TITULO

**Contenido:**

[1 Estándares 3](#_Toc387355480)

[1.1 REST 3](#_Toc387355481)

[1.2 SOAP 3](#_Toc387355482)

[1.3 Web Services Geográficos 4](#_Toc387355483)

[1.3.1 WMS 4](#_Toc387355484)

[1.3.2 WFS 4](#_Toc387355485)

[1.4 WS-SECURITY 5](#_Toc387355486)

[1.5 MTOM 5](#_Toc387355487)

[1.6 ESB 5](#_Toc387355488)

[2 Tecnologías 5](#_Toc387355489)

[2.1 GEOSERVER 5](#_Toc387355490)

[2.2 JBOSS ESB 6](#_Toc387355491)

[2.3 Plataforma de eGob de Uruguay 6](#_Toc387355492)

[2.3.1 Plataforma de Interoperabilidad 7](#_Toc387355493)

[3 Referencias 8](#_Toc387355494)

# Estándares

## REST

Una API REST es un tipo de arquitectura de desarrollo web basada en el estándar HTTP y que permite interconectar sistemas hypermedia distribuidos.

A diferencia de SOAP no está orientado a servicios (RPC) sino a recursos. Un recurso es cualquier información que pueda ser almacenada, una entidad que representa un concepto de negocio que a su vez puede ser accedido por otro sistema. Cada recurso posee un identificador único que lo distingue de otros recursos.  Algunos recursos son estáticos, o sea, en cualquier momento en que sean examinados luego de su creación siempre les va a corresponder el mismo resultado. Otros cambian a medida que pasa el tiempo[8].

La comunicación es stateless, cada recurso posee un estado interno, este estado no puede ser accedido desde el exterior. Cada pedido hecho del cliente al servidor debe tener toda la información necesaria para que el request pueda ser comprendido. Y no se puede sacar ventaja de tener un contexto guardado para cada sesión. La información de las sesiones son almacenadas enteramente en el cliente. Esto también mejora la visibilidad, confiabilidad y escalabilidad del servicio[8].

Un servicio REST permite las siguientes 4 operaciones sobre un recurso :

**GET:** Para obtener información sobre un recurso.

**POST:** Para crear una nueva instancia del recurso.

**PUT:** Para modificar un recurso existente.

**DELETE:** Para eliminar un recurso.

Se puede entender como una aplicación CRUD (Create, Read, Update and Delete), pero a diferencia de estas, un servicio REST le puede asignar otro comportamiento a una de estas operaciones estándar.

## SOAP

SOAP(Simple Object Access Protocol) es un protocolo para la comunicación entre aplicaciones por medio de internet. Es independiente de la plataforma y el lenguaje y define un formato para envío y recepción de mensajes basado en XML[11].

Se publican servicios como RPC (remote call procedure) y la comunicación funciona por medio de un request enviado por el cliente a lo que se devuelve una respuesta del servidor.

Se utiliza el lenguaje WSDL (Web Service Description Language) para describir los servicios que se ofrecen, así como el formato de la entrada y la salida de cada servicio. Un web service SOAP  tiene una especificación asociada escrita en este lenguaje que está basado en XML[12].

## Web Services Geográficos

Open Geospatial Consortium (OGC) [3] es la organización que propone los estándares de Web Services Geográficos, los cuales están agrupados bajo el nombre OWS.

Las siguientes descripciones de los protocolos Web Map Service (WMS) y Web Feature Service (WFS) se arman a partir de la conjunción de [4], [5], [6], y [7].

### WMS

WMS [4] es un estándar que define un protocolo para obtener mapas dinámicos, de datos referenciados espacialmente a partir de información geográfica distribuida. El mapa que se obtiene para representar la información geográfica es un archivo de imagen en formato PNG, GIF o JPEG, o en ocasiones como elementos gráficos basados en las especificaciones Scalable Vector Graphics (SVG) o Web Computer Graphics Metafile (WebCGM).

WMS define tres operaciones para publicar datos geográficos:

**getCapabilities:** Devuelve los metadatos del servicio, tales como, formatos que soporta, datos que posee, información de los valores admitidos de los parámetros, etc.

**getMap:** Devuelve un mapa geográfico a partir de ciertas capas.

**getFeatureInfo:** Operación opcional que devuelve información acerca de las características particulares mostradas en el mapa.

### WFS

WFS [5] es un estándar que define un protocolo para consultar y modificar información geográfica codificada en Geography Markup Language (GML).

WFS define seis operaciones para consultar y modificar datos geográficos:

**GetCapabilities:** Devuelve información del servicio.

**DescribeFeatureType:** Describe la estructura de cualquier tipo de entidad soportado por el servicio.

**GetFeature:** Devuelve los datos geográficos resultado de la consulta realizada por el cliente (analogía con SELECT).

**GetGmlObject:** Devuelve cualquier elemento GML a través de su identificador único.

**Transaction:** Operación que permite modificar las características de los datos geográficos, tales como, darlos de alta, modificarlos y darlos de baja.

**LockFeature:** Operación opcional que permite bloquear una entidad al momento de realizar una transacción.

En base a las operaciones ofrecidas los servicios WFS se clasifican de la siguiente manera:

**WFS Básico:** Estos servicios son únicamente de lectura y ofrecen las operaciones GetCapabilities, DescribeFeatureType y GetFeature.

**WFS con XLink:** Ofrecen las operaciones de un WFS básico más la operación GetGmlObject. Con XLink significa que el servidor puede solicitar datos extra a otros servidores.

**WFS Transaccional:** Ofrecen las operaciones de un WFS básico más la operación Transaction.

**WFS Transaccional con XLink:** Ofrecen las operaciones de un servicio WFS Transaccional más la operación GetGmlObject.

## WS-SECURITY

Es una extensión de los estándares de Web Services, publicado y mantenido por OASIS, encargado de brindar seguridad extremo a extremo a nivel del mensaje SOAP. Características similares pueden obtenerse utilizando TLS, aunque tiene el problema de que el mensaje no puede utilizarse para ruteo intermedio, debido que TLS encripta todo el mensaje http.

WSS brinda dos servicios básicos, confidencialidad e integridad. Los cuales pueden alcanzarse utilizando diferentes métodos, XML Signature y XML Encription. Estos se basan e integran con diferentes tecnologías de seguridad, como tokens de seguridad, certificados X.509 y Kerberos, entre otras.

Provee ciertas ventajas sobre la opción de TLS, como por ejemplo se puede cifrar todo o parte del mensaje, lo cual permite utilizar las partes no criticas para ruteo o controles de acceso.

Se integra con otros estándares de seguridad como WS-SecureConversation y WS-Trust.

También soporta la integración con SAML utilizado para proveer SingleSign on a lo largo de internet, permitiendo la interacción entre Proveedores de identidad y proveedores de servicio que confían en los mismos.

## MTOM

## Es un estándar de la W3C para la transmisión de datos binarios de forma optima, como lo indica su nombre: Message Transmission Optimization Mechanism[13].

Debido a que XML no provee una forma de incluir datos binarios, previo al surgimiento de MTIOM la única forma de enviar o recibir este tipo de datos era transformarlo a Base64.

Base64 es una técnica que transforma datos binarios en una tira de caracteres ASCII de forma reversible. Esta solución que todavía se utiliza para algunos contextos. El problema de base64 es que hace crecer el tamaño de los datos, porque para la transformación convierte 6 bits de información en su correspondiente carácter ASCII de 8 bits, agregando un total de 25% extra de información. A priori puede parecer poco, pero para archivos grandes, de Gigabytes, un 25% es mucho.

MTOM soluciona este problema haciendo uso de los pedidos multi-parte de http. Funciona agregando en el body del mensaje SOAP “punteros” hacia las partes del mensaje http donde vienen los datos binarios sin codificar. De esta manera no agrega el overhead de Base64.

## ESB

Enterprise service bus es un modelo de arquitectura de software usado para implementar comunicación entre sistemas orientados a servicios. Generalmente se basa en plataformas que implementan y potencian sistemas de mensajería, definiendo puntos de entrada, principalmente web services SOAP, y agregan servicios dentro del bus como seguridad y confidencialidad, así como también ruteo de mensajes según condiciones del mensaje o del contexto. También proveen servicios de transformación de datos, para adecuar los mensajes a diferentes sistemas interactuando los cuales agregan o quitan información.



**Figura X – Esquema de ESB[[1]](#footnote-1)wiki**

# Tecnologías

## GEOSERVER

GeoServer es un servidor WMS de código abierto que permite a los usuarios compartir y editar datos geoespaciales. Está diseñado para la interoperabilidad, publica los datos de cualquier fuente importante de datos espaciales usando estándares abiertos[9].

Es la implementación de referencia de los estándares Open Geospatial Consortium (OGC) Web Feature Service (WFS) y Web Coverage Service (WCS). Soporta WFS y WFS-Transaccional [9].

Posee los siguientes features[10]:

- Es el más adecuado para los desarrolladores java ya que ofrece facilidad para interoperar con aplicaciones escritas en esa plataforma.

- Soporta WFS y WFS-T.

- Como servidor soporta una variedad de formatos de almacenamiento (PostGis, Oracle spatial, Mysql,  GeoTiff) tanto vectorial como raster.

- Reproyección al vuelo.

- WMS Tiling cache (Usa GeoWebCache como cliente de tiles WMS).

## JBOSS ESB

Es un producto de Red Hat y su comunidad, una implementación de la arquitectura ESB en Java, basada también en el servidor de aplicaciones JBoss de la misma compañía. Se basa en estándares, tanto de Java como de la Web para brindar conectividad y comunicación con muchos otros tipos de tecnologías. En la siguiente figura vemos un esquema de todas las tecnologías de comunicación soportadas por este producto.



**Figura X – Esquema de ESB[[2]](#footnote-2)wiki**

Básicamente en Jboss ESB un servicio define un conjunto de Listeners de los cuales quiere estar al tanto cuando llegan mensajes, luego el servicio es notificado de que un mensaje ha llegado y define para éste, filtros o procesadores que trabajarán con la información contenida en él. Por último el servicio ruteará el mensaje hacia algún conector de salida para invocar otro servicio fuera del ESB o dará una respuesta a quien inició la comunicación en primera instancia.

## Plataforma de eGob de Uruguay

El gobierno electrónico o gobierno en red radica en la idea de la construcción de una Administración Pública enfocada en el ciudadano, siempre accesible y más cercana y que hace uso intensivo de las Tecnologías de la Información y las Comunicaciones (TIC) [1].

La plataforma de gobierno electrónico es un facilitador para el desarrollo de servicios y tramites en línea que provee servicios transversales y herramientas comunes a los Organismos del Estado así como también servicios a personas, empresas y organismos. Ambiente que permite instrumentar la interoperabilidad y el intercambio de información entre Organismos. Es el contexto tecnológico y legal que permite asegurar que la información intercambiada cumpla con los requisitos legales y tecnológicos predefinidos [1].



**Figura X - Plataforma de Gobierno Electrónico y Otros Servicios[[3]](#footnote-3)**

### Plataforma de Interoperabilidad

*"La Plataforma de Interoperabilidad (PDI)  forma parte de la Plataforma de Gobierno Electrónico (PGE) de AGESIC y tiene como objetivo general facilitar y promover la implementación de servicios de Gobierno Electrónico en Uruguay. Para esto, la PDI brinda mecanismos que apuntan a simplificar la integración entre los organismos del Estado y a posibilitar un mejor aprovechamiento de sus activos." [[4]](#footnote-4)*



**Figura X - Plataforma de Interoperabilidad [[5]](#footnote-5)**

La PDI está basada en una arquitectura orientada a servicios (SOA) y compuesta por un sistema de control de acceso, un sistema de gestión de metadatos y una plataforma de middleware [2].

**Sistema de control de acceso:** Es el punto de entrada a la plataforma y provee mecanismos de autenticación y autorización para el consumo de servicios basados en XML [2].

**Sistema de gestión de metadatos:** Provee especificación de alto nivel de los conceptos relativos a de los servicios públicos [2].

**Plataforma de middleware:** Cuenta con mecanismos para facilitar el desarrollo, despliegue e integración de servicios y aplicaciones. Está integrado por dos ESB, uno de tecnología Microsoft y otro de tecnología Java, con el fin de obtener lo mejor de ambos y ampliar el espectro de posibilidades en cuanto a los métodos de conexión.  Los organismos pueden utilizar esta plataforma para publicar y descubrir servicios, así como utilizar las diferentes capacidades de mediación, las cuales permiten desacoplar clientes y servicios [2].

# Referencias

1. AGSIC. [En línea] [Citado el: 05 de 05 de 2014.] http://www.agesic.gub.uy/innovaportal/v/452/1/agesic/plataforma\_de\_gobierno\_electronico.html?menuderecho=5.

2. AGSIC. [En línea] [Citado el: 6 de 5 de 2014.] http://agesic.gub.uy/innovaportal/v/1710/1/agesic/plataforma\_de\_interoperabilidad.html?menuderecho=5.

3. Open Geospatial Consortium. [En línea] [Citado el: 02 de 05 de 2014.] http://www.opengeospatial.org/.

4. Open Geospatial Consortium. [En línea] [Citado el: 02 de 05 de 2014.] http://www.opengeospatial.org/standards/wms.

5. Open Geospatial Consortium. [En línea] [Citado el: 02 de 05 de 2014.] http://www.opengeospatial.org/standards/wfs.

6. Facultad de Ingeniería de la Universidad de la República Oriental del Uruguay. [En línea] [Citado el: 02 de 05 de 2014.] http://www.fing.edu.uy/inco/cursos/tsi/TSIG/clases2012/WebServicesGeograficos2012.pdf.

7. Facultad de Ingeniería de la Universidad de la República Oriental del Uruguay. *Tesis de Maestría de Raquel Sosa .* [En línea] [Citado el: 02 de 05 de 2014.] http://www.fing.edu.uy/~raquels/TesisRaquelSosa\_vf\_1.2.pdf.

8. **Fielding, Roy Thomas.** DISSERTATION. [En línea] 2000. [Citado el: 09 de 05 de 2014.] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\_arch\_style.htm.

9. GeoServer. [En línea] http://geoserver.org/display/GEOS/Welcome.

10. OSGeo Live. [En línea] http://live.osgeo.org/es/overview/geoserver\_overview.html.

11. W3Schools. *SOAP Introduction.* [En línea] [Citado el: 9 de 5 de 2014.] http://www.w3schools.com/webservices/ws\_soap\_intro.asp.

12. W3Schools. *Introduction to WSDL.* [En línea] http://www.w3schools.com/webservices/ws\_wsdl\_intro.asp.

13. W3C MTOM http://www.w3.org/TR/soap12-mtom/

14. WS-Security https://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

15. SAML https://www.oasis-open.org/committees/tc\_home.php?wg\_abbrev=security

16. TLS http://es.wikipedia.org/wiki/Transport\_Layer\_Security

1. Imagen tomada de [1]. [↑](#footnote-ref-1)
2. Imagen tomada de [1]. [↑](#footnote-ref-2)
3. Imagen tomada de [1]. [↑](#footnote-ref-3)
4. Cita tomada de [2]. [↑](#footnote-ref-4)
5. Imagen tomada de [2]. [↑](#footnote-ref-5)