtención del documento, el webservice devolverá una excepción con información acerca del error que se haya producido. El error tendrá un código genérico de error y un texto explicativo del error producido. Ejemplos de este tipo de excepciones sería por ejemplo que el usuario no se haya autenticado correctamente en el sistema, o que no se haya obtenido ningún documento con los parámetros enviados.

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

2.2.4 EJEMPLOS

Código del cliente:

```

public byte[] getDocument(String user, String password, String
domain, String group, int exportType, Param[] parameters) {
    byte[] data = null;
    try {
        data = ediwinIntegrationService.getDocument(user,
password, domain, group, exportType, parameters);
    } catch ( EdiwinIntegrationException eie) {
        eie.printStackTrace();
        System.out.println("EdiwinIntegrationException: " +
eie.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:

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ediw="http://ediwinIntegration.service.ediwinws.edicom.com
">
    <soapenv:Header/>
    <soapenv:Body>
        <ediw:getDocument>
            <ediw:user>xxxxxx</tic:user>
            <ediw:password>xxxxxx</tic:password>
            <ediw:domain>xxxxxx</tic:domain>
            <ediw:group>xxxxxx</tic:group>
            <ediw:exportType>2</tic:exportType>
            <!--1 or more repetitions:-->
            <ediw:parameters>
                <ediw:name>id</tic:name>
                <ediw:operator>=</tic:operator>
                <ediw:value>23456</tic:value>
            </ediw:parameters>
            <ediw:parameters>
                <ediw:name>REFERENCIA</tic:name>
                <ediw:operator>LIKE</tic:operator>
                <ediw:value>123456%</tic:value>
            </ediw:parameters>
            <ediw:parameters>
                <ediw:name>GUIA</tic:name>
                <ediw:operator>IS NOT NULL</tic:operator>
            </ediw:parameters>
        </ediw:getDocument>
    </soapenv:Body>
</soapenv:Envelope>

```

```

        <ediw:value></tic:value>
    </ediw:parameters>
    <ediw:parameters>
        <ediw:name>FECHACAMBIOESTADO</tic:name>
        <ediw:operator>&lt;=</tic:operator>
        <ediw:value>2015/06/02</tic:value>
    </ediw:parameters>
</ediw:getDocument>
</soapenv:Body>
</soapenv:Envelope>

```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de un **envío satisfactorio** con datos de respuesta.

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
        <getDocumentResponse
xmlns="http://ediwinIntegration.service.ediwinws.edicom.com">

            <getDocumentReturn>UESDBBQACAAIABaGZkAAAAAAAAAAAAAAAAAMAAAY29tc
G9zZXIucGRm3LvZtrrakjd4zxjnhZaCIiAqrYKKIp0tNtj3iCAISqOezPNdfK9YF
/UGVS9SrP9eZ+99qnJnZdb4rmpdAL0JmL+IGREzpiNWbiKrOFGi/wbk/o//83/73
/8G/A0gvipf4eX+N+Dr66vR+Pr1Tv/KUmKb7zD
[...]
```

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 **con errores**.

```

<?xml version="1.0" encoding="UTF-8"?>
<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>El tipo de exportación 6 no es
válido.</faultstring>
        <detail>
            <ns1:fault
xmlns:ns1="http://ediwinIntegration.service.ediwinws.edicom.com"
>
                <ns1:cod>120</ns1:cod>
                <ns1:errors xsi:nil="true"/>
                <ns1:text>El tipo de exportación 6 no es
válido.</ns1:text>
            </ns1:fault>
            <ns2:hostname
xmlns:ns2="http://xml.apache.org/axis/">web.sedeb2b.com</ns2:hos
tname>
        </detail>
    </soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>

```

2.3 OBTENCIÓN DE DATOS PERSONALIZADOS.

```
getCustomData
```

Permite obtener datos personalizados de la explotación del dominio de Ediwin.

Para poder implementar la llamada, el usuario debe autenticarse como usuario del dominio y especificar la función personalizada que se quiere ejecutar junto con los parámetros necesarios para su ejecución.

2.3.1 LLAMADA:

```

public byte[] getCustomData(String user, String password, String
domain, String group, String process, Param[] parameters) throws
EdiwinIntegrationException

```

2.3.2 PARÁMETROS:

- **user:** Usuario que va realizar la conexión.
- **password:** Contraseña del usuario que va realizar la conexión.
- **domain:** Dominio de Ediwin en el que se va a realizar la conexión.
- **group:** Grupo donde se encuentra el dominio de Ediwin. Este parámetro es opcional.
- **process:** Mapa personalizada por dominio que se quiere ejecutar. Cada dominio tendrá sus mapas específicos.
- **parameters:** Son un conjunto de datos con los que se hará referencia a los parámetros de ejecución del mapa. Consisten en una secuencia de datos con formato campo/operacion/valor. La operación se ignora y se toma como un "=" ya que en esta función siempre se hace una asignación directa del valor del campo. El operador podría no incluirse en la petición.

2.3.3 REQUERIMIENTOS

Para poder ejecutar el mapa personalizado, el mapa espera un fichero con una máscara determinada como fichero origen para ejecutar el proceso.

Para unificar criterios y no pasar esta máscara dentro de los datos del webservice, el propio webservice define este fichero origen como un fichero en blanco con el nombre "fichero.txt".

Por este motivo, todos aquellos mapas personalizados deben tener como origen un interfaz que lea un fichero en blanco con el nombre "fichero.txt".

2.3.4 RESPUESTA:

El resultado de la función será un fichero comprimido en formato ZIP con el resultado de la ejecución de la función de la llamada del webservice.

- **Respuesta satisfactoria:** En caso de que se haya realizado la operación, el servicio devolverá un documento comprimido en formato ZIP con el resultado de la ejecución de la función.
- **Respuesta con errores:** En caso de que se produzca algún error durante el proceso de ejecución de la función, el webservice devolverá una excepción con información acerca del error que se haya producido. El error tendrá un código genérico de error y un texto explicativo del error producido. Ejemplos de este tipo de excepciones sería por ejemplo que el usuario no se haya autenticado correctamente en el sistema, o que no se haya obtenido ningún documento con los parámetros enviados.

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

Código del cliente:

```
public byte[] getCustom(String user, String password, String
domain, String group, String process, Param[] parameters) {
    byte[] data = null;
    try {
        data = ediwinIntegrationService.getCustomData(user,
password, domain, group, process, parameters);
    } catch (EdiwinIntegrationException eie) {
        eie.printStackTrace();
        System.out.println("EdiwinIntegrationException: " +
eie.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:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ediw="http://ediwinIntegration.service.ediwinws.edicom.com">
  <soapenv:Header/>
  <soapenv:Body>
    <tic:getCustomData>
      <ediw:user>xxxxxx</tic:user>
      <ediw:password>xxxxxx</tic:password>
      <ediw:domain>xxxxxx</tic:domain>
      <ediw:group>xxxxxx</tic:group>
      <ediw:process>process1</tic:process>
      <!--1 or more repetitions:-->
      <ediw:parameters>
        <ediw:name>id</tic:name>
        <ediw:operator>=</tic:operator>
        <ediw:value>23456</tic:value>
      </ediw:parameters>
    </ediw:getCustomData>
  </soapenv:Body>
</soapenv:Envelope>
```

Respuesta SOAP:

El siguiente ejemplo es una respuesta de un **envío satisfactorio**.

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
    <getCustomDataResponse
xmlns="http://ediwinIntegration.service.ediwinws.edicom.com">

      <getCustomDataReturn>UESDBBQACAAIABaGZkAAAAAAAAAAAAAAAAAAAAAAY29
tcG9zZXIucGRm3LvZtrrakjd4zxjnhZaCIiAqrYKKIp0tNtj3iCAISqOezPNdfK9
YF/UGVS9SrP9eZ+99qnJnZdb4rmpdAL0JmL+IGREzpiNWbiKrOFGi/wbk/o//83/
73/8G/A0gvipf4eX+N+Dr66vR+Pr1Tv/KUMKb7zD
[...]
```

```
dMd81N8wvT+iPpF/D4wDodI1Y/eIVqBe7j/8eK9yFz2i905Xb5rTAGSBW8Yjj/11
DvkT1BLBwgT+YmIPjsAADVUAABQSwECFAAUAAgACAAWhmZAE/mJiD47AAA1VAAAD
AAAAAAAAAAAAAAAAAAAAAAY29tcG9zZXIucGRmUESFBgAAAAABAAEAOgAAAHg7A
AAAAA==</getCustomDataReturn>
    </getCustomDataResponse>
  </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 **con errores**.

```
<?xml version="1.0" encoding="UTF-8"?>
<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>El parámetro process no puede estar
vacío.</faultstring>
      <detail>
        <ns1:fault
xmlns:ns1="http://ediwinIntegration.service.ediwinws.edicom.com"
>
          <ns1:cod>120</ns1:cod>
          <ns1:errors xsi:nil="true"/>
          <ns1:text>El parámetro process no puede estar
vacío.</ns1:text>
        </ns1:fault>
        <ns2:hostname
xmlns:ns2="http://xml.apache.org/axis/">web.sedeb2b.com</ns2:hos
tname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

3 LISTA DE CÓDIGOS DE ERROR

3.1 INTRODUCCIÓN

El servicio de Integración de Ediwin tiene códigos de error genéricos y específicos de cada dominio.

3.1.1 CÓDIGOS GENÉRICOS DEFINIDOS POR EDICOM

Estos códigos han sido definidos como códigos de error genéricos por Edicom. Se indican también los posibles escenarios en que se presentan estos errores.

Código	Error
100	Error genérico del servicio de Integración de Ediwin.
110	Error de autenticación.
120	Error de datos.
130	Error al publicar documentos.
131	Error al publicar documento origen.
140	Error al exportar documentos.
150	Error al obtener datos personalizados.

3.1.2 CÓDIGOS ESPECÍFICOS PARA EL DOMINIO.

Cada dominio puede definir ciertos tipos de códigos de errores y aplicarlos en las distintas ejecuciones de los procesos. De esta forma, cada dominio puede configurarse sus propios códigos de error específicos.

3.1.3 EJEMPLO DE UNA RESPUESTA DE ERROR

```
<?xml version="1.0" encoding="UTF-8"?>
<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>El parámetro process no puede estar
vacío.</faultstring>
      <detail>
        <ns1:fault
xmlns:ns1="http://ediwinIntegration.service.ediwinws.edicom.com"
>
          <ns1:cod>120</ns1:cod>
          <ns1:errors xsi:nil="true"/>
          <ns1:text>El parámetro process no puede estar
vacío.</ns1:text>
        </ns1:fault>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <ns2:hostname
xmlns:ns2="http://xml.apache.org/axis/">web.sedeb2b.com</ns2:hos
tname>
    </detail>
</soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>

```

3.2 WSDL WEBSERVICE EDIWININTEGRATION

El contenido del wsdl del servicio es el siguiente:

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
targetNamespace="http://ediwinIntegration.service.ediwinws.edicom.com"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://ediwinIntegration.service.ediwinws.edicom.com"
xmlns:intf="http://ediwinIntegration.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 elementFormDefault="qualified"
targetNamespace="http://ediwinIntegration.service.ediwinws.edicom.com"
xmlns="http://www.w3.org/2001/XMLSchema">
            <element name="getDocument">
                <complexType>
                    <sequence>
                        <element name="user" type="xsd:string"/>
                        <element name="password" type="xsd:string"/>
                        <element name="domain" type="xsd:string"/>
                        <element name="group" type="xsd:string"/>
                        <element name="exportType" type="xsd:int"/>
                        <element name="idVolume" type="xsd:int"/>
                        <element maxOccurs="unbounded" name="parameters"
type="impl:Param"/>
                    </sequence>
                </complexType>
            </element>
            <complexType name="Param">
                <sequence>
                    <element name="name" nillable="true" type="xsd:string"/>
                    <element name="operator" nillable="true"
type="xsd:string"/>
                    <element name="value" nillable="true" type="xsd:string"/>
                </sequence>
            </complexType>
            <element name="getDocumentResponse">
                <complexType>
                    <sequence>

```

```

        <element name="getDocumentReturn"
type="xsd:base64Binary"/>
    </sequence>
</complexType>
</element>
<complexType name="Error">
    <sequence>
        <element name="cod" type="xsd:int"/>
        <element name="text" nillable="true" type="xsd:string"/>
    </sequence>
</complexType>
<complexType name="ArrayOfError">
    <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="item"
type="impl:Error"/>
    </sequence>
</complexType>
<complexType name="EdiwinIntegrationException">
    <sequence>
        <element name="cod" type="xsd:int"/>
        <element name="errors" nillable="true"
type="impl:ArrayOfError"/>
        <element name="text" nillable="true" type="xsd:string"/>
    </sequence>
</complexType>
<element name="fault"
type="impl:EdiwinIntegrationException"/>
<element name="getCustomData">
    <complexType>
        <sequence>
            <element name="user" type="xsd:string"/>
            <element name="password" type="xsd:string"/>
            <element name="domain" type="xsd:string"/>
            <element name="group" type="xsd:string"/>
            <element name="process" type="xsd:string"/>
            <element maxOccurs="unbounded" name="parameters"
type="impl:Param"/>
        </sequence>
    </complexType>
</element>
<element name="getCustomDataResponse">
    <complexType>
        <sequence>
            <element name="getCustomDataReturn"
type="xsd:base64Binary"/>
        </sequence>
    </complexType>
</element>
<element name="publishDocument">

```

```

<complexType>
  <sequence>
    <element name="user" type="xsd:string"/>
    <element name="password" type="xsd:string"/>
    <element name="domain" type="xsd:string"/>
    <element name="group" type="xsd:string"/>
    <element name="publishType" type="xsd:int"/>
    <element name="message" type="xsd:base64Binary"/>
    <element name="process" type="xsd:string"/>
    <element name="saveOriginalMessage" type="xsd:boolean"/>
    <element name="sendDocument" type="xsd:boolean"/>
    <element name="returnData" type="xsd:boolean"/>
    <element name="returnDataType" type="xsd:int"/>
    <element name="returnDataProcess" type="xsd:string"/>
  </sequence>
</complexType>
</element>
<element name="publishDocumentResponse">
  <complexType>
    <sequence>
      <element name="publishDocumentReturn"
type="xsd:base64Binary"/>
    </sequence>
  </complexType>
</element>
</schema>
</wsdl:types>
<wsdl:message name="getCustomDataRequest">
  <wsdl:part element="impl:getCustomData" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="publishDocumentRequest">
  <wsdl:part element="impl:publishDocument"
name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="publishDocumentResponse">
  <wsdl:part element="impl:publishDocumentResponse"
name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getDocumentRequest">
  <wsdl:part element="impl:getDocument" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getCustomDataResponse">
  <wsdl:part element="impl:getCustomDataResponse"
name="parameters">
  </wsdl:part>

```

```

</wsdl:message>
<wsdl:message name="getDocumentResponse">
  <wsdl:part element="impl:getDocumentResponse"
name="parameters">
    </wsdl:part>
  </wsdl:message>
<wsdl:message name="EdiwinIntegrationException">
  <wsdl:part element="impl:fault" name="fault">
    </wsdl:part>
  </wsdl:message>
<wsdl:portType name="EdiwinIntegration">
  <wsdl:operation name="getDocument">
    <wsdl:input message="impl:getDocumentRequest"
name="getDocumentRequest">
      </wsdl:input>
    <wsdl:output message="impl:getDocumentResponse"
name="getDocumentResponse">
      </wsdl:output>
    <wsdl:fault message="impl:EdiwinIntegrationException"
name="EdiwinIntegrationException">
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getCustomData">
      <wsdl:input message="impl:getCustomDataRequest"
name="getCustomDataRequest">
        </wsdl:input>
      <wsdl:output message="impl:getCustomDataResponse"
name="getCustomDataResponse">
        </wsdl:output>
      <wsdl:fault message="impl:EdiwinIntegrationException"
name="EdiwinIntegrationException">
        </wsdl:fault>
      </wsdl:operation>
    <wsdl:operation name="publishDocument">
      <wsdl:input message="impl:publishDocumentRequest"
name="publishDocumentRequest">
        </wsdl:input>
      <wsdl:output message="impl:publishDocumentResponse"
name="publishDocumentResponse">
        </wsdl:output>
      <wsdl:fault message="impl:EdiwinIntegrationException"
name="EdiwinIntegrationException">
        </wsdl:fault>
      </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="EdiwinIntegrationSoapBinding"
type="impl:EdiwinIntegration">
      <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="getDocument">

```

```

        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getDocumentRequest">
            <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="getDocumentResponse">
            <wsdlsoap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinIntegrationException">
            <wsdlsoap:fault name="EdiwinIntegrationException"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getCustomData">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getCustomDataRequest">
            <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="getCustomDataResponse">
            <wsdlsoap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinIntegrationException">
            <wsdlsoap:fault name="EdiwinIntegrationException"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="publishDocument">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="publishDocumentRequest">
            <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="publishDocumentResponse">
            <wsdlsoap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="EdiwinIntegrationException">
            <wsdlsoap:fault name="EdiwinIntegrationException"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="EdiwinIntegrationService">
    <wsdl:port binding="impl:EdiwinIntegrationSoapBinding"
name="EdiwinIntegration">
        <wsdlsoap:address
location="https://web.sedeb2b.com/EdiwinWS/services/EdiwinIntegr
ation"/>
    </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/EdiwinIntegration?wsdl>