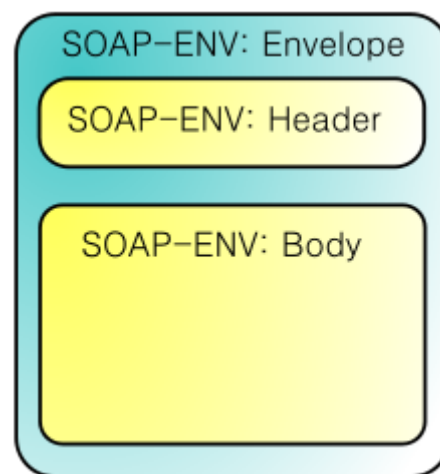


# Simple Object Access Protocol

**SOAP** (originalmente las siglas de *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por Dave Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros. Está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.



Estructura de un mensaje SOAP

## Índice

### Características

### Historia

### Estructura del mensaje

### Modelo de procesado

### SOAP sobre correo electrónico

### Ejemplos de mensajes SOAP

### Ventajas y desventajas

#### Ventajas

#### Desventajas

### Casos de uso

### Implementación de un servicio web SOAP

### Véase también

### Referencias

#### Recomendaciones de W3C

#### Artículos

### Enlaces externos

## Características

SOAP es un **paradigma de mensajería de una dirección sin estado**, que puede ser utilizado para formar protocolos más completos y complejos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "pila de protocolos de web service", ofreciendo un framework de mensajería básica en el cual los web services se pueden construir. Este protocolo está basado en XML y se conforma de tres partes:

- **Sobre (envelope):** el cual define qué hay en el mensaje y cómo procesarlo.
- **Conjunto de reglas de codificación** para expresar instancias de tipos de datos.
- **La Convención** para representar llamadas a procedimientos y respuestas.

El protocolo SOAP tiene tres características principales:

- **Extensibilidad** (seguridad y WS-routing son extensiones aplicadas en el desarrollo).
- **Neutralidad** (bajo protocolo de transporte TCP puede ser utilizado sobre cualquier protocolo de aplicación como HTTP, SMTP o JMS).
- **Independencia** (permite cualquier modelo de programación).

Como ejemplo de cómo el modelo SOAP pueda ser utilizado, consideraremos un mensaje SOAP que podría ser enviado a un web service para realizar la búsqueda de algún precio en una base de datos, indicando para ello los parámetros necesitados en la consulta. El servicio podría retornar un documento en formato XML con el resultado, un ejemplo, precios, localización o características. Teniendo los datos de respuesta en un formato estandarizado procesable (en inglés "parsable"), éste puede ser integrado directamente en un sitio Web o aplicación externa.

La arquitectura SOAP está formada por varias capas de especificación: MEP (*Message Exchange Patterns*) para el formato del mensaje, enlaces subyacentes del protocolo de transporte, el modelo de procesamiento de mensajes, y la capa de extensibilidad del protocolo. SOAP es el sucesor de XML-RPC, a pesar de que toma el transporte y la neutralidad de la interacción, así como el *envelope / header / body*, de otros modelos (probablemente de WDDX).

## Historia

---

La preocupación por los sistemas distribuidos y de cómo diferentes máquinas podían comunicarse entre sí surgió en la década de los 90. Hasta ese momento, era suficiente con que las aplicaciones de un mismo ordenador pudieran establecer una comunicación.

En 1990, surgieron los modelos COM y CORBA. El primero, Component Object Model fue creado por Microsoft, y el segundo, CORBA, por el Object Management Group. No obstante, estos dos modelos presentaban una dificultad muy importante: no eran fácilmente interoperables ya que las dos máquinas que llevaran a cabo la comunicación debían soportar COM o CORBA, por tanto únicamente se podía utilizar con dos máquinas COM o dos máquinas CORBA. Más adelante, Microsoft creó DCOM y Sun, RMI (Remote Method Invocation). Aunque estos métodos permitían establecer una conexión entre ordenadores a través de la red, tampoco eran interoperables ya que RMI está disponible únicamente para Java, y por tanto, es dependiente del lenguaje de programación. Por todo ello, Microsoft empezó a interesarse por la computación distribuida basada en XML en el año 1997. Su objetivo era terminar con los problemas de interoperabilidad de las soluciones anteriores y permitir que las aplicaciones se conectaran mediante RPCs (Remote Procedure Calls), utilizando los estándares de comunicación XML y HTTP.

SOAP fue diseñado como un protocolo de acceso a objetos en 1998 por Dave Winer, Don Box, Bob Atkinson y Mohsen Al-Ghosein por Microsoft, donde Atkinson y Al-Ghosein trabajaban en aquel entonces. La especificación SOAP actualmente es mantenida por el XML Protocol Working Group del World Wide Web Consortium.

La versión SOAP 1.1 se presentó en el año 2000 e IBM participó en su creación. Esta participación resultó muy positiva ya que se produjeron cambios significativos y cruciales para su posterior uso: se diseñó de una forma más modular y escalable, eliminando los problemas derivados de una tecnología propietaria, en este caso de Microsoft. Además, IBM llevó a cabo una implementación de SOAP en Java y SOAP se integró en Web Services J2EE.

**SOAP** originalmente significaba "Simple Object Access Protocol", pero esta sigla se abandonó con la versión 1.2 de la norma. La versión 1.2 se convirtió en una recomendación del W3C el 24 de junio de 2003. El acrónimo se confunde a veces con SOA, siglas de arquitectura orientada a servicios, pero las

siglas no están relacionados.

Después que SOAP se introdujo por primera vez, se convirtió en la capa subyacente de un conjunto más complejo de los web services, basada en la WSDL (Web Services Description Language) y UDDI (Universal Description Discovery and Integration). Estos servicios, especialmente UDDI, han demostrado ser de mucho menos interés, pero una apreciación de ellos da una comprensión más completa del esperado rol de SOAP comparado a como los web services están actualmente desarrollados.

## Estructura del mensaje

---

Un mensaje SOAP es un documento XML ordinario con una estructura definida en la especificación del protocolo. Dicha estructura la conforman las siguientes partes:

- **Envelope (obligatoria):** raíz que de la estructura, es la parte que identifica al mensaje SOAP como tal.
- **Header:** esta parte es un mecanismo de extensión ya que permite enviar información relativa a cómo debe ser procesado el mensaje. Es una herramienta para que los mensajes puedan ser enviados de la forma más conveniente para las aplicaciones. El elemento "Header" se compone a su vez de "**Header Blocks**" que delimitan las unidades de información necesarias para el header.
- **Body (obligatoria):** contiene la información relativa a la llamada y la respuesta.
- **Fault:** bloque que contiene información relativa a errores que se hayan producido durante el procesamiento del mensaje y el envío desde el "SOAP Sender" hasta el "Ultimate SOAP Receiver".

En los próximos apartados de este documento se podrá apreciar esta estructura con ejemplos concretos.

## Modelo de procesamiento

---

El modelo de procesamiento de SOAP está definido como un sistema distribuido, en el que intervienen diferentes nodos. En un escenario básico, los nodos SOAP se comunican uno asumiendo el rol de **SOAP Sender** y otro el de **SOAP Receiver**. Aun así, la especificación define diferentes tipos de nodos en función del rol que asumen en el envío del mensaje:

- **SOAP Sender:** nodo que transmite un mensaje SOAP.
- **SOAP Receiver:** nodo que acepta un mensaje.
- **SOAP message path:** conjunto de nodos por los cuales debe pasar un mensaje SOAP, incluyendo el nodo inicial, cero o más nodos intermediarios y el SOAP Receiver definitivo.
- **Initial SOAP sender:** el "sender" que origina el mensaje y que es el punto de inicio del camino que seguirá el mensaje.
- **SOAP intermediary:** el intermediario actúa como SOAP receiver y como SOAP sender, ya que primero recibe el mensaje para después reenviarlo al siguiente nodo en el camino.
- **Ultimate SOAP receiver:** destino final del mensaje SOAP, es el responsable de procesarlo. Cabe destacar que el mensaje podría no llegar al receptor definitivo debido a que problemas en los intermediarios hagan que se pierda.

Un nodo SOAP puede actuar con uno o varios roles, cada uno de los cuales se encuentra definido mediante una URI conocida como el nombre de rol. Los roles asumidos por un nodo son invariantes durante el envío de un mensaje, teniendo en cuenta la especificación el procesamiento individual de mensajes. Tal y como se ha comentado, una aplicación puede crear protocolos de comunicación más complejos como capas superiores sobre SOAP, pudiendo definir sus propios roles para poder cumplir con sus necesidades.

La especificación de SOAP define unas normas sobre cómo deben ser procesados, definiendo una serie de pasos que deben cumplir las implementaciones del protocolo. Estos pasos se pueden encontrar en la sección 2.6 de la especificación de W3C (<http://www.w3.org/TR/2007/REC-soap12-part1-20070427/#procsoapmsg>).

## SOAP sobre correo electrónico

Los desarrolladores de aplicaciones hoy en día, pueden utilizar la infraestructura de correo electrónico de Internet para transmitir mensajes SOAP ya sea como mensajes de correo electrónico de texto o como adjuntos. Los ejemplos que se muestran a continuación muestran un modo de transmitir mensajes SOAP, y deben ser tomados como el modo estándar de hacerlo. Las especificaciones SOAP Versión 1.2 no especifican tal vínculo. Sin embargo, existe una Nota W3C no-normativa [SOAP Email Binding] que describe un vínculo de SOAP con el correo electrónico. Su propósito principal es comenzar a demostrar la aplicación de la Infraestructura general de Vínculos con el Protocolo SOAP.

El ejemplo muestra el mensaje de petición de reserva de viaje del Ejemplo 1 transportado como un mensaje de correo electrónico entre un agente de usuario remitente y un agente de usuario destinatario. Está implícito que el nodo destinatario tiene capacidad para entender SOAP, por lo que se envía el mensaje de correo electrónico para su procesamiento. (Se asume que también el nodo remitente puede manejar errores SOAP que pudiera recibir en la respuesta o correlacionar cualesquiera mensajes SOAP recibidos en respuesta a éste).

### Ejemplo

```
De: a.oyvind@miempresa.example.com
A: reservas@empresaviajes.example.org
Asunto: Viaje a LA
Fecha: Thu, 29 Nov 2001 13:20:00 EST
Message-Id: <EE492E16A090090276D208424960C0C@miempresa.example.com>
Content-Type: application/soap+xml
```

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reserva xmlns:m="http://www.example.org"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <m:referencia>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:referencia>
      <m:fechaYHora>2001-11-29T13:20:00.000-05:00</m:fechaYHora>
    </m:reserva>
    <n:pasajero xmlns:n="http://miempresa.example.com/empleados"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <n:nombre>Åke Jógvan Øyvind</n:nombre>
    </n:pasajero >
  </env:Header>
  <env:Body>
    <p:itinerario xmlns:p="http://empresaviajes.example.org/reserva/viaje">
      <p:ida>
        <p:salida>Nueva York</p:salida>
        <p:llegada>Los Angeles</p:llegada>
        <p:fechaSalida>2001-12-14</p:fechaSalida>
        <p:horaSalida>última hora de la tarde</p:horaSalida>
        <p:preferenciaAsiento>pasillo</p:preferenciaAsiento>
      </p:ida>
      <p:vuelta>
        <p:salida>Los Angeles</p:salida>
        <p:llegada>Nueva York</p:llegada>
        <p:fechaSalida>2001-12-20</p:fechaSalida>
        <p:horaSalida>media-mañana</p:horaSalida>
        <p:preferenciaAsiento />
      </p:vuelta>
    </p:itinerario>
  </env:Body>
</env:Envelope>
```

```
</p:itinerario>
<q:alojamiento xmlns:q="http://empresaviajes.example.org/reserva/hoteles">
  <q:preferencia>ninguna</q:preferencia>
</q:alojamiento>
</env:Body>
</env:Envelope>Mensaje SOAP del Ejemplo 1 transportado como un mensaje SMTP
```

El encabezado del Ejemplo tiene la forma estándar de [RFC 2822] para mensajes de correo electrónico.

Aunque el correo electrónico es un intercambio de mensajes en un solo sentido, y no se da ninguna garantía de entrega, infraestructuras como la de la especificación Simple Mail Transport Protocol (SMTP) ofrecen un mecanismo de notificación de entrega que, en el caso de SMTP, se denominan Delivery Status Notification (DSN) [Notificación de Estado de Entrega] y Message Disposition Notification (MDN) [Notificación de Disposición de Mensaje]. Estas notificaciones toman la forma de mensajes de correo electrónico enviados a la dirección de correo electrónico especificada en el encabezamiento del mensaje de correo. Las aplicaciones, así como los usuarios finales del correo, pueden utilizar estos mecanismos para proporcionar el estado de una transmisión de correo electrónico, pero estos, si existiesen, serían notificaciones al nivel SMTP. El desarrollador de aplicaciones debe comprender completamente las capacidades y limitaciones de estas notificaciones de entrega o el riesgo de asumir que haya existido una entrega del mensaje con éxito cuando podría no haberse producido.

Los mensajes de estado de entrega SMTP son separados del procesamiento del mensaje en la capa SOAP. Las respuestas SOAP resultantes a los datos SOAP serán devueltas a través de un mensaje de correo electrónico nuevo que podría tener o no un enlace con el mensaje de la petición original al nivel SMTP. El uso del encabezado In-reply-to: [En-respuesta-a] según [RFC 2822] puede conseguir una correlación al nivel SMTP, pero no implica necesariamente una correlación al nivel SOAP.

## Ejemplos de mensajes SOAP

Como ejemplo se muestra la forma en que un cliente solicitaría información de un producto a un proveedor de servicios Web:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productId>827635</productId>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

Y esta sería la respuesta del proveedor:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Toptimate 3-Piece Set</productName>
        <productId>827635</productId>
        <description>3-Piece luggage set. Black Polyester.</description>
        <price>96.50</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

A pesar de no ser la única opción posible, HTTP fue elegido como protocolo de transporte por sus ventajas, para lidiar con cortafuegos, por ejemplo. Otros protocolos como GIOP/IIOP o DCOM (utilizados en CORBA, RMI y DCOM) suelen ser repelidos por estos cortafuegos.

## Ventajas y desventajas

---

### Ventajas

- Debido al uso de XML permite invocar procedimientos remotos de muchos lenguajes, por lo tanto, presenta una gran interoperabilidad.
- Al utilizar una comunicación vía HTTP es fácilmente escalable, además de ser casi siempre permitido por los cortafuegos.
- Puede ser implementado utilizando cualquier lenguaje y ejecutado en cualquier plataforma.
- Es posible utilizarlo mediante usuario anónimo y mediante autenticación.
- Es posible transmitirlo mediante cualquier protocolo de transporte capaz de transmitir texto, típicamente HTTP o SMTP.

### Desventajas

- Debido al uso de XML para el paso de mensajes, SOAP es considerablemente más lento que otros middleware como CORBA ya que los datos binarios se codifican como texto. Para contrarrestar este punto débil en el caso de XML con código binario incrustado se desarrolló un método optimizado de transmisión de mensajes.
- Depende del WSDL (Web Services Description Language).
- Al contrario que Java, PHP o Python ciertos lenguajes no ofrecen un apoyo adecuado para su uso ya sea a nivel de integración o de soporte IDE.

## Casos de uso

---

La forma más habitual de utilizar el protocolo SOAP es mediante el patrón **petición-respuesta** con remitente SOAP y destinatario final SOAP, el cual es utilizado cuando los mensajes SOAP están predefinidos y únicamente se desea enviar una petición y consultar su valor de retorno.

No obstante, muchas veces este patrón no es suficiente, y es necesario establecer un intercambio múltiple de mensajes entre los nodos. La W3C define dos tipos de intercambios de mensajes SOAP para formar una conversación:

- **Intercambio de mensajes Conversacionales:** permite redefinir la información de la petición. Estos intercambios pueden acabar comportándose como un patrón de mensajes de ida y vuelta.
- **Llamadas a Procedimientos Remotos:** permite encapsular la funcionalidad de procedimientos remotos utilizando las ventajas de XML de extensibilidad y funcionalidad, por este motivo se ha definido en la especificación una representación uniforme para realizar invocaciones y respuestas RPC mediante mensajes SOAP.

En ocasiones, es necesario el uso de intermediarios en las comunicaciones SOAP, la especificación SOAP 1.2 define dos tipos:

- **Intermediario redirector:** se trata de un nodo SOAP, el cual redirige el mensaje SOAP a otro nodo SOAP según lo establecido en un bloque de encabezado que ha recibido el nodo destino o según el patrón de mensajes en uso.
- **Intermediario activo:** realiza un procesamiento adicional del mensaje SOAP antes de redirigirlo, sin utilizar criterios descritos en el encabezado del mensaje o del patrón de mensajes en uso.

## Implementación de un servicio web SOAP

---


Todos los lenguajes de uso mayoritario en el desarrollo de sistemas web implementan o incluyen algún tipo de soporte para la implementación tanto de servicios web SOAP como de los clientes que los consumen. Además de bibliotecas que implementan el protocolo a nivel básico, encontramos otras que implementan diferentes escenarios de uso y establecen interfaces más sencillas simplificando la programación.

Estas bibliotecas, utilizadas en conjunto con frameworks de desarrollo de sistemas web agilizan el proceso de desarrollo tanto del servicio web como de sus clientes, en especial si se genera un fichero WSDL que comunique a los clientes las características del servicio.

- **JAVA:** dentro de su biblioteca estándar se encuentran implementaciones concretas a las que se da **soporte oficial**. También podemos encontrar bibliotecas de terceros que, tal y como se ha comentado, ayudan al desarrollador simplificando las interfaces e implementando los casos de uso más habituales. Cabe destacar que los **entornos de desarrollo más utilizados ofrecen soporte para la creación de servicios web SOAP** que, entre otras cosas, **generan automáticamente el fichero WSDL** y permiten diseñar de forma visual el API y las llamadas que contendrá. En cuanto el servidor a utilizar, se pueden considerar las opciones típicas en Java: Tomcat, Glassfish, etc. Aun así, la elección del servidor puede suponer algunas ventajas, por ejemplo, Glassfish genera una sencilla interfaz web para probar las diferentes llamadas del servicio. Además, la mayoría de herramientas permiten la generación del cliente del servicio automáticamente a partir de su fichero WSDL (<https://netbeans.org/kb/docs/websvc/jax-ws.html>).
- **PHP:** ofrece soporte y unas bibliotecas de apoyo habilitando la **extensión SOAP** (<http://www.php.net/manual/es/book.soap.php>) en el servidor. Se ha desarrollado un gran número de librerías de terceros (<https://packagist.org/search/?q=SOAP>), que combinadas con el uso de frameworks MVC, simplifican las interfaces e implementan los escenarios de uso más habituales. También son habituales las implementaciones de clientes para servicios web públicos concretos.
- **Python:** no ofrece un soporte en sus bibliotecas estándar, sin embargo, existe un gran número de paquetes de terceros que permiten la implementación de servicios web SOAP y sus clientes. En el ámbito del desarrollo de servicios web en Python, predomina la utilización del **Framework Django** (<https://www.djangoproject.com/>) que se puede combinar con cualquiera de las implementaciones de SOAP.
- **.NET:** dentro del Framework se ofrecen herramientas similares a las de Java para el diseño visual del servicio y la creación automática de WSDL. También da soporte para la creación de los clientes a partir del fichero de definición del servicio. En el caso de .NET, el entorno de desarrollo destacado es Visual Studio. En cuanto a bibliotecas encontramos que el ecosistema .NET ofrece múltiples opciones en varios lenguajes, aunque la apuesta actual de Microsoft para el desarrollo web es su **Framework .NET MVC**. Se debe tener en cuenta, que Microsoft creó el formato **Windows Communication Foundation** (<http://msdn.microsoft.com/es-es/vstudio/aa663324.aspx>) que es un modelo para la creación de sistemas orientados a servicios, similar y complementario al WSDL.

## Véase también

---

-  Portal:Informática. Contenido relacionado con **Informática**.
- XML-RPC: protocolo antecesor de SOAP.
- Formulario web
- Pasarela de firma
- Resaltado de sintaxis
- Servicio web
- WS-Security
- WSDL

## Referencias

---

### Recomendaciones de W3C

- W3C Recommendation (27 April 2007). *SOAP Version 1.2 Part 0: Primer (Second Edition)* (<http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>). W3C (en inglés)
- W3C Recommendation (27 April 2007). *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)* (<http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>). W3C (en inglés)
- W3C Recommendation (27 April 2007). *SOAP Version 1.2 Part 2: Adjuncts (Second Edition)* (<http://www.w3.org/TR/2007/REC-soap12-part2-20070427/>). W3C (en inglés)
- W3C Recommendation (27 April 2007). *SOAP Version 1.2 Specification Assertions and Test Collection (Second Edition)* (<http://www.w3.org/TR/2007/REC-soap12-testcollection-20070427/>). W3C (en inglés)
- W3C Recommendation (25 January 2005). *SOAP Message Transmission Optimization Mechanism* (<http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/>). W3C (en inglés)
- W3C Recommendation (25 January 2005). *Resource Representation SOAP Header Block* (<http://www.w3.org/TR/2005/REC-soap12-rep-20050125/>). W3C (en inglés)
- Recomendación del W3C (24 de junio de 2003). *SOAP Versión 1.2 Parte 0: Fundamentos* (<http://www.w3c.es/Traducciones/es/TR/2003/REC-soap12-part0-20030624/>). W3C. (en español)

### Artículos

- Joshy Joseph, Software Engineer, IBM Software Group. (01 Feb 2002). *Handling attachments in SOAP* (<http://www.ibm.com/developerworks/webservices/library/ws-soapatt/?open&l=930,t=grws>). IBM (en inglés)

## Enlaces externos

---

- XML Protocol Working Group (<http://www.w3.org/2000/xp/Group/>) W3C Architecture Domain (en inglés)
- XML protocol activity (<http://www.w3.org/2000/xp/>) (en inglés)
- Technology Report (<http://xml.coverpages.org/soap.html>) (en inglés)
- two-way SOAP to CORBA bridge (<http://soap2corba.sourceforge.net/>) (en inglés)



- [Discussion on Web Services technology \(SOAP and REST\) \(http://www.advogato.org/article/464.html\)](http://www.advogato.org/article/464.html) (en inglés)
- [Web del creador de XML-RPC \(precursor de SOAP\) \(http://www.userland.com/\)](http://www.userland.com/)
- [What is SOAP? Why is SOAP required? Advantages of SOAP \(http://careerride.com/SOAP-What-is-SOAP.aspx\)](http://careerride.com/SOAP-What-is-SOAP.aspx) (en inglés)

---

Obtenido de «[https://es.wikipedia.org/w/index.php?title=Simple\\_Object\\_Access\\_Protocol&oldid=136277209](https://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&oldid=136277209)»

---

**Esta página se editó por última vez el 12 jun 2021 a las 15:14.**

El texto está disponible bajo la Licencia Creative Commons Atribución Compartir Igual 3.0; pueden aplicarse cláusulas adicionales. Al usar este sitio, usted acepta nuestros términos de uso y nuestra política de privacidad. Wikipedia® es una marca registrada de la Fundación Wikimedia, Inc., una organización sin ánimo de lucro.