

# GeoBovino: Un ejemplo de Geo-trazabilidad

M.A. Manso Callejo  
M. Núñez Jiménez

Universidad Politécnica de Madrid, ETSI en Topografía, Geodesia y Cartografía:  
LatinGEO (Laboratorio de Tecnológicas de la Información Geográfica)  
Autovía de Valencia Km 7, 28031 Madrid (España)  
[m.manso@upm.es](mailto:m.manso@upm.es)

## Resumen

GeoBovino nace como una iniciativa que pretende ofrecer un servicio sencillo para consultar los datos referentes a la trazabilidad del ganado bovino.

Por trazabilidad se entiende: *“aquellos procedimientos preestablecidos y autosuficientes que permiten conocer el histórico, la ubicación y la trayectoria de un producto o lote de productos a lo largo de la cadena de suministros en un momento dado, a través de unas herramientas determinadas”*. [1]

La información sobre la trazabilidad animal esta legislada, y existen normativas para el correcto etiquetado y seguimiento del animal, pero el problema es hacerle llegar esa información tanto a consumidores como a profesionales.

La finalidad de Geobovino es paliar ese problema, permitiendo recuperar la información sobre el animal o piezas del mismo. Para ello el proyecto se basa en todo ese trabajo previo para conseguir los datos de la trazabilidad del animal.

GeoBovino ha sido concebido como un servicio Web que, a través de Internet, permite la recuperación de la información almacenada por los distintos organismos y entidades que se ocupan de la trazabilidad del ganado bovino. El servicio permite recuperar toda la información disponible a partir del crotal del animal (identificador), el cual se halla presente en las etiquetas de canal y corte según el estándar de etiquetado UCC-EAN-128. [2]

## 1. Introducción

La trazabilidad de la carne de vacuno en España está garantizada por distintas directivas de rango comunitario ((CE) num.1760/2000, (CE) num. 1825/2000) y otras disposiciones adicionales a nivel nacional (Real Decreto 1698/2003 de 12 de diciembre). Estas normativas definen como deben identificarse las reses y como debe ser etiquetada toda la carne o los productos derivados de la misma. Aunque

esta legislación posibilita identificar el animal al que pertenece el producto cárnico, no se definen normas para documentar la vida del animal antes de su sacrificio.

Geo-Bovino se ha definido como un prototipo de Servicio de Información en la Web que proporciona información alfanumérica, gráfica, descriptiva y enlaces en relación con la trazabilidad cárnica, sanitaria y geográfica de un producto cárnico derivado del vacuno. En el prototipo de Servicio "Geo-Bovino" se han modelado las diferentes fuentes de información que gestionan los distintos actores (administración, ganaderos e industrias cárnicas), en relación a un producto cárnico de vacuno. Posteriormente, cada una de las fuentes de información es gestionada por un servicio web específico. Estos servicios tienen la finalidad de flexibilizar el modelo arquitectónico del prototipo hacia un sistema distribuido en la Web. Como consecuencia, se proporciona una mayor independencia en la gestión y explotación de la información. El prototipo, desarrollado en Java, se encarga de coordinar las consultas y procesar los resultados, para mostrarlos en un simple visor de páginas Web, de acuerdo con unos estilos de visualización definidos.

La componente geográfica de la trazabilidad se gestiona mediante pares de coordenadas geográficas asociadas a una fecha ya sea la de nacimiento, movimiento o sacrificio del animal. El servicio Web, que proporciona dicha información, codifica las coordenadas y el atributo fecha en un archivo codificado en GML. Esta traza representa los movimientos registrados del animal a lo largo de su vida. La visualización de dicha traza se realiza de forma conjunta con otras fuentes de información, en forma de cliente ligero de servicios de mapas (WMS). De este modo los usuarios pueden navegar y acercarse en la medida de lo posible a la ubicación de la explotación ganadera por la que paso la res [3,4]. El prototipo diseñado tiene por objeto ser una maqueta funcional y operativa que demuestre como se podría proporcionar a los consumidores, de una forma amigable, más información sobre todos los productos cárnicos de vacuno que consumen.

El resto del documento se estructura de la siguiente forma: en primer lugar se hace una aproximación del problema que se pretende resolver, después se describe la estructura que se le ha dado al prototipo, para pasar seguidamente a describir su funcionamiento. Por último se valoran los resultados obtenidos y se finaliza con las conclusiones y los agradecimientos.

## **2. Aproximación al problema**

Toda la información sobre trazabilidad animal, necesaria para poder ofrecer este servicio, se halla en manos del ministerio de Agricultura Pesca y Alimentación

(MAPA) o en el de las Comunidades Autónomas (CCAA), de los ganaderos, de las asociaciones de ganaderos, de los centros de sacrificio y despiece, así como de los establecimientos de venta.

El principal problema desde el punto de vista del consumidor reside en que el etiquetado, que transporta la información necesaria, se ajusta a un estándar hasta que llega a los establecimientos de venta y una vez allí cada comercio gestiona internamente la relación de cada producto que pone a la venta con el canal o pieza suministrada y la mayor parte de la información presente en el etiquetado no es accesible para el consumidor. Por esta razón no es viable aplicar la recuperación de información desde los datos presentes en las etiquetas finales de venta al público (bandeja con un producto cárnico). Por este motivo, el proyecto GeoBovino sólo contempla el seguimiento del animal desde su registro al nacer, hasta su sacrificio y despiece.

Dado que existe una serie de entidades encargadas de gestionar la información de los animales, el primer paso fue comprender el cometido de cada uno de estos actores:

- El MAPA o las CCAA, mantienen la información de los traslados del animal (SIMOGAN) y el registro de explotaciones ganaderas (REGA).
- Los centros cárnicos, que llevan la información del sacrificio y despiece de los animales.
- Por último la asociación de ganaderos y los propios ganaderos, que llevan toda la información de los acontecimientos que le suceden al animal en el día a día. Se decidió unir ambos actores al apreciar que la información que gestiona la asociación de ganadera y los ganaderos son muy dependientes entre sí.

Los siguientes pasos tras definir dichos actores, fueron definir los modelos de datos de cada una de las entidades y crear un servicio Web autónomo que permitiera la recuperación de la información para cada una de las entidades de manera independiente.

## **2.1) Base de datos**

Se ha utilizado como sistema gestor de bases de datos (SGDB) PostgreSQL version 8.14, y se han creado 3 bases de datos distintas, simulando cada una de ellas a uno de los diferentes organismos o entidades anteriormente citados. En cada uno de ellos se almacena información de la vida del animal. Estas tres bases de datos podrán estar en lugares físicos y/o lógicos distintos, pero la información de cada una de ellas está relacionada con las demás por el crotal del animal [5].

- 1) Movimientos y explotaciones ganaderas: En ella se registran las coordenadas geográficas de las explotaciones ganaderas, y se asocia a un desplazamiento o traslado en base a esas coordenadas y el crotal del animal.
- 2) Asociación de ganadería y ganadero: Aquí se registra toda la información del animal, desde las características propias, pasando por tratamientos y controles sanitarios realizados, hasta su alimentación.
- 3) Centro cárnico de sacrificio y despiece: En ella se almacena la información relativa al sacrificio y al despiece del animal, reflejando además que pieza proviene de que animal.

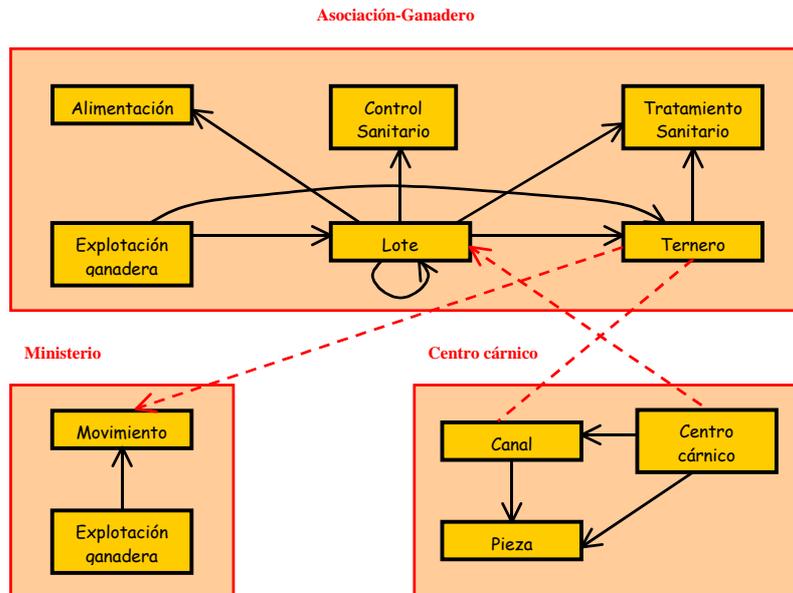


Figura 1. Esquema de la base de datos.

## 2.2) Servicios web

“Un servicio web, es un recurso software que se ejecuta en un servidor web remoto, en respuesta a la solicitud hecha por un cliente. Los servicios web son equivalentes a cualquier aplicación que corre en un equipo local, sólo que la información necesaria para llevar a cabo una tarea específica es enviada al servidor y el resultado de esa tarea, devuelto al usuario” [6].

Los servicios Web se han implementado usando Java, el JDK version 5.0 y el servidor de aplicaciones Tomcat de apache versión 5.5 [7, 8].

En base a la idea de dar independencia a cada uno de los actores participantes en la trazabilidad, se diseñó un servicio Web por cada actor y después uno adicional que pretende dar un servicio global basándose en la información de los tres, y además permite un acceso más sencillo a estos servicios Web en los que se apoya.

### **2.2.1) Servicios Web basados en SOAP**

*“SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar creado por el W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.”* [9,10].

Se han creado tres servicios Web, cada uno de ellos se encarga del acceso y recuperación de la información de su correspondiente base de datos. De esta manera cada servicio Web representa a cada una de las entidades y ofrece un servicio de acceso a la información contenida en la base de datos de su entidad.

Cada uno de los servicios permite hacer peticiones (basadas en SOAP) desde cualquier programa que implemente este sistema, pudiendo invocar los métodos expuestos por el servicio para obtener la información correspondiente de la entidad.

### **2.2.1) Servicio Web global**

Este servicio, no es propiamente un servicio Web sino una aplicación que se ejecuta en servidor y permite el uso de los servicios Web anteriormente descritos. Está implementado de tal forma que simula un servicio Web, haciendo el papel de puerta de entrada a los 3 servicios Web basados en SOAP.

Este servicio acepta peticiones por URL y mediante XML usando los métodos POST y GET. Estas peticiones deben ser realizadas según unas reglas específicas de construcción, pero se puede consultar la forma de usarlo mediante la petición `getCapabilities` del servicio. El resultado de esta petición es la información que describe el acceso al servicio codificada en XML.

Entre las peticiones que es capaz de procesar, las dos más importantes son los métodos de acceso global a la información de los servicios Web descritos, ya que cada una de esas peticiones resulta a su vez en otras tres peticiones sobre los servicios Web SOAP, y el posterior procesamiento de la información de respuesta para generar la respuesta global.

### **2.3) Aplicación Web de consulta**

Se ha implementado una aplicación Web que tras visitar la página introductoria del servicio, nos permite elegir entre 2 opciones de consulta:

#### **2.3.1) Servicio básico:**

Esta página Web en HTML presenta un pequeño formulario que permite realizar peticiones al servicio global, pudiendo invocar los 2 métodos distintos que presenta, el de acceso a la información de canal o el de acceso a la información de pieza del animal [11]; para ello solo hay que introducir el crotal del animal y el número de pieza de modo que se recupera la información del animal en formato de página web.

#### **2.3.2) Servicio avanzado:**

Esta segunda página Web utilizando programada en javascript, y que muestra un formulario mas complejo, que permite atacar al servicio global o a los servicios Web, incluso permite invocar métodos más concretos [12]. Además de introducir el crotal y la pieza, se debe de indicar el formato de salida ya que se permite solicitar la información en formato XML ó en forma de página Web. Este servicio también permitirá ver un ejemplo de construcción de petición URL o XML construida a partir de los datos seleccionados en el formulario. De esta forma se dota al sistema final de una interfaz sencilla para aprender a explotar el servicio.

### **3) Proceso de consulta**

#### **3.1) Servicios Web basados en SOAP**

Los servicios Web basados en SOAP permiten realizar solicitudes externas a los mismos utilizando la metodología que impone SOAP. Esta implementación de los servicios los deja abiertos para su uso desde otras aplicaciones que implementen este sistema de mensajería, de modo que se pueda acceder a los métodos públicos de cada uno de los servicios Web. Esto se realiza importando las librerías adecuadas de Axis (herramienta de trabajo basada en SOAP y WSDL) que pone a disposición de los programadores una serie de librerías, que contienen métodos para realizar los accesos a los servicios Web. Por otro lado “*WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo*” [13].

### 3.2) Servicio Web global

Si lo que se desea es el acceso al servicio global, o simplemente acceder a los servicios independientes pero de una manera más sencilla, entonces se puede acceder utilizando un navegador:

- Utilizando la página Web de consulta con dos opciones (básica y avanzada).
- Enviando parejas de parámetros-valores (GET o POST) como URL:  
<http://localhost:8080/GEOBOVINO/servlet/Principal?service=SWMinisterio&request=getMovimientos&version=1.0&crotal=501&output=xml>
- Enviando un texto en XML (GET o POST) como URL en el navegador:  
[http://localhost:8080/GEOBOVINO/servlet/Principal?xml=<?xml version="1.0" encoding="UTF-8"?><trazability xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="file:C:\\...\\peticion.xsd"> <service version="1.0">SWMinisterio</service><request>getMovimientos </request><parameters crotal="501" /><output>xml</output></trazability>](http://localhost:8080/GEOBOVINO/servlet/Principal?xml=<?xml version='1.0' encoding='UTF-8'?><trazability xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xsi:noNamespaceSchemaLocation='file:C:\\...\\peticion.xsd'> <service version='1.0'>SWMinisterio</service><request>getMovimientos </request><parameters crotal='501' /><output>xml</output></trazability>)

El acceso a los servicios implementados exige utilizar una serie de parámetros para su funcionamiento:

- Service, se trata del nombre del servicio que se desea utilizar: SWGeoBovino (servicio global), SWMinisterio (lleva la parte de movimientos del animal), SWCarnico (informa sobre despiece y sacrificio del animal), SWAsogan (informa sobre datos del ternero, alimentación y sanidad)
- Version, indica la versión del servicio.
- Request, es el nombre del método que se invoca.
- SWGeoBovino: *getCapabilities*, indica las capacidades del servicio, *getCanal*, recupera todos los datos de un animal. La resolución de esta petición se realiza llamando a los tres serviciosWeb, y llamando a sus métodos globales (*getMovimientos*, *getCarnico* y *getAsogan*), *getPieza*, recupera todos los datos de una pieza. La resolución de esta petición se realiza llamando a los tres serviciosWeb, y llamando a sus métodos globales (*getMovimientos*, *getCarnico* y *getAsogan*).
- SWMinisterio: *getCapabilities*, indica las capacidades del servicio, *getMovimientos*, recupera todos los traslados que ha hecho el animal, y las coordenadas geográficas de las explotaciones ganaderas de origen y destino.
- SWCarnico: *getCapabilities*, indica las capacidades del servicio, *getCarnico*, recupera toda la información tanto de sacrificio como de despiece, *getDespiece*, recupera la información relativa a la pieza, *getSacrificio*, recupera la información relativa al sacrificio.

- **SWAsogan:** *getCapabilities*, indica las capacidades del servicio, *getAsogan*, recupera toda la información del ternero, su alimentación e información sanitaria, *getTernero*, recupera los datos básicos del animal, *getCebo*, recupera la información relativa al cebo o alimentación del animal, *getSanidad*, recupera la información relativa a sanidad del animal, es decir, tratamientos y controles sanitarios.
- **Parámetros:** Son los parámetros propios de la consulta. El crotal siempre es obligatorio, y la pieza en cambio es opcional, debiendo aparecer solamente si se está invocando un método que nos devuelva información sobre ella.
- **Output:** Con este parámetro, se indica el formato en que queremos recibir la respuesta de la consulta: HTML, desde la aplicación Web, por defecto se devuelve este formato o XML, al acceder al servicio por peticiones URL este será el formato por defecto.

### 3.3) Funcionamiento del servicio

Una vez realizada la petición, el servicio Web lo procesa. La primera tarea que se realiza es la transformación de la petición a XML si es necesario. El siguiente paso es la validación de la petición mediante un XML-Schema, de manera que se valida la estructura las peticiones. Si algo fallase se devolvería un mensaje de error como respuesta.

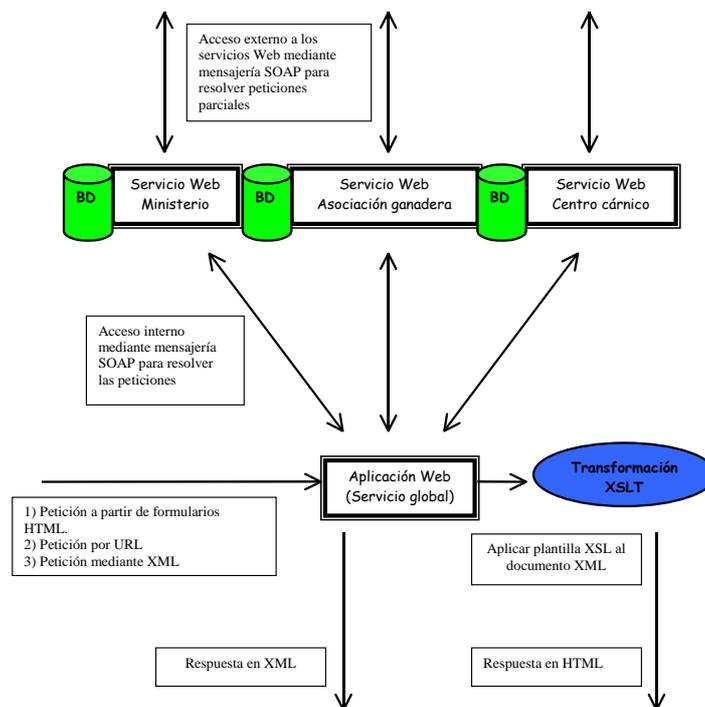


Figura 2. Esquema del prototipo.

Si la solicitud es correcta el servicio Web global atenderá a la petición. Si se trata de una petición a alguno de los servicios Web SOAP, el servicio global redirige la petición mediante los métodos de mensajería SOAP; si fuera una llamada a algún método del servicio global, éste realizaría peticiones a los tres servicios Web SOAP y recogería todas las respuestas para crear una respuesta compuesta por las tres respuestas de los servicios Web SOAP. Una vez creada la respuesta en XML y dependiendo del formato de salida seleccionado, el resultado se muestra directamente el XML o se toma el XML y se le aplica una plantilla XSL para que mediante un proceso XSLT se transforme ese fichero XML en una página Web de respuesta al usuario [15].

### **3.4) Interfaz del servicio Web global**

Como se indicó anteriormente no es estrictamente un servicio Web, y aunque implementa métodos públicos que orquestan las llamadas a otros servicios, es capaz de procesar peticiones dirigidas expresamente a él. Este servicio a diferencia de los otros tres, permite peticiones por paso de parámetro-valor a través de URL, e incluso mediante XML usando los métodos POST y GET. Estas peticiones deben ser realizadas según unas reglas de construcción, pero se puede consultar la forma de usarlo mediante la petición `getCapabilities` del servicio. Este servicio pone a disposición del usuario dos métodos de acceso global a la información de los servicios Web descritos, ya que procesa dos tipos de peticiones que resultan en tres peticiones sobre los servicios Web SOAP, y un posterior procesamiento de la información recogida para su presentación al usuario.

- A) `GetCanal`: Este método se apoya en los otros tres servicios Web, y devuelve el conjunto de las peticiones de los tres. Invoca los métodos de cada servicio Web que devuelven toda la información de cada servicio. Solamente se que no se incluye información en el apartado de despiece, debido a que se está solicitando información a nivel de canal y no se incluye un nombre de pieza que recuperar.
- B) `getPieza`: Este método también se apoya en los otros tres servicios Web, y devuelve el conjunto de las peticiones de los tres. Además incluye información en el apartado de despiece debido a que se está solicitando información a nivel de pieza, por lo cual se le proporciona un nombre de pieza para recuperar.

- C) **getCapabilities:** La petición `getCapabilities`, se responderá con un texto en formato XML que especifica las capacidades del servicio, es decir los métodos que se pueden invocar.

#### **4. Resultados**

Hasta la fecha el prototipo presenta las siguientes características:

Se ha diseñado una BBDD que no sólo atiende a las exigencias del problema, sino que se ha realizado de la manera mas genérica posible en perspectiva a futuros cambios o ampliaciones. El código fuente se ha estructurado notablemente para favorecer de igual manera, las posibles ampliaciones o modificaciones que puedan tener lugar en un futuro. Se han desplegado los 3 servicios Web basados en mensajería SOAP de manera satisfactoria dotando al sistema de independencia y autonomía. La estructura con la que se ha dotado al proyecto permite la explotación independiente de los servicios Web basados en SOAP, pero al mismo tiempo son la base para el funcionamiento del servicio web global. Se ha puesto a disposición de los usuarios un sistema de consulta muy sencillo a través de una página Web, de manera que sin comprender el funcionamiento sea fácil de usar ya que solo deben suministrarse los datos del animal y pieza del mismo. Se realiza la validación de las peticiones. Se ha conseguido devolver a los usuarios de los servicios Web los errores que ocurren: peticiones, validaciones, servicios y acceso a BBDD. Se ha dotado al prototipo de ficheros de configuración que permitan independencia en su implantación. El hecho de resolver las peticiones con respuestas escritas en lenguaje XML las hace especialmente compatibles por cualquier sistema, ya que es un estándar de libre distribución. Además para mayor facilidad del usuario se ha implementado una plantilla utilizando el lenguaje XSL que permite la transformación de las respuestas, pasando de XML a una página Web lo que hace que el contenido y presentación de las consultas al servicio global sean mas claras y completas.

#### **5. Conclusiones**

En la fase de análisis se ha detectado dificultades a la hora de integrar la información de los distintos centros cárnicos y asociaciones ganaderas, ya que cada asociación ganadera se realiza de manera independiente, gestionando su propia información, y por tanto se decidió enfocar el proyecto desde el punto de vista de la asociación ganadera. Por ello se decidió que existiría un servicio Web de consulta por cada asociación ganadera, conteniendo cada una de ellas la información sobre los ganaderos asociados, los centros cárnicos a los que se acude desde dicha asociación y además la información que almacenan las administraciones (Ministerio y CCAA).

El grado de desarrollo del prototipo está al 80% y aunque esta funcionando, aún tiene que someterse a un riguroso proceso de pruebas de funcionamiento, y además se le pretenden añadir las siguientes mejoras: La más destacable es la inclusión de un cliente WMS, incluir la metodología de Struts para dar mas consistencia y seguridad al espacio Web de la aplicación [16], añadir un esquema de despiece donde se resalte la pieza sobre la que se ha consultado, valorar la posibilidad de introducir un repositorio de fotografías de los animales y añadir estilos, para mejorar el aspecto del espacio Web de consulta mediante hojas de estilo CSS [17].

GeoBovino pretende además de poner a disposición de los consumidores y profesionales un espacio Web donde realizar sencillas consultas de la información del animal y más importante aún, ofrecer el soporte necesario para futuras aplicaciones que requieran de dicha información ofreciéndola en forma de Servicios Web.

## 6. Referencias

- [1] Concepto de Trazabilidad: <http://www.aecoc.es/web/codificacion.nsf/0/925B46B62071AAB5C1256F2E00506B2E?OpenDocument>
- [2] Etiquetado de ganado: [http://www.belt.es/articulos/fichas\\_prof/seg\\_inform/copia\\_seg\\_CD.htm](http://www.belt.es/articulos/fichas_prof/seg_inform/copia_seg_CD.htm)
- [3] Geography Markup Language (GML): [es.wikipedia.org/wiki/GML](http://es.wikipedia.org/wiki/GML)
- [4] Web Map Service(WMS): [es.wikipedia.org/wiki/WMS](http://es.wikipedia.org/wiki/WMS)
- [5] Sistema gestor de Bases de datos Postgre: [www.postgresql.org](http://www.postgresql.org)
- [6] Servicio Web: [fbio.uh.cu/bioinfo/glosario.html](http://fbio.uh.cu/bioinfo/glosario.html)
- [7] Lenguaje Java: [www.java.sun.com](http://www.java.sun.com)
- [8] Servidor de aplicaciones Tomcat: [www.tomcat.apache.org](http://www.tomcat.apache.org)
- [9] Mensajería SOAP: [es.wikipedia.org/wiki/SOAP](http://es.wikipedia.org/wiki/SOAP)
- [10] eXtensible Markup Language (XML): [es.wikipedia.org/wiki/XML](http://es.wikipedia.org/wiki/XML)
- [11] Javascript: [www.w3schools.com/js](http://www.w3schools.com/js)
- [12] Hyper Text Markup Language (HTML): <http://www.w3schools.com/html>
- [13] Descripción de Web Service WSDL: [es.wikipedia.org/wiki/WSDL](http://es.wikipedia.org/wiki/WSDL)
- [14] Tutoriales sobre XML-Schemas: [www.w3schools.com/schema](http://www.w3schools.com/schema)
- [15] Extensible Stylesheet Language (XSL): [www.w3.org/Style/XSL](http://www.w3.org/Style/XSL)
- [16] Paquete de seguridad para espacios Web, Struts: [www.struts.apache.org](http://www.struts.apache.org)
- [17] Cascading Style Sheets, (CSS): <http://es.wikipedia.org/wiki/Css>