



**Govern
de les Illes Balears**

Vicepresidència i Conselleria
d'Innovació, Recerca i Turisme
Direcció General de Desenvolupament
Tecnològic

una manera de hacer
europa 

Fondo Europeo de
Desarrollo Regional



Unió Europea

EMISERV: Manual de creació de backoffices

19 de gener de 2022

Serveis d'Administració Electrònica en el Govern de les Illes Balears

Lot 2 (Serveis de tramitació d'expedients electrònics)

Oficina Tècnica de Direcció de Projecte

Control de versions del document

Control de Canvis			
Data	Autor	Versió	Canvis
19/01/2022	Límit Technologies	v1.0	Versió inicial

Revisat per		
Nom	Data	Àrea, departament o empresa

Aprovat per		
Nom	Data	Àrea, departament o empresa

Llista de distribució		
Nom	Àrea, departament o empresa	Correu electrònic

Índex

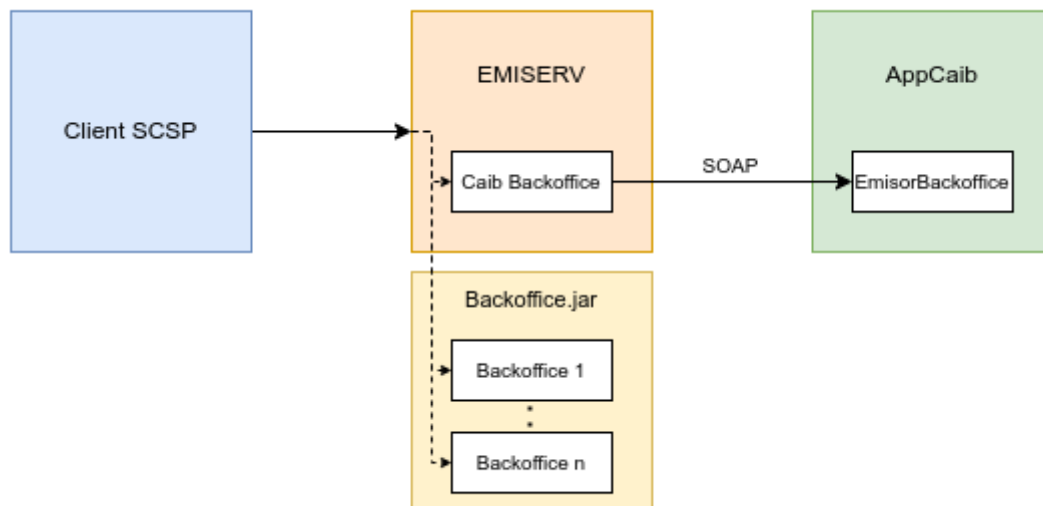
Control de versions del document.....	2
1.Introducció.....	5
2.Capçalera de l'aplicació.....	6
3.Introducció.....	7
3.1.Alta de serveis.....	7
3.2.Modificació de serveis.....	9
3.3.Permissions de serveis.....	10
3.4.Configuració SCSP.....	11
3.5.Configuració de rutes.....	13
3.5.1.Configuració de l'adreça de les rutes.....	14
3.6.Activació i desactivació de serveis.....	14
3.7.Eliminació de serveis.....	15
4.Auditories.....	16
4.1.Serveis de tipus backoffice.....	16
4.2.Serveis de tipus enrutador.....	17

1. Introducció

En aquest document es detalla el procés que s'ha de seguir per a desenvolupar serveis de backoffice per a l'aplicació EMISERV.

Es disposa de 2 tipus diferents d'implementacions de backoffices:

- Serveis web externs (SOAP) que implementen l'EmiservBackoffice.wsdl. Aquests serveis web es criden mitjançant el backoffice CaibBackoffice disponible a Emiserv.
- Implementacions de la interfície Backoffice de l'SCSP (*Sustitución de Certificados en Soporte Papel*), encapsulats en una llibreria jar.



2. Backoffices SOAP per CaibBackoffice

En aquest apartat es detallarà el procés que s'ha de seguir per a desenvolupar serveis de backoffice per a l'aplicació EMISERV fent ús del CaibBackoffice.

2.1. Generació de les classes a partir del WSDL

La primera passa per a desenvolupar el servei consistirà en la generació de les classes Java a partir del fitxer WSDL del servei. Aquest fitxer es pot trobar a la carpeta doc/backoffice del projecte.

La generació de les classes es farà utilitzant l'eina wsimport inclosa amb el JDK. La comanda a executar serà la següent:

```
$ wsimport -s src EmiservBackoffice.wsdl
```

L'eina ens generarà dues carpetes:

- Una carpeta "src" amb el codi font de les classes.
- Una carpeta "es" amb les classes compilades.

Per a la creació del servei web es poden emprar directament el codi font de la carpeta "src" o també es poden comprimir les classes compilades en un JAR i emprar-lo a dins el projecte.

A dins la carpeta doc/backoffice del projecte es pot trobar un fitxer emiserv-backoffice.jar ja creat a partir de les classes generades per wsimport.

2.2. Implementació del backoffice

Per a implementar el backoffice utilitzarem les llibreries WS de JBoss, que permeten implementar fàcilment un servei web com a un EJB.

El codi font de l'EJB podria ser el següent:

```
package es.caib.emiserv.backoffice;

import javax.jws.WebService;
import es.caib.emiserv.backoffice.ConfirmacionPetición;
import es.caib.emiserv.backoffice.EmiservBackoffice;
import es.caib.emiserv.backoffice.Petición;
import es.caib.emiserv.backoffice.Respuesta;
import es.caib.emiserv.backoffice.SolicitudRespuesta;

@Stateless
@WebService(
    name = "EmiservBackoffice",
    serviceName = "EmiservBackofficeService",
    portName = "EmiservBackofficePort",
    targetNamespace = "http://caib.es/emiserv/backoffice")
@WebContext(
    contextRoot = "/emiserv/ws",
    urlPattern = "/EmiservBackoffice",
    authMethod = "WSBASIC",
    transportGuarantee = "NONE",
    secureWSDLAccess = false)
public class EmiservBackofficeImpl implements EmiservBackoffice {
    @Override
    public Respuesta peticiónSincrona(Petición petición) {
        // TODO
        return new Respuesta();
    }
    @Override
    public ConfirmacionPetición peticiónAsincrona(Petición petición) {
        // TODO
        return new ConfirmacionPetición();
    }
    @Override
    public Respuesta solicitarRespuesta(SolicitudRespuesta solicitudRespuesta)
    {
        // TODO
        return new Respuesta();
    }
}
```

2.3. Client de prova

Per a comprovar el correcte funcionament del servei de backoffice es pot emprar la següent classe Java:

```
package es.caib.emiserv.backoffice.test;

import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

import javax.xml.namespace.QName;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.Service;
import javax.xml.ws.handler.Handler;

import org.w3c.dom.Document;
import org.w3c.dom.Element;

import es.caib.emiserv.backoffice.DatosGenericos;
import es.caib.emiserv.backoffice.EmiservBackoffice;
import es.caib.emiserv.backoffice.Peticion;
import es.caib.emiserv.backoffice.SolicitudTransmision;
import es.caib.emiserv.backoffice.Solicitudes;
import es.caib.emiserv.backoffice.Transmision;
public class BackofficeTest {

    private static final String ENDPOINT_ADDRESS =
"http://localhost:8080/backoffice/ws/EmiservBackoffice";
    private static final String USERNAME = null;
    private static final String PASSWORD = null;

    public static void main(String[] args) {
        try {
            BackofficeTest test = new BackofficeTest();
            test.testPeticio();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    private void testPeticio() throws Exception {
        Peticion peticion = new Peticion();
        Solicitudes solicitudes = new Solicitudes();
        SolicitudTransmision st = new SolicitudTransmision();
        DatosGenericos datosGenericos = new DatosGenericos();
        Transmision transmision = new Transmision();
        transmision.setIdSolicitud("000001");
        datosGenericos.setTransmision(transmision);
        st.setDatosGenericos(datosGenericos);
        st.setDatosEspecificos(generarDatosEspecificos());
        solicitudes.getSolicitudTransmision().add(st);
        peticion.setSolicitudes(solicitudes);
        getBackofficeService().peticionSincrona(peticion);
    }

    private Element generarDatosEspecificos() throws Exception {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        Document doc = factory.newDocumentBuilder().newDocument();
        Element datosEspecificos = doc.createElement("DatosEspecificos");
        Element solicitud = doc.createElement("Solicitud");
        Element tutor = doc.createElement("Tutor");
```

```
tutor.setTextContent("SI");
solicitud.appendChild(tutor);
Element provincia = doc.createElement("Provincia");
Element provinciaNom = doc.createElement("Nombre");
provinciaNom.setTextContent("Illes Balears");
provincia.appendChild(provinciaNom);
Element provinciaCodi = doc.createElement("Codigo");
provinciaCodi.setTextContent("07");
provincia.appendChild(provinciaCodi);
solicitud.appendChild(provincia);
Element municipi = doc.createElement("Municipio");
Element municipiNom = doc.createElement("Nombre");
municipiNom.setTextContent("Manacor");
municipi.appendChild(municipiNom);
Element municipiCodi = doc.createElement("Codigo");
municipiCodi.setTextContent("033");
municipi.appendChild(municipiCodi);
solicitud.appendChild(municipi);
datosEspecificos.appendChild(solicitud);
return datosEspecificos;
}

private EmiservBackoffice getBackofficeService() throws MalformedURLException {
    URL url = new URL(ENDPOINT_ADDRESS + "?wsdl");
    QName qname = new QName(
        "http://caib.es/emiserv/backoffice",
        "EmiservBackofficeService");
    Service service = Service.create(url, qname);
    EmiservBackoffice backoffice = service.getPort(EmiservBackoffice.class);
    @SuppressWarnings("rawtypes")
    List<Handler> handlerChain = new ArrayList<Handler>();
    handlerChain.add(new DatosEspecificosHandler());
    handlerChain.add(new LogMessageHandler());
    BindingProvider bp = (BindingProvider)backoffice;
    bp.getBinding().setHandlerChain(handlerChain);
    if (USERNAME != null) {
        bp.getRequestContext().put(
            BindingProvider.USERNAME_PROPERTY,
            USERNAME);
        bp.getRequestContext().put(
            BindingProvider.PASSWORD_PROPERTY,
            PASSWORD);
    }
    return backoffice;
}
```


3. Llibreria de backoffices

Emiserv permet configurar una llibreria amb backoffices per a poder-los utilitzar a Emiserv.

3.1. Configuració de la llibreria

Per a poder utilitzar la llibreria és necessari configurar el següent paràmetre:

es.caib.emiserv.backoffice.jar.path

Aquest paràmetre s'inclourà al fitxer `emiserv.system.properties`, i el valor d'aquesta propietat serà la ruta absoluta de la llibreria.

ex. `es.caib.emiserv.backoffice.jar.path=/opt/emiserv/backoffice.jar`

3.2. Implementació del backoffice

La llibreria de backoffices serà un fitxer tipus JAR de java.

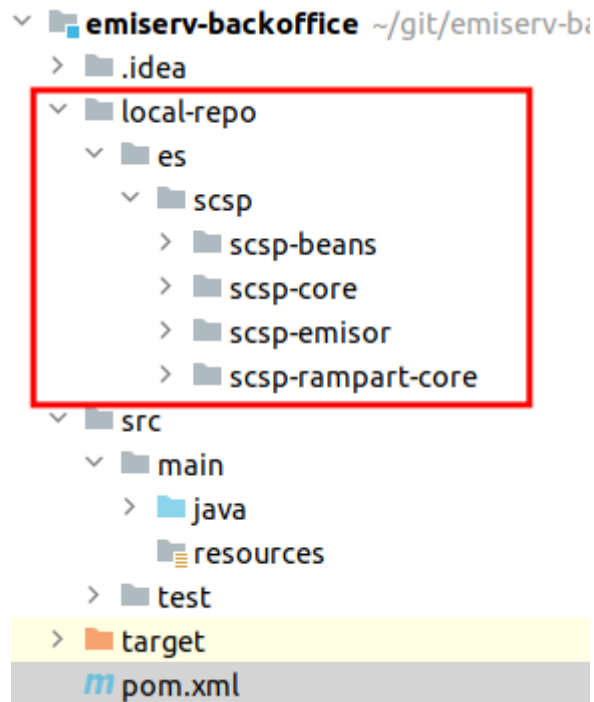
Emiserv llegirà la llibreria i detectarà totes les classes java que implementin la interfície *es.scsp.common.backoffice.BackOffice* de les llibreries SCSP. Aquestes classes seran les que estaran disponibles per a utilitzar als serveis de Emiserv.

A la llibreria no s'han d'incloure les llibreries SCSP, però si serà necessari incloure les llibreries que no es trobin disponibles a emiserv.

3.3. Exemple de llibreria utilitzant maven

Si es vol crear la llibreria de backoffices utilitzant maven, es pot realitzar en una sèrie de passes:

1. Crear un projecte maven sense utilitzar cap arquetip maven
2. Copiar a l'arrel del projecte la carpeta `local-repo/es/scsp` que es pot trobar en la darrera branca estable del projecte de Emiserv en github (<https://github.com/GovernIB/emiserv>), que conté les llibreries SCSP, respectant l'estructura de directoris.



3. Modificar el pom.xml per afegir les dependències del SCSP necessàries per a desenvolupar un backoffice SCSP, així com un repositori que faci referència a les llibreries SCSP que acabam de copiar al projecte:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>es.caib.emiserv</groupId>
  <artifactId>emiserv-backoffice</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>

  <properties>
    <scsp.core.version>4.20.0</scsp.core.version>
    <scsp.beans.version>4.3.0</scsp.beans.version>

    <maven.compiler.source>11</maven.compiler.source>
    <maven.compiler.target>11</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>es.scsp</groupId>
      <artifactId>scsp-core</artifactId>
      <version>${scsp.core.version}</version>
      <scope>provided</scope>
```

```
</dependency>
<dependency>
  <groupId>es.scsp</groupId>
  <artifactId>scsp-beans</artifactId>
  <version>${scsp.beans.version}</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>org.apache.axis2</groupId>
  <artifactId>axis2-kernel</artifactId>
  <version>1.6.4</version>
  <scope>provided</scope>
</dependency>
</dependencies>

<repositories>
  <repository>
    <id>mvn-local-repo</id>
    <url>file://${basedir}/local-repo</url>
  </repository>
</repositories>

</project>
```

4. Crear tantes classes com backoffices es vulguin desenvolupar, que implementin la interfície es.scsp.common.backoffice.BackOffice

```
package es.caib.emiserv.backoffice;

import es.scsp.bean.common.ConfirmacionPetición;
import es.scsp.bean.common.Petición;
import es.scsp.bean.common.Respuesta;
import es.scsp.bean.common.SolicitudRespuesta;
import es.scsp.common.backoffice.BackOffice;
import es.scsp.common.exceptions.ScspException;

public class BackofficeTest implements BackOffice {

    @Override
    public Respuesta NotificarSincrono(Petición petición) throws ScspException {
        ...
    }

    @Override
    public ConfirmacionPetición NotificarAsincrono(Petición petición) throws
    ScspException {
        ...
    }

    @Override
    public Respuesta SolicitudRespuesta(SolicitudRespuesta solicitudRespuesta) throws
    ScspException {
        ...
    }
}
```

4. Annexos

.1. Annex I: EmiservBackoffice.wsdl

```
<?xml version='1.0' encoding='UTF-8'?>
<wsdl:definitions name="EmiservBackofficeService"
  targetNamespace="http://caib.es/emiserv/backoffice"
  xmlns:ns1="http://schemas.xmlsoap.org/wsdl/soap/http"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://caib.es/emiserv/backoffice"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xs:schema elementFormDefault="unqualified"
      targetNamespace="http://caib.es/emiserv/backoffice" version="1.0"
      xmlns:tns="http://caib.es/emiserv/backoffice"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="peticionAsincrona" type="tns:peticionAsincrona" />
      <xs:element name="peticionAsincronaResponse"
        type="tns:peticionAsincronaResponse" />
      <xs:element name="peticionSincrona" type="tns:peticionSincrona" />
      <xs:element name="peticionSincronaResponse"
        type="tns:peticionSincronaResponse" />
      <xs:element name="solicitarRespuesta" type="tns:solicitarRespuesta" />
      <xs:element name="solicitarRespuestaResponse"
        type="tns:solicitarRespuestaResponse" />
      <xs:complexType name="solicitarRespuesta">
        <xs:sequence>
          <xs:element minOccurs="0" name="solicitudRespuesta"
            type="tns:solicitudRespuesta" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="solicitudRespuesta">
        <xs:sequence>
          <xs:element minOccurs="0" name="atributos"
            type="tns:atributos" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="atributos">
        <xs:sequence>
          <xs:element minOccurs="0" name="codigoCertificado"
            type="xs:string" />
          <xs:element minOccurs="0" name="estado" type="tns:estado" />
          <xs:element minOccurs="0" name="idPeticion" type="xs:string" />
          <xs:element minOccurs="0" name="numElementos"
            type="xs:string" />
          <xs:element minOccurs="0" name="timeStamp" type="xs:string" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="estado">
        <xs:sequence>
          <xs:element minOccurs="0" name="codigoEstado"
            type="xs:string" />
          <xs:element minOccurs="0" name="codigoEstadoSecundario"
            type="xs:string" />
          <xs:element minOccurs="0" name="literalError"
            type="xs:string" />
          <xs:element minOccurs="0" name="tiempoEstimadoRespuesta"
            type="xs:int" />
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```
</xs:complexType>
<xs:complexType name="solicitarRespuestaResponse">
  <xs:sequence>
    <xs:element minOccurs="0" name="respuesta"
type="tns:respuesta" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="respuesta">
  <xs:sequence>
    <xs:element minOccurs="0" name="atributos"
type="tns:atributos" />
    <xs:element minOccurs="0" name="transmisiones"
type="tns:transmisiones" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="transmisiones">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0"
name="transmisionDatos" nillable="true"
type="tns:transmisionDatos" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="transmisionDatos">
  <xs:sequence>
    <xs:element minOccurs="0" name="datosEspecificos"
type="xs:anyType" />
    <xs:element minOccurs="0" name="datosGenericos"
type="tns:datosGenericos" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="datosGenericos">
  <xs:sequence>
    <xs:element minOccurs="0" name="id" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="datosGenericos">
  <xs:sequence>
    <xs:element minOccurs="0" name="emisor" type="tns:emisor" />
    <xs:element minOccurs="0" name="solicitante"
type="tns:solicitante" />
    <xs:element minOccurs="0" name="titular" type="tns:titular" />
    <xs:element minOccurs="0" name="transmision"
type="tns:transmision" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="emisor">
  <xs:sequence>
    <xs:element minOccurs="0" name="nifEmisor" type="xs:string" />
    <xs:element minOccurs="0" name="nombreEmisor"
type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="solicitante">
  <xs:sequence>
    <xs:element minOccurs="0" name="consentimiento"
type="tns:consentimiento" />
    <xs:element minOccurs="0" name="finalidad" type="xs:string" />
    <xs:element minOccurs="0" name="funcionario"
type="tns:funcionario" />
    <xs:element minOccurs="0" name="idExpediente"
type="xs:string" />
    <xs:element minOccurs="0" name="identificadorSolicitante"
type="xs:string" />
    <xs:element minOccurs="0" name="nombreSolicitante"
type="xs:string" />
    <xs:element minOccurs="0" name="procedimiento"
type="tns:procedimiento" />
  </xs:sequence>
</xs:complexType>
```

```

type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="funcionario">
    <xs:sequence>
      <xs:element minOccurs="0" name="nifFuncionario"
type="xs:string" />
      <xs:element minOccurs="0" name="nombreCompletoFuncionario"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="procedimiento">
    <xs:sequence>
      <xs:element minOccurs="0" name="codProcedimiento"
type="xs:string" />
      <xs:element minOccurs="0" name="nombreProcedimiento"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="titular">
    <xs:sequence>
      <xs:element minOccurs="0" name="apellido1" type="xs:string"
/>
      <xs:element minOccurs="0" name="apellido2" type="xs:string"
/>
      <xs:element minOccurs="0" name="documentacion"
type="xs:string" />
      <xs:element minOccurs="0" name="nombre" type="xs:string" />
      <xs:element minOccurs="0" name="nombreCompleto"
type="xs:string" />
      <xs:element minOccurs="0" name="tipoDocumentacion"
type="tns:tipoDocumentacion" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="transmision">
    <xs:sequence>
      <xs:element minOccurs="0" name="codigoCertificado"
type="xs:string" />
      <xs:element minOccurs="0" name="fechaGeneracion"
type="xs:string" />
      <xs:element minOccurs="0" name="idSolicitud"
type="xs:string" />
      <xs:element minOccurs="0" name="idTransmision"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType abstract="true" name="abstractCollection">
    <xs:sequence />
  </xs:complexType>
  <xs:complexType abstract="true" name="abstractList">
    <xs:complexContent>
      <xs:extension base="tns:abstractCollection">
        <xs:sequence />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="arrayList">
    <xs:complexContent>
      <xs:extension base="tns:abstractList">
        <xs:sequence />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="peticionAsincrona">
    <xs:sequence>

```

```

                                <xs:element minOccurs="0" name="peticion"
type="tns:peticion" />
                                </xs:sequence>
                                </xs:complexType>
                                <xs:complexType name="peticion">
                                <xs:sequence>
                                <xs:element minOccurs="0" name="atributos"
type="tns:atributos" />
                                <xs:element minOccurs="0" name="solicitudes"
type="tns:solicitudes" />
                                </xs:sequence>
                                </xs:complexType>
                                <xs:complexType name="solicitudes">
                                <xs:sequence>
                                <xs:element maxOccurs="unbounded" minOccurs="0"
name="solicitudTransmision" nillable="true"
type="tns:solicitudTransmision" />
                                </xs:sequence>
                                </xs:complexType>
                                <xs:complexType name="solicitudTransmision">
                                <xs:sequence>
                                <xs:element minOccurs="0" name="datosEspecificos"
type="xs:anyType" />
                                <xs:element minOccurs="0" name="datosGenericos"
type="tns:datosGenericos" />
                                <xs:element minOccurs="0" name="id" type="xs:string" />
                                </xs:sequence>
                                </xs:complexType>
                                <xs:complexType name="peticionAsincronaResponse">
                                <xs:sequence>
                                <xs:element minOccurs="0" name="confirmacionPeticion"
type="tns:confirmacionPeticion" />
                                </xs:sequence>
                                </xs:complexType>
                                <xs:complexType name="confirmacionPeticion">
                                <xs:sequence>
                                <xs:element minOccurs="0" name="atributos"
type="tns:atributos" />
                                </xs:sequence>
                                </xs:complexType>
                                <xs:complexType name="peticionSincrona">
                                <xs:sequence>
                                <xs:element minOccurs="0" name="peticion"
type="tns:peticion" />
                                </xs:sequence>
                                </xs:complexType>
                                <xs:complexType name="peticionSincronaResponse">
                                <xs:sequence>
                                <xs:element minOccurs="0" name="respuesta"
type="tns:respuesta" />
                                </xs:sequence>
                                </xs:complexType>
                                <xs:simpleType name="consentimiento">
                                <xs:restriction base="xs:string">
                                <xs:enumeration value="Si" />
                                <xs:enumeration value="Ley" />
                                </xs:restriction>
                                </xs:simpleType>
                                <xs:simpleType name="tipoDocumentacion">
                                <xs:restriction base="xs:string">
                                <xs:enumeration value="CIF" />
                                <xs:enumeration value="NIF" />
                                <xs:enumeration value="DNI" />
                                <xs:enumeration value="Pasaporte" />
                                <xs:enumeration value="NIE" />
                                </xs:restriction>
                                </xs:simpleType>

```



```
</xs:schema>
</wsdl:types>
<wsdl:message name="peticionAsincronaResponse">
  <wsdl:part element="tns:peticionAsincronaResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="peticionSincronaResponse">
  <wsdl:part element="tns:peticionSincronaResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="peticionAsincrona">
  <wsdl:part element="tns:peticionAsincrona" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="solicitarRespuesta">
  <wsdl:part element="tns:solicitarRespuesta" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="solicitarRespuestaResponse">
  <wsdl:part element="tns:solicitarRespuestaResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="peticionSincrona">
  <wsdl:part element="tns:peticionSincrona" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="EmiservBackoffice">
  <wsdl:operation name="solicitarRespuesta">
    <wsdl:input message="tns:solicitarRespuesta" name="solicitarRespuesta">
    </wsdl:input>
    <wsdl:output message="tns:solicitarRespuestaResponse"
      name="solicitarRespuestaResponse">
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="peticionAsincrona">
    <wsdl:input message="tns:peticionAsincrona" name="peticionAsincrona">
    </wsdl:input>
    <wsdl:output message="tns:peticionAsincronaResponse"
      name="peticionAsincronaResponse">
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="peticionSincrona">
    <wsdl:input message="tns:peticionSincrona" name="peticionSincrona">
    </wsdl:input>
    <wsdl:output message="tns:peticionSincronaResponse"
name="peticionSincronaResponse">
    </wsdl:output>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="EmiservBackofficeServiceSoapBinding"
  type="tns:EmiservBackoffice">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="solicitarRespuesta">
    <soap:operation soapAction="" style="document" />
    <wsdl:input name="solicitarRespuesta">
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output name="solicitarRespuestaResponse">
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="peticionAsincrona">
    <soap:operation soapAction="" style="document" />
    <wsdl:input name="peticionAsincrona">
      <soap:body use="literal" />
    </wsdl:input>
  </wsdl:operation>
```



```
        <wsdl:output name="peticionAsincronaResponse">
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="peticionSincrona">
        <soap:operation soapAction="" style="document" />
        <wsdl:input name="peticionSincrona">
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output name="peticionSincronaResponse">
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="EmiservBackofficeService">
    <wsdl:port binding="tns:EmiservBackofficeServiceSoapBinding"
        name="EmiservBackofficePort">
        <soap:address location="http://localhost:8080/emiserv/ws/EmiservBackoffice"
/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

4.1.