



Junta de Andalucía

CONSEJERÍA DE LA PRESIDENCIA,
ADMINISTRACIÓN PÚBLICA E INTERIOR

Manual de integración

Aplicación SelloWar v4.0

Versión: v01r00

Fecha: 09/06/2021



ÍNDICE

1	Introducción.....	4
1.1	Objeto.....	4
1.2	Alcance.....	4
1.3	Definición.....	4
2	Consideraciones generales.....	5
2.1	Definición de tipos de datos.....	5
2.1.1	Tipo SelloException.....	5
2.1.2	Tipo getParametrosSelloRequest.....	5
2.1.3	Tipo getDocumentosSelladoRequest.....	5
2.1.4	Tipo getParametrosSelloResponse.....	5
2.1.5	Tipo getDocumentoSelladoResponse.....	6
2.1.6	Tipo getSellosResponse.....	6
3	Servicios.....	7
3.1	Método getSellos.....	7
3.1.1	Ficha del servicio.....	7
3.1.2	Descripción funcional.....	7
3.1.3	Parámetro de salida.....	7
3.1.4	Listado de errores.....	7
3.1.5	Ejemplo de petición.....	7
3.1.6	Ejemplo de respuesta.....	8
3.2	Método getParametrosSello.....	8
3.2.1	Ficha del servicio.....	8
3.2.2	Descripción funcional.....	8
3.2.3	Parámetros de entrada.....	8
3.2.4	Parámetro de salida.....	9
3.2.5	Listado de errores.....	9
3.2.6	Ejemplo de petición.....	9
3.2.7	Ejemplo de respuesta.....	9
3.3	Método getDocumentoSellado.....	10



3.3.1 Ficha del servicio.....	10
3.3.2 Descripción funcional.....	10
3.3.3 Parámetros de entrada.....	10
3.3.4 Parámetro de salida.....	10
3.3.5 Listado de errores.....	10
3.3.6 Ejemplo de petición.....	11
4 Tipos de sellos.....	12
4.1 Decoradores.....	12
4.1.1 Decorador Registro de entrada.....	12
4.1.2 Decorador Registro de salida.....	13
5 Dependencias software.....	14
5.1 Descarga del proyecto SelloWar v4.0.....	14
6 Ejemplo aplicación.....	15
7 Resultados.....	17
7.1 Aplicación sello Registro de entrada.....	18
7.2 Aplicación sello Registro de salida.....	19
8 Glosario.....	20



1 Introducción

1.1 Objeto

Este documento es la guía de referencia de desarrollo para el consumo de los servicios web de la aplicación *SelloWar*. El objetivo del presente documento, es proporcionar una descripción técnica de las operaciones soportadas por los servicios web aquí descritos. Encontraremos una sección donde se describirán los aspectos comunes y después una descripción de las particularidades de cada uno de los servicios.

1.2 Alcance

El objetivo de la aplicación *SelloWar* es el de proveer servicios web para ser consumidos por aquellos sistema que requieran de la operación de estampar un sello en documentos PDF.

1.3 Definición

La aplicación *SelloWar* utiliza el protocolo de SOAP(Simple Object Access Protocol) para la comunicación entre sistemas que requieran de los servicios ofrecidos por el aplicativo.



2 Consideraciones generales

2.1 Definición de tipos de datos

A continuación se explican los distintos tipos de datos usados en los distintos servicios del aplicativo.

2.1.1 Tipo SelloException

Parámetro	Descripción	Tipo	Cardinalidad
descripcion	Descripción del error	String	1..1

2.1.2 Tipo getParametrosSelloRequest

Parámetro	Descripción	Tipo	Cardinalidad
nombreDecorador	Nombre del decorador del que se desea recoger los parámetros necesarios	String	1..1

2.1.3 Tipo getDocumentosSelladoRequest

Parámetro	Descripción	Tipo	Cardinalidad
documento	DocumentoDTO al que se le va a realizar el sellado	String	1..1
decoradoresUsados	Lista de DecoradorXSdto que se van a usar en el proceso de sellado.	List<DecoradorXSdto>	1..1

2.1.4 Tipo getParametrosSelloResponse

Parámetro	Descripción	Tipo	Cardinalidad
getParametrosSelloReturn	Array de String de los parámetros que se usan según el decorador que se utilice	String[]	1..1



2.1.5 Tipo getDocumentoSelladoResponse

Parámetro	Descripción	Tipo	Cardinalidad
getParametrosSelladoReturn	Array de Bytes correspondiente al documento sellado	byte[]	1..1

2.1.6 Tipo getSellosResponse

Parámetro	Descripción	Tipo	Cardinalidad
getSellosReturn	Array de String correspondiente a los sellos.	String[]	1..1



3 Servicios

3.1 Método getSellos

3.1.1 Ficha del servicio

Nombre del servicio	getSelloException
Signatura	<getSellosResponse> getSellos()
Resumen	Devuelve un Array de String con los sellos.
WSDL	http://servidor:puerto/sello-war/services/SellarWS?wsdl
Endpoint desarrollo	http://servidor:puerto/sello-war/services/SellarWS?wsdl

3.1.1 Descripción funcional

Método definido para devolver el array de String de todos los decoradores que implementa la aplicación.

3.1.2 Parámetro de salida

Nombre	Descripción	Tipo	Obligatoriedad
resultado	Array de String que corresponde con el nombre del decorador.	getSellosResponse	Sí

3.1.3 Listado de errores

Posible error:

- SelloException

3.1.4 Ejemplo de petición

```
<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:ser="http://servicioweb.sello.juntadeandalucia.es">
  <soapenv:Header />
  <soapenv:Body>
    <ser:getSellos
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </soapenv:Body>
  </soapenv:Envelope>
```



3.1.5 Ejemplo de respuesta

```

<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:getSellosResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://servicioweb.sello.juntadeandalucia.es">
      <getSellosReturn soapenc:arrayType="xsd:string[2]"
        xsi:type="soapenc:Array"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <getSellosReturn
          xsi:type="xsd:string">registroEntradav2</getSellosReturn>
        <getSellosReturn
          xsi:type="xsd:string">registroSalidav2</getSellosReturn>
        </ns1:getSellosResponse>
      </soapenv:Body>
    </soapenv:Envelope>
  
```

3.2 Método getParametrosSello

3.2.1 Ficha del servicio

Nombre del servicio	getParametrosSello
Signatura	<getParametrosSelloResponse> getParametrosSello (getParametrosSelloRequest)
Resumen	Recoge un String[] con los parametros que se usan según el decorador que se consulte.
WSDL	http://servidor:puerto/sello-war/services/SellarWS?wsdl
Endpoint desarrollo	http://servidor:puerto/sello-war/services/SellarWS?wsdl

3.2.2 Descripción funcional

Método definido para obtener los parámetros que se usan para el sellado según el decorador que se consulte.

3.2.3 Parámetros de entrada

Nombre	Descripción	Tipo	Obligatoriedad
nombreDecorador	Nombre del DecoradorXSDTO del que se quieren recoger los parámetros a usar	getParametrosSelloRequest	Sí



3.2.4 Parámetro de salida

Nombre	Descripción	Tipo	Obligatoriedad
resultado	Parametros devueltos correspondientes al decorador consultado.	getParametrosSelloResponde	Sí

3.2.5 Listado de errores

Posible error:

- SelloException

3.2.6 Ejemplo de petición

Ejemplo de petición SOAP para recuperar los parámetros necesarios del sello de registro de entrada:

```
<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:ser="http://servicioweb.sello.juntadeandalucia.es">
  <soapenv:Header />
  <soapenv:Body>
    <ser:getParametrosSello
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      <nombreDecorador xsi:type="xsd:string">registroEntrada2</nombreDecorador>
    </ser:getParametrosSello>
  </soapenv:Body>
</soapenv:Envelope>
```

3.2.7 Ejemplo de respuesta

```
<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:getParametrosSelloResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://servicioweb.sello.juntadeandalucia.es">
      <getParametrosSelloReturn
        soapenc:arrayType="xsd:string[3]" xsi:type="soapenc:Array"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <getParametrosSelloReturn
          xsi:type="xsd:string">numAriesEntrada</getParametrosSelloReturn>
        <getParametrosSelloReturn
          xsi:type="xsd:string">fechaAriesEntrada</getParametrosSelloReturn>
        <getParametrosSelloReturn
          xsi:type="xsd:string">horaAriesEntrada</getParametrosSelloReturn>
        </getParametrosSelloReturn>
      </ns1:getParametrosSelloResponse>
    </soapenv:Body>
  </soapenv:Envelope>
```



3.3 Método getDocumentoSellado

3.3.1 Ficha del servicio

Nombre del servicio	getDocumentoSellado
Signatura	<getDocumentoSelladoResponse> getDocumentoSellado (getDocumentoSelladoRequest)
Resumen	Devuelve el array de bytes del documento sellado con el decorador utilizado.
WSDL	http://servidor:puerto/sello-war/services/SellarWS?wsdl
Endpoint desarrollo	http://servidor:puerto/sello-war/services/SellarWS?wsdl

3.3.2 Descripción funcional

Método definido para devolver el array de bytes del resultado de aplicar el decorador indicado sobre el documento que se envía.

3.3.3 Parámetros de entrada

Nombre	Descripción	Tipo	Obligatoriedad
documento	DocumentoDTO al que se le va a realizar el sellado	getDocumentoSellado Request	Sí
decoradoresUsados	Lista de DecoradorXSdto que se van a usar en el proceso de sellado.	getDocumentoSellado Request	Sí

3.3.4 Parámetro de salida

Nombre	Descripción	Tipo	Obligatoriedad
resultado	Array de bytes que resultado de realizar el sellado al documento	getDocumentoSellado Response	Sí

3.3.5 Listado de errores

Posible error:

- SelloException



3.3.6 Ejemplo de petición

Ejemplo petición SOAP para la aplicación del sello de registro de entrada:

```
<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:ser="http://servicioweb.sello.juntadeandalucia.es"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <soapenv:Header />
  <soapenv:Body>
    <ser:getDocumentoSellado
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <documento xsi:type="dom:DocumentoDTO"
        xmlns:dom="http://domain.sello.juntadeandalucia.es">
        <contenido xsi:type="xsd:base64Binary">DOCUMENTO</contenido>
      </documento>
      <decoradoresUsados
        xsi:type="cli:ArrayOf_tns2_DecoradorDTO"
        soapenc:arrayType="dom:DecoradorDTO[1]"
        xmlns:cli="http://clienteSello"
        xmlns:dom="http://domain.sello.juntadeandalucia.es">
        <decorador>
          <nombre xsi:type="xsd:string">registroEntradaV2</nombre>
          <parametros
            xsi:type="cli:ArrayOf_tns2_ParametroDecoradorDTO"
            soapenc:arrayType="dom:ParametroDecoradorDTO[3]"
            xmlns:cli="http://clienteSello"
            xmlns:dom="http://domain.sello.juntadeandalucia.es">
            <parametro xsi:type="dom:ParametroDecoradorDTO">
              <nombreParametro
                xsi:type="xsd:string">numAriesEntrada</nombreParametro>
              <valorParametro
                xsi:type="xsd:string">202100000000000</valorParametro>
            </parametro>
            <parametro xsi:type="dom:ParametroDecoradorDTO">
              <nombreParametro
                xsi:type="xsd:string">fechaAriesEntrada</nombreParametro>
              <valorParametro
                xsi:type="xsd:string">17/02/2021</valorParametro>
            </parametro>
            <parametro xsi:type="dom:ParametroDecoradorDTO">
              <nombreParametro
                xsi:type="xsd:string">horaAriesEntrada</nombreParametro>
              <valorParametro
                xsi:type="xsd:string">00:00:00</valorParametro>
            </parametro>
          </parametros>
        </decorador>
      </decoradoresUsados>
    </ser:getDocumentoSellado>
  </soapenv:Body>
</soapenv:Envelope>
```



4 Tipos de sellos

Los distintos tipos de sellos de los que dispone la aplicación *SelloWar* se definen a través de decoradores. En función del decorador definido, se aplicará un sello u otro.

El texto que se estampa en los documentos tiene la siguiente apariencia:

- Fuente: Noto Sans
- Tamaño: 9 puntos
- Color: Negro

A continuación se explicarán cada uno de los decoradores y las especificaciones que hay que indicar para su aplicación.

4.1 Decoradores

La aplicación contiene los siguientes decoradores:

- Registro de entrada
- Registro de salida

Cada uno de los decoradores contiene información que se debe definir para la aplicación del sello en el documento. El resultado de la aplicación del sello es el array de bytes del documento sellado.

4.1.1 Decorador Registro de entrada

Para la aplicación del sello de registro de entrada es necesario definir el decorador *registroEntradaV2*. A este decorador se le tiene que definir los siguientes parámetros:

Parámetro	Descripción	Tipo	Obligatoriedad
numAriesEntrada	Número de registro de @ries	String	Sí
fechaAriesEntrada	Fecha de registro de @ries	String (dd/MM/yyyy)	Sí
horaAriesEntrada	Hora de registro de @ries	String (HH:mm:ss)	Sí

La posición de este sello es la siguiente:

- Localización en el lateral izquierdo del documento
- Separación de 0,6cm del borde del lateral izquierdo.
- El texto empieza a un tercio del alto del documento. En caso de estar en posición horizontal(apaisado) , el alto es el lado más corto.



4.1.2 Decorador Registro de salida

Para la aplicación del sello de registro de salida es necesario definir el decorador *registroSalidav2*.

A este decorador se le tiene que definir los siguientes parámetros:

Parámetro	Descripción	Tipo	Obligatoriedad
numAriesSalida	Número de registro de @ries	String	Sí
fechaAriesSalida	Fecha de registro de @ries	String (dd/MM/yyyy)	Sí
horaAriesSalida	Hora de registro de @ries	String (HH:mm:ss)	Sí

La posición de este sello es la siguiente:

- Localización en el lateral izquierdo del documento
- Separación de 0,9cm del borde del lateral izquierdo.
- El texto empieza a un tercio del alto del documento. En caso de estar en posición horizontal(apaisado) , el alto es el lado más corto.



5 Dependencias software

La aplicación *SelloWar* está implementada en java y compilada con jdk 8. La aplicación contiene las siguientes dependencias:

- org.apache.logging.log4j:log4j-core:2.12.1
- com.thoughtworks.xstream:xstream:1.4.13
- com.itextpdf:itext-pdfa:5.5.7
- org.bouncycastle:bcprov-jdk15on:1.67
- junit:junit:4.4
- axis:axis:1.4
- wsdl4j:wsdl4j:1.6.2

5.1 Descarga del proyecto SelloWar v4.0

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

6 Ejemplo aplicación

A continuación se expone un ejemplo donde se invocan los servicios indicados anteriormente:

```
byte[] docSellado = null;
try {
    /* Creación del cliente */
    SellarWSSoapBindingStub cliente = new SellarWSSoapBindingStub(
        new URL("http://servidor:puerto/sello-war/services/SellarWS?wsdl"),
        null);
    /* Llamada servicio getSellos */
    String[] sellos = cliente.getSellos();
    System.out.println("#####SELLOS#####");
    System.out.println("Sellos:\n");
    for (String sello : sellos) {
        System.out.println(sello);
    }
    System.out.println("#####");

    /* Llamada servicio getParametrosSello decorador registroEntradaV2 */
    String[] parametrosSello = cliente.getParametrosSello("registroEntradaV2");
    System.out.println("#####PARAMETROS#####");
    System.out.println("Parametros:\n");
    for (String parametro : parametrosSello) {
        System.out.println(parametro);
    }
    System.out.println("#####");

    /* Llamada servicio getDocumentoSellado */
    Date fechaRegistro = new Date();
    byte[] docASellar = Files.readAllBytes(Paths.get("//directorio//prueba.pdf"));

    /* Definición documento a sellar */
    DocumentoDTO documento = new DocumentoDTO();
    documento.setContenido(docASellar);

    /* Definición parámetros sello */
    ParametroDecoradorDTO[] parametros = new ParametroDecoradorDTO[3];
    parametros[0] = new ParametroDecoradorDTO();
    parametros[0].setNombreParametro("numAriesEntrada");
    parametros[0].setValorParametro("20210000000000");

    parametros[1] = new ParametroDecoradorDTO();
    parametros[1].setNombreParametro("fechaAriesEntrada");
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
    String fecha = sdf.format(fechaRegistro);
    parametros[1].setValorParametro(fecha);

    parametros[2] = new ParametroDecoradorDTO();
    parametros[2].setNombreParametro("horaAriesEntrada");
    SimpleDateFormat sdh = new SimpleDateFormat("HH:mm:ss");
    String hora = sdh.format(fechaRegistro);
    parametros[2].setValorParametro(hora);
}
```



```
/*
 * Creación tipo decorador a aplicar. El nombre debe coincidir con el
 * identificador del tipo de sello
 */
DecoradorDTO decorator = new DecoradorDTO();
decorator.setNombre("registroEntradaV2");
decorator.setParametros(parametros);

docSellado = cliente.getDocumentoSellado(documento, new DecoradorDTO[] { decorator });
} catch (Exception e) {
    e.printStackTrace();
}
```



7 Resultados

A continuación se muestran imágenes del resultado obtenido para el servicio *getDocumentoSellado* para los dos tipos de sellos explicados anteriormente.



7.1 Aplicación sello Registro de entrada

DOCUMENTO DE PRUEBA

Nº Reg. Entrada: 2021000000000000. Fecha/Hora: 17/02/2021 00:00:00



7.2 Aplicación sello Registro de salida

DOCUMENTO DE PRUEBA

Nº Reg. Salida: 2021000000000000. Fecha/Hora: 17/02/2021 00:00:00



8 Glosario

A continuación se indican los términos utilizados, en el presente documento:

Término	Descripción
SOAP	Siglas de Simple Object Access Protocol es un protocolo estándar que está actualmente bajo el auspicio de la W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.
WSDL	Siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios Web.
UDDI	Siglas del catálogo de negocios de Internet denominado Universal Description, Discovery and Integration. El registro en el catálogo se hace en XML. UDDI es una iniciativa industrial abierta (sufragada por la OASIS) entroncada en el contexto de los servicios Web. UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.
XSDL	XML Schema es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se consigue así, una percepción del tipo de documento con un nivel alto de abstracción. Fue desarrollado por el World Wide Web Consortium (W3C) y alcanzó el nivel de recomendación en mayo de 2001.
SOA	Siglas de Service Oriented Architecture, es decir, Arquitectura Orientada a Servicios.