



Gobierno Bolivariano  
de Venezuela

Ministerio del Poder Popular  
para Ciencia, Tecnología e Innovación

Centro Nacional de  
Tecnologías de Información (CNTI)





# GUÍA DE BUENAS PRÁCTICAS PARA LA CREACIÓN DE SERVICIOS BASADOS EN TECNOLOGÍAS WEB



# CC Creative Commons

Reconocimiento-No comercial-Compartir bajo la misma licencia 3.0.

Usted es libre de:

-  Copiar, distribuir y reproducir públicamente la obra.
-  Hacer obras derivadas.

Bajo las siguientes condiciones:

-  **Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
-  **No comercial.** no puede utilizar esta obra para fines comerciales.

 **Compartir bajo la misma licencia.** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra. Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor.

Nada en esta licencia menoscaba o restringe los derechos morales del autor.

**Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.**

Esto es un resumen fácilmente legible del texto legal de su versión original en idioma inglés (la licencia completa)

<http://creativecommons.org/licenses/by-nc-sa/3.0/ve/>

## Índice de contenido

1.- Introducción.....	3
2.- Glosario de términos.....	3
3.- Siglas.....	4
4.- Conceptualización .....	5

5.- Arquitectura. ....	6
6.- Pasos para la creación de Servicios Web.....	7
6.1.- Descripción del Servicio .....	7
6.2.- Construcción del SOAP.....	12
6.3.- Protocolo de transporte.....	15
7.-Recomendaciones para el desarrollo ágil de Servicio Web.....	16
8. Condiciones y términos para la implementación de Servicios Web.....	17
Documento de Especificaciones del Servicio de Información Interoperable.....	19

## 1.- Introducción

Con el objetivo de construir un Estado más eficiente, transparente y participativo que preste mejores servicios a los ciudadanos, por medio del aprovechamiento de las Tecnologías de Información y Comunicación, surge el Decreto con Rango, Valor y Fuerza de Ley sobre Acceso e intercambio

electrónico de datos, información y documentos entre los órganos y entes del Estado, cuyo propósito es promover el desarrollo de un modelo nacional para el intercambio, publicación e interpretación de los datos, información y documentos, que apoye el establecimiento de políticas, lineamientos y estrategias públicas.

Es por ello que el Centro Nacional de Tecnologías de Información (CNTI) ha venido desarrollando recomendaciones técnicas que sirvan de referencia para efectuar el intercambio electrónico de datos, información y documentos en los órganos y entes del Estado. Para ello, deben considerarse la implementación de mecanismos automatizados que permitan conocer, procesar, reusar y disponer de datos información y documentos en las instituciones públicas en forma adecuada, confiable, oportuna y de fácil acceso, basados en las mejores prácticas tecnológicas aplicadas con estándares abiertos.

En tal sentido, surge la presente guía de buenas prácticas con el fin de brindar a los órganos y entes del Estado una herramienta sencilla para el diseño, construcción e implementación de servicios basados en tecnologías Web, facilitando el proceso de incorporación de los datos, información y documentos a la Plataforma Nacional de Interoperabilidad que contribuya con la ejecución de trámites y servicios requeridos por los ciudadanos.

## 2.- Glosario de términos

A los fines de esta guía se han considerado pertinente la definición de los términos que se presentan a continuación:

**Dato:** Hecho, concepto, instrucción o caracteres, que se expresa por sí mismo, representado de una manera apropiada para que sea comunicado, transmitido o procesado por seres humanos o por medios automáticos, y al cual se le asigna o se les puede asignar un significado.

**Estándar:** especificación establecida por un organismo de reconocimiento en la industria, que sirve de guía para regular la generación de componentes que habiliten la interoperabilidad entre entidades. Sirve como apoyo para alguna recomendación.

**Información:** Significado que el ser humano le asigna al dato o al conjunto organizado de datos

procesados, utilizando las convenciones conocidas y generalmente aceptadas.

**Interoperabilidad:** Capacidad de los órganos y entes del Estado de intercambiar por medios electrónicos datos, información y documentos de acceso público.

**Plataforma Nacional de Interoperabilidad:** conjunto de componentes tecnológicos, sistemas y servicios, que permite a los órganos y entes del Estado, intercambiar datos, información y documentos haciendo uso del estándar de interoperabilidad.

**Servicio:** conjunto de actividades que buscan proporcionar valor agregado a los usuarios al ofrecer un beneficio o satisfacer sus necesidades.

**Servicios de Información Interoperables:** Servicio que reúne, procesa, reusa y dispone datos, información y documentos, en función de la demanda de los órganos y entes del Estado, en forma adecuada, confiable, oportuna y de fácil acceso.

### 3.- Siglas

A continuación se listan los acrónimos utilizados en la guía.

1. **XML:** Extensible Markup Language (Lenguaje de marcas extensible).
2. **HTML:** Hyper Text Markup Language (Lenguaje de marcas de hipertexto)
3. **SOAP:** Simple Object Access Protocol
4. **Tags:** Etiqueta utilizada para delimitar una región en los lenguajes basados en XML.
5. **Rest:** Representational State Transfer ( Transferencia de estado representacional - técnica de arquitectura de software)
6. **RPC:** Remote Procedure Call (Procedimiento remoto)
7. **URL:** Localizador de Recurso uniforme (Uniform Resource Locator)
8. **W3C:** World Wide Web Consortium (Consortio Internacional que produce recomendaciones para la World Wide Web)
9. **WS:** Web Service (Servicio Web)
10. **WDSL:** Lenguaje de descripción de servicios web (Web Services Description Language)

## 11. XSD: XML Schema Definition (formato para definir la estructura de un documento XML)

### 4.- Conceptualización

Los intercambios de información entre las instituciones públicas implementados han sido desarrollados sobre una amplia variedad de formatos, tipos e interacciones, impulsando el crecimiento y proliferación de prácticas no estandarizadas que han contribuido con la desarticulación de los organismos responsables en la prestación de servicios al ciudadano. Es por ello que se requiere el desarrollo de servicios tecnológicos que cumplan con lineamientos técnicos establecidos por las leyes que rigen la materia y estén adaptados a las necesidades de los ciudadanos y a los procesos institucionales, disminuyendo las dificultades de integración entre los sistemas de información presentes en el sector público nacional.

Los órganos y entes del Estado deben orientar el desarrollo de sus sistemas de información a:

- El diseño y utilización de Arquitecturas Orientadas a Servicios que eviten así el desarrollo de aplicaciones monolíticas, que dificulten la Interoperabilidad con sus sistemas y en consecuencia también dificulten el trabajo cohesionado con otros órganos y entes del Estado.
- La utilización de las tecnologías de desarrollo de aplicaciones, protocolos de comunicación, formatos de intercambio de datos y mecanismos de seguridad que permitan y faciliten la Interoperabilidad de sus Sistemas de Información.
- La incorporación y uso de estándares abiertos que faciliten su incorporación a la Plataforma Nacional de Interoperabilidad provista por el ente rector de la materia.

Las mejores prácticas para la creación de servicios de esta naturaleza vienen dadas por lo indicado en el Perfil Básico o WS-I Basic Profile, definido por la Organización para la Interoperabilidad de Servicios Web - Web Services Interoperability (WS-I) . El Perfil Básico o WS-I Basic Profile es un protocolo cuyo objetivo es fomentar y promover la Interoperabilidad de Servicios Web sobre cualquier plataforma, aplicaciones y sobre lenguajes de programación. Su intención es ser un integrador de estándares para ayudar al avance de los servicios web de una manera estructurada y coherente.

La WS-I ha organizado los estándares para la interoperabilidad de los Servicios Web en una estructura

basada en diversas funcionalidades.

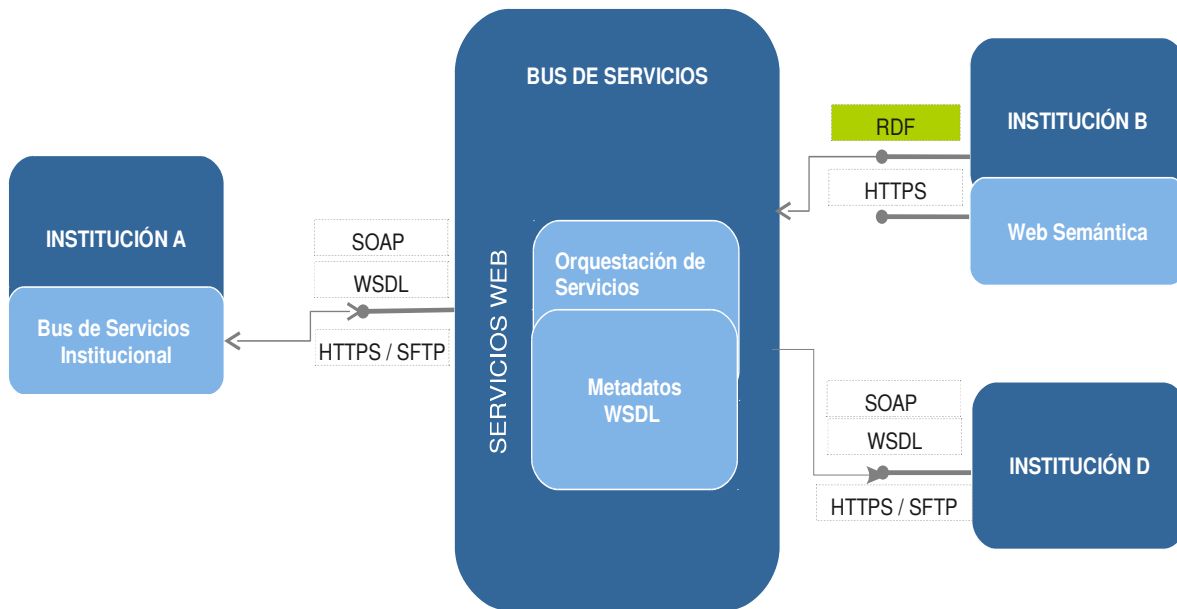
Capa	Clasificación	Estándar
Aplicación	Consulta	UDDI
	Dominio	DNS
Datos	Representación de datos	XML
	Definición de datos para el intercambio	XSD
		XML Schema
		WSDL
Comunicación	Servicios Web	SOAP
Transporte	Transferencia de archivos e hipertexto	HTTP
Seguridad	General	WS-Security, WS-Policy WS-Trust WS-Addressing

En consecuencia esta guía orientada a los órganos y entes del Estado pretende:

- Proporcionar información acerca de los conceptos básicos relacionados con los Servicios Web.
- Proporcionar los estándares aplicables para el diseño, construcción e implementación de servicios basados en tecnologías Web.
- Establecer un conjunto de pasos para facilitar la incorporación de servicios a la Plataforma Nacional de Interoperabilidad.
- Brindar recomendaciones acerca de herramientas basadas en estándares abiertos para la construcción de servicios basados en tecnologías Web.

## 5.- Arquitectura.

La arquitectura propuesta para el intercambio de información entre varias instituciones en la Plataforma Nacional de Interoperabilidad se basa en la definición y publicación de los diferentes Servicios Web a ser desplegados en el Bus de Servicios Institucional, tal y como se describe en la siguiente figura:



Esquema de Intercambio de información con base a una Arquitectura de Servicios Web

## 6.- Pasos para la creación de Servicios Web

Descripción del Servicio, Construcción de SOAP y el Protocolo de Transporte.

### 6.1.- Descripción del Servicio

El primer paso para el desarrollo de un Servicio Web es la descripción del servicio, la cual se crea a través del documento WSDL, utilizando los siguientes elementos:

Elemento	Descripción
<types>	Define los tipos de datos utilizados por el servicio web.
<message>	Define los elementos de datos de una operación.
<portType>	Define en un servicio web, las operaciones que se pueden realizar, y los mensajes que están involucrados.
<binding>	Proporciona detalles específicos sobre la manera cómo se transmite una operación <portType> sobre la comunicación.



La estructura principal de un documento WSDL se describe de la siguiente manera:

```
<definitions>

<types>
  data type definitions.....
</types>

<message>
  definition of the data being communicated...
</message>

<portType>
  set of operations.....
</portType>

<binding>
  protocol and data format specification....
</binding>

</definitions>
```

Sin embargo el documento WSDL puede contener también otros elementos, como elementos de extensión y un elemento de servicio, cuyo propósito es agrupar las definiciones de varios Servicios Web en un solo documento WSDL, como sigue:

- **Elemento <message>** define los elementos de datos de una operación. Cada mensaje puede constar de una o más partes. Las partes se pueden comparar con los parámetros de una llamada de función en un lenguaje de programación tradicional.
- **Elemento <types>** define los tipos de datos que son utilizados por el servicio Web. Para conseguir la máxima neutralidad plataforma, WSDL utiliza la sintaxis de esquema XML para definir tipos de datos.

A continuación se ilustra una fracción simplificada de un documento WSDL:

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

En el ejemplo ilustrado, el elemento **<portType>** define "glossaryTerms" como el nombre de un **port**, y "getTerm" como el nombre de una **operation**.

La operación "getTerm" tiene un **input message** llamado "getTermRequest" y un **output message** llamado "getTermResponse".

Los elementos **<message>** definen las **parts** de cada mensaje y los tipos de datos asociados.

- **Elemento <portType>** Es el elemento más importante del WSDL. En él se describe el servicio web, las operaciones que se pueden realizar, y los mensajes que están involucrados. Éste elemento se puede comparar con una biblioteca de funciones (o un módulo, o una clase) en un lenguaje de programación tradicional.

En la siguiente tabla se indican los cuatro tipos de operaciones definidos por el WSDL:

Tipo	Definición
One-way	La operación puede recibir un mensaje, pero no devolverá una respuesta .
Request-response	La operación puede recibir una petición y devolverá una respuesta.
Solicit-response	La operación puede enviar una solicitud y esperará una respuesta.
Notification	La operación se puede enviar un mensaje, pero no va a esperar por una respuesta.

A continuación se ilustra un ejemplo de operación One-Way(un sólo sentido):

```
<message name="newTermValues">
  <part name="term" type="xs:string"/>
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="setTerm">
    <input name="newTerm" message="newTermValues"/>
  </operation>
</portType >
```

En el ejemplo anterior, los "glossaryTerms" **portType** define una operación unidireccional llamado "setterm".

La operación "setterm" permite la entrada de los nuevos términos del glosario mensajes mediante un mensaje "newTermValues" con los parámetros de entrada "term" y "value". Sin embargo, no hay salida se define para la operación.

El siguiente ejemplo hace referencia a una una operación request-response (petición - respuesta);

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

En el ejemplo ilustrado, los "glossaryTerms" portType define una operación de petición-respuesta llamado "getTerm". La operación "getTerm" requiere un mensaje de entrada llamado "getTermRequest" con un parámetro llamado "term", y devolverá un mensaje de salida llamado

"getTermResponse" con un parámetro llamado "value".

- **Elemento <binding>** define el formato de datos y protocolo para cada tipo de puerto y tiene dos atributos de estilo y de transporte.

El atributo de estilo puede ser "rpc" o "documento". En este caso usamos documento. El atributo de transporte define el protocolo SOAP para utilizar. En este caso usamos HTTP.

El elemento de **operation** define cada operación que el portType expone. Para cada operación de la acción SOAP correspondiente tiene que ser definido. También debe especificar la codificación de la entrada y la salida. En este caso usamos la palabra "literal".

A continuación se ilustra un ejemplo de una operación de petición – respuesta;

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>

<binding type="glossaryTerms" name="b1">
  <soap:binding style="document"
  transport="http://schemas.xmlsoap.org/soap/http" />
  <operation>
    <soap:operation soapAction="http://example.com/getTerm"/>
    <input><soap:body use="literal"/></input>
    <output><soap:body use="literal"/></output>
  </operation>
</binding>
```

En el ejemplo ilustrado , el elemento **<binding>** tiene dos atributos - el nombre y el tipo.

El atributo de nombre (puede utilizar el nombre que desee) define el nombre de la unión, y los puntos de atributo de tipo de puerto para la unión, en este caso el "glossaryTerms" puerto.

## 6.2.- Construcción del SOAP

Una vez descrito el Servicio Web se procede a la construcción del SOAP, en este apartado indicamos la manera de hacerlo.

Un mensaje SOAP es un documento XML que contiene los siguientes elementos:

- Un elemento **Envelope** que identifica el documento XML como un mensaje SOAP
- Un elemento **Header** que contiene información de encabezado.
- Un elemento **Body** que contiene la información de llamada y respuesta.
- Un elemento **Fault** contiene errores y la información de estado

Para la creación de un mensaje SOAP es importante cumplir las siguientes reglas de sintaxis :

- Debe ser codificado utilizando XML.
- Debe utilizar el espacio de nombres SOAP.
- Debe utilizar el espacio de nombres de codificación de SOAP.
- No debe contener una referencia DTD.
- No deben contener instrucciones de procesamiento XML

A continuación se ilustra el esqueleto de un mensaje SOAP:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
...
</soap:Header>

<soap:Body>
...
  <soap:Fault>
    ...
  </soap:Fault>
</soap:Body>

</soap:Envelope>
```

- **Elemento SOAP – Envelope**

Es el elemento raíz de un mensaje SOAP, el cual define el documento XML como un mensaje SOAP.

Un ejemplo del elemento SOAP Envelope es:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  ...
  Message information goes here
  ...
</soap:Envelope>
```

- **Elemento SOAP – Header:**

Es opcional contiene información específica de la aplicación como la autenticación, el pago, etc sobre el mensaje SOAP. Si el elemento header ésta presente, debe ser el primer elemento secundario del elemento Envelope.

**Nota:** Todos los elementos secundarios inmediatos del elemento header deben ser compatibles con espacio de nombres.

Ejemplo del elemento SOAP header:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
  <m:Trans xmlns:m="http://www.w3schools.com/transaction/"
  soap:mustUnderstand="1">234
  </m:Trans>
</soap:Header>
  ...
  ...
</soap:Envelope>
```

El ejemplo anterior contiene un header (encabezado) con los atributos "trans" y "mustUnderstand" con un valor de 1, y 234.

Así mismo, SOAP define tres atributos del espacio de nombres por defecto ("http://www.w3.org/2001/12/soap-envelope"). Estos atributos son: mustUnderstand, actor y encodingStyle. Los atributos definidos en el SOAP Header definen cómo un receptor debe procesar el mensaje SOAP.

- **Elemento SOAP - Body**

Es un elemento obligatorio del SOAP, que contiene información dirigida al destinatario final del mensaje. Elementos secundarios inmediatos del elemento SOAP Body pueden ser namespace.

Por ejemplo:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body>
    <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>

</soap:Envelope>
```

El ejemplo ilustrado es una solicitud del precio de las manzanas.

Una respuesta a éste mensaje sería así:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body>
    <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices">
      <m:Price>1.90</m:Price>
    </m:GetPriceResponse>
  </soap:Body>
```

```
</soap:Envelope>
```

- **Elemento SOAP – Fault:**

El elemento de error es opcional y se utiliza para notificar los mensajes de error. Si un elemento Fault está presente, debe aparecer como un elemento secundario del elemento Body y sólo puede aparecer una vez en un mensaje SOAP.

El elemento de Fault tiene los siguientes elementos secundarios:

Sub-elemento	Descripción
<faultcode>	Un código de identificación del error
<faultstring>	Una explicación legible del error
<faultactor>	Información acerca de quién causó el error
<detail>	Contiene información específica de la aplicación de error relacionado con el elemento body.

Los códigos de error SOAP definidos a continuación se deben utilizar en el elemento **faultcode** al describir errores:

Error	Descripción
VersionMismatch	Encontró un espacio de nombres no válidos para el elemento SOAP
MustUnderstand	Un elemento secundario inmediato del elemento Header, con el atributo “mustUnderstand” establece en "1", no se entendió.
Client	El mensaje fue incorrectamente formado o contiene información incorrecta
Server	Hubo un problema con el servidor por lo que el mensaje no se pudo procesar .

### 6.3.- Protocolo de transporte

El Protocolo de Transporte HTTP se comunica a través de TCP / IP. Un cliente HTTP se conecta a un servidor HTTP utilizando TCP. Después de establecer una conexión, el cliente puede enviar un mensaje de solicitud HTTP al servidor, como se muestra a continuación:

```
POST /item HTTP/1.1
```





```
Host: 189.123.255.239
Content-Type: text/plain
Content-Length: 200
```

Después, el servidor procesa la solicitud y envía una respuesta HTTP de vuelta al cliente. La respuesta contiene un código que indica el estado de la solicitud, así:

```
200 OK
Content-Type: text/plain
Content-Length: 200
```

En el ejemplo anterior, el servidor devuelve un código de estado de 200. Este es el código de éxito estándar para HTTP.

Si el servidor no pudo descifrar la solicitud, podría haber devuelto algo como esto:

```
400 Bad Request
Content-Length: 0
```

#### **Nota:** HTTP - SOAP Binding

- Un método SOAP es una petición / respuesta HTTP que cumple con las reglas de codificación SOAP.  
HTTP + XML = SOAP
- Una petición SOAP podría ser un HTTP POST o una solicitud GET HTTP.
- La solicitud HTTP POST especifica al menos dos cabeceras HTTP: Content-Type y Content-Length.

#### **7.-Recomendaciones para el desarrollo ágil de Servicio Web.**

- a) Existen implementaciones a nivel de lenguajes de programación de manera automática denominados framework que crean WSDL y SOAP a nivel cliente y servidor, como los que se mencionan a continuación:

En Java las APIS de desarrollo de servicios web utilizadas son:

- JAX-WS y JAX-RS.
- Apache Axis.



- Apache Axis2.
- Metro.

En PHP las librerías de desarrollo usadas son:

- Nusoap.
- En php 5 hay soporte de librerías nativas para el desarrollo de servicios web, estas son:
  - ✓ SOAP.
  - ✓ OAUTH.
  - ✓ SCA.
  - ✓ YAR.
  - ✓ XML-RPC.

En Python las librerías de desarrollo usadas son:

- OOP.
- CHILKAT.
- SOAPLIB.
- MOD\_WSGI.
- PyXML.
- FPCONST.
- SOAPpy.
- SUDS.

b) Para alcanzar el grado de interoperabilidad deseado es necesario que el cliente y el servidor que implementa el servicio, acuerden un protocolo mediante el cual se define como se invoca el servicio, como se pasan parámetros, como se recibe un resultado, como se manejan errores, entre otros. Los protocolos mas usados son XML (Web Service) y RPC (Web Semántica).

c) Definir adecuadamente el esquema de seguridad cuyo objetivo es lograr la integridad, confidencialidad y autenticidad de los datos. Los estándares para la seguridad de los servicios son los recomendados por la autoridad con competencia en la materia de seguridad de la información, sin embargo en esta guía se recomienda el Basic Security Profile WS-I.

## 8. Condiciones y términos para la implementación de Servicios Web.

Los órganos y entes del Estado deben considerar las premisas que a continuación se describen, con el fin de garantizar la interoperabilidad entre los servicios basados en Tecnologías Web.



- a) Las decisiones de la unidad de tecnología de un ente del Estado, durante el desarrollo de un servicio web, deben guiarse por el uso de especificaciones técnicas que faciliten su interconexión con el mayor número de sistemas que conforman el ecosistema de soluciones con el que interopera.
- b) Los órganos y entes del Estado a cargo del servicio web deben describir la siguiente información asociada al servicio:
- Definir el nombre del servicio de intercambio de información.
  - Incluir una descripción general del servicio de información que va a ser intercambiado.
  - Asignar un responsable de la gestión del servicio de intercambio de información con los datos asociados: nombre y apellido, cargo, teléfonos de contacto y correo electrónico.
  - Describir las formas de proveer el servicio, brindando los detalles técnicos asociados con la mayor claridad posible.
  - Describir igualmente los datos relacionados con la versión y estado así como la documentación relacionada al servicio.

Nota: Ver Anexo A.

## ANEXO A

### Documento de Especificaciones del Servicio de Información Interoperable

El documento de especificaciones del Servicio de Información Interoperable, incorpora en tres secciones descriptivas con información del servicio de información interoperable:

#### *Sección I – Información General*

Institución:			
Cód del SII	Nombre del SII	Versión	Última actualización

#### *Sección II – Soporte*

Responsable General			
Datos de Contacto	Telf:		
	E-mail:		
Responsables de Soporte	Contacto 1		
	Contacto 2		

#### *Sección III – Especificaciones Técnicas*

Arquitectura	
Tipo	
Protocolo de comunicación	
Fuente de datos	Tipo de archivo
Transmisión	
En línea	
Fuera de Línea	



Gobierno Bolivariano  
de Venezuela

Ministerio del Poder Popular  
para Ciencia, Tecnología e Innovación

Centro Nacional de  
Tecnologías de Información (CNTI)

