

# Herramienta web para almacenar y visualizar objetos distribuidos

José M. Hernández, Carlos R. Jaimez

Universidad Autónoma Metropolitana, Unidad Cuajimalpa,  
Departamento de Tecnologías de la Información, México

2123064604@alumnos.cua.uam.mx, cjaimez@correo.cua.uam.mx

**Resumen.** En los últimos años ha habido un incremento en el desarrollo de aplicaciones que permiten la administración de diversos tipos de recursos a través de web, debido a que sus usuarios únicamente requieren de un navegador web para acceder a la aplicación. Los objetos distribuidos son un ejemplo del tipo de recursos que se beneficiarían de una aplicación que permitiera tener acceso a ellos en línea. En este artículo se presenta una herramienta web que permite almacenar objetos, así como visualizar su estado y métodos. Esta herramienta web es un complemento de *Web Objects in XML* (WOX), el cual es un *framework* para programación de aplicaciones distribuidas basadas en objetos.

**Palabras clave:** Objetos remotos, objetos distribuidos, web objects in XML, visualización de objetos, almacenamiento de objetos.

## Web Tool for Storing and Visualizing Distributed Objects

**Abstract.** In recent years there has been an increase in the development of applications that allow the administration of several types of resources through the web, because their users only require a web browser for accessing the application. Distributed objects are an example of the type of resources that would benefit of an application that allow to have online access to them. This paper presents a web tool that allows storing objects, as well as visualizing their state and methods. This web tool is a plug-in for *Web Objects in XML* (WOX), which is a *framework* for programming object-based distributed applications.

**Keywords.** Remote objects, distributed objects, web objects in XML, visualization of objects, object storage.

### 1. Introducción

La herramienta web que se presenta en este artículo es un complemento de *Web Objects in XML* (WOX) [1], el cual es un *framework* para programación de aplicacio-

nes distribuidas basadas en objetos. WOX permite construir sistemas distribuidos, utiliza XML como representación para los objetos y mensajes intercambiados [2], y proporciona comunicación síncrona y asíncrona entre clientes y servidores [3]. WOX tiene características especiales tomadas de dos paradigmas utilizados para construir sistemas distribuidos: el paradigma basado en objetos y el paradigma basado en web. En WOX es posible almacenar objetos Java y C#, los cuales pueden ser generados por aplicaciones locales o distribuidas.

El resto del artículo está organizado de la siguiente manera. El trabajo relacionado se presenta en la sección 2 con un análisis comparativo de diferentes herramientas que tienen un propósito similar al de la herramienta web que se describe en este artículo. En la sección 3 se proporciona una introducción del *framework* WOX. En la sección 4 se describe la funcionalidad de la herramienta web, su estructura e interfaz. La sección 5 se encarga de mostrar la herramienta web en funcionamiento, en particular se presenta el almacenamiento de objetos y el repositorio para visualización de objetos. Finalmente, en la sección 6 se presentan las conclusiones y el trabajo futuro.

## 2. Trabajo relacionado

En esta sección se describen las características y el funcionamiento de algunas herramientas que son similares a la herramienta web propuesta en este artículo. Las herramientas analizadas son las siguientes: CORBAWeb [4], SopView+ [5], PESTO [6], CORBA Object Browser [7] y Apache Axis2 [8]. Al final de la sección se presenta una comparación de estas herramientas, junto con una breve descripción de las características que se tomaron en cuenta.

### 2.1. CORBAWeb

Es una herramienta [4] que actúa como una puerta de enlace entre la web y CORBA (*Common Object Request Broker Architecture*, CORBA por sus siglas en inglés). A esta herramienta se le conoce como un navegador de objetos genérico, ya que su idea es lograr que los clientes puedan visualizar e invocar métodos sobre cualquier objeto CORBA local o remoto, a través de un navegador web. Con CORBAWeb un cliente puede navegar a través de enlaces de objetos CORBA utilizando URL generados dinámicamente para cada objeto remoto.

CORBAWeb funciona mediante un navegador web, en el que un usuario puede acceder e invocar métodos de objetos remotos que residan en un servidor, para ello se generan automáticamente formularios HTML a partir del lenguaje de definición de interfaces (*Interface Definition Language*, IDL por sus siglas en inglés), las cuales permiten la invocación de métodos de cualquier objeto CORBA. CORBAWeb recibe las acciones del usuario y las traduce, accede al objeto remoto requerido para invocar el método del objeto, obtiene el resultado y finalmente devuelve un documento HTML que contiene los resultados de la invocación.

## 2.2. SOPView+

Es un proyecto [5] de la Universidad Nacional de Seúl, Corea del Sur, desarrollado en un entorno UNIX y utiliza la herramienta *Motif widget* para poder proporcionar una interfaz gráfica. Este proyecto busca crear un navegador y visualizador de objetos, principalmente para la consulta y gestión de bases de datos orientadas a objetos. Con esta herramienta se puede explorar dentro de la base de datos con el fin de localizar algún objeto deseado, recuperar la información y visualizarla de forma gráfica. Además, esta herramienta visualiza los objetos de una forma jerárquica y permite la navegación en grandes bases de datos cambiando el objeto base, el cual es un objeto que se vuelve el nodo principal por el que se inicia la navegación.

SOPView+ permite a los usuarios cambiar el objeto base a lo largo de la búsqueda de objetos en las bases de datos; para ello se pone un *ancla* sobre el objeto. Esto hace que sea posible para los usuarios explorar objetos en una gran base de datos de una forma más sencilla.

## 2.3. PESTO

Esta herramienta [6] se crea a partir del proyecto GARLIC [9], cuyo objetivo es construir un sistema de información capaz de integrar datos que residen en diferentes sistemas de bases de datos. Los datos pueden consultarse a través de un lenguaje similar a SQL, el cual se ha ampliado para incluir características orientadas a objetos. El proyecto GARLIC busca proporcionar una interfaz novedosa que ofrezca la consulta y navegación de objetos, llamada PESTO, que es el trabajo conjunto entre IBM y la Universidad de Wisconsin - Madison, en donde trabajan en el desarrollo de una interfaz para usuarios.

El explorador portable de objetos estructurados (*Portable Explorer of Structured Objects*, PESTO por sus siglas en inglés) ofrece una interfaz propia en la que se puede navegar sobre objetos, y al igual que SOPView+, está diseñado para explorar bases de datos de objetos y ofrece la posibilidad de hacer consultas complejas de objetos en bases de datos, a través de un lenguaje similar a SQL gracias al paradigma *query in place*. Esta herramienta proporciona a los usuarios una interfaz que permite la navegación interactiva y la consulta de los contenidos de las bases de datos de GARLIC. Al igual que SOPView+, PESTO permite a los usuarios moverse hacia delante y hacia atrás en la base de datos, para consultar y navegar por los objetos, los cuales se muestran en una jerarquía de referencia que puede representarse por la conexión de nodos de objetos a través de enlaces.

## 2.4. CORBA object browser

Esta herramienta [7] tiene como finalidad acceder a objetos CORBA directamente desde un navegador web utilizando un esquema URI; permite la navegación e invocación sobre objetos CORBA de la misma forma en que los usuarios navegan en Internet. En esta herramienta es posible visualizar y ejecutar los métodos de un objeto determinado desde un navegador web; sin embargo, para realizar lo anterior, se debe

acceder mediante un prototipo de navegador llamado HotJava Web Browser, el cual ya no está disponible actualmente.

La ventaja del navegador HotJava Web Browser es que contaba con un mecanismo para acceder a objetos CORBA que se ejecutan en un ORB (*Object Request Broker*, ORB por sus siglas en inglés) seguro, a través del CORBA Object Browser. El acceso a objetos seguros a través del navegador requería la autenticación con el ORB remoto y también tener una comunicación segura.

## 2.5. Apache axis2

Es un motor de servicios web [8] desarrollado por *Apache Software Foundation*, el cual es interoperable, implementado en C++ y en Java, en el que se pueden crear aplicaciones interoperables y distribuidas. Al igual que WOX, Apache Axis2 es un *framework* de código abierto, basado en XML, que utiliza SOAP para el intercambio de mensajes. A pesar de que Apache Axis2 trabaja con objetos, no guarda el estado de los objetos, por lo que sus métodos solo se invocan como si fueran métodos estáticos. Una característica particular de esta herramienta es que a pesar de que ofrece la ejecución de métodos sobre objetos, no cuenta con una interfaz para sus usuarios, ya que es necesario llamar a los objetos por medio de su URL.

## 2.6. Tabla comparativa

En la Tabla 1 se presenta una comparación de las herramientas analizadas en el estado del arte: H1) CORBAWeb, H2) SOPView+, H3) PESTO, H4) CORBA Object Browser y H5) Apache Axis2. Se muestra un símbolo de verificación si la herramienta tiene la característica, o una *x* si no la tiene. A continuación se proporciona una breve descripción de las características que se tomaron en cuenta.

**Tabla 1.** Comparación de características de las herramientas analizadas.

<i>Características</i>	<i>H1</i>	<i>H2</i>	<i>H3</i>	<i>H4</i>	<i>H5</i>
Código libre	✓	✗	✗	✓	✓
Interoperabilidad	✓	✗	✗	✓	✓
Basado en objetos	✓	✓	✓	✓	✗
Servicio web	✓	✗	✗	✗	✓
Uso de XML	✗	✗	✗	✗	✓
Visualización de objetos	✓	✓	✓	✓	✗
Visualización de atributos	✗	✓	✓	✓	✗
Ejecución de métodos	✓	✗	✗	✓	✓
Interfaz web	✓	✗	✗	✓	✗
Manejo de bases de objetos	✗	✓	✓	✗	✗

**Código libre.** Se refiere al software distribuido y desarrollado libremente, es gratuito y cualquier usuario puede tener acceso al código fuente.

**Interoperabilidad.** Esta característica se refiere a que la herramienta es capaz de comunicarse en diferentes plataformas o lenguajes de programación.

**Basado en objetos.** Se refiere a que la herramienta utiliza objetos remotos.

**Servicio web.** Se refiere a que la herramienta está enfocada a servicios web.

**Uso de XML.** Esto significa que la herramienta utiliza XML como medio de comunicación entre el cliente y el servidor.

**Visualización de objetos.** Esta característica se refiere a que la herramienta analizada permite una forma de visualización de los objetos de forma gráfica.

**Visualización de atributos de los objetos.** Esta característica se refiere a que la herramienta analizada puede mostrar los atributos que contiene cada objeto.

**Ejecución métodos.** Se refiere a que se permite la ejecución de métodos pertenecientes a un objeto a través de un navegador web.

**Interfaz web.** Esta característica se refiere a que la herramienta analizada proporciona una interfaz web en la que el usuario puede visualizar los métodos de los objetos remotos.

**Manejo de bases de objetos.** Quiere decir que la herramienta es capaz de navegar por bases de objetos o repositorios.

### 3. Objetos web en XML

Esta sección proporciona una breve introducción del *framework* WOX, el cual combina características de los sistemas distribuidos basados en objetos y los sistemas distribuidos basados en web. Algunas de las características de este *framework* se presentan en los siguientes párrafos.

WOX utiliza URLs para identificar objetos remotos de manera única, siguiendo los principios de la arquitectura de transferencia de estados de representación (*Representational State Transfer*, REST por sus siglas en inglés) [10]. Esta es una característica importante porque todos los objetos son identificados de manera única por su URL y pueden ser accedidos desde cualquier lugar en la web, ya sea a través de un navegador web o mediante programación.

WOX emplea un serializador eficiente y fácil de usar, llamado serializador WOX [2], el cual es la base del *framework* para serializar objetos, solicitudes y respuestas intercambiadas entre clientes y servidores. Este serializador es una biblioteca independiente basada en XML, la cual es capaz de serializar objetos Java y C# a XML y viceversa. Una de sus principales características es la generación de XML estándar para objetos, el cual es independiente del lenguaje de programación y permite alcanzar interoperabilidad entre diferentes lenguajes de programación orientados a objetos. Aplicaciones escritas en los lenguajes Java y C# pueden interoperar.

WOX tiene un conjunto de operaciones estándar y especiales que se aplican sobre objetos locales y remotos. Estas operaciones incluyen la solicitud de referencias remotas, invocación de métodos estáticos (llamadas de servicios web), invocación de métodos de instancia, destrucción de objetos, solicitud de copias, duplicación de objetos, actualización de objetos, subida de objetos, invocación de métodos asíncronos,

entre otras. Algunas de las operaciones son descritas en [1]. El mecanismo utilizado por WOX en una invocación a un método se ilustra en la Figura 1.

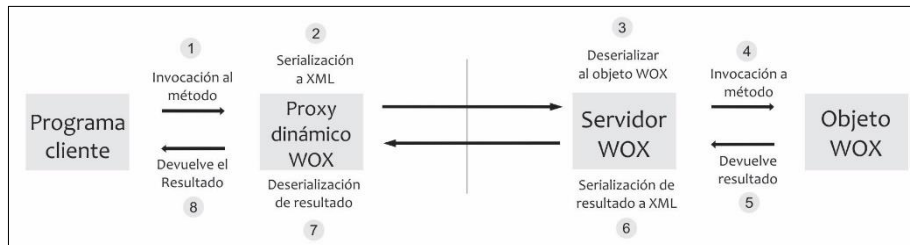


Fig. 1. Mecanismo de invocación de método remoto en WOX.

La serie de pasos llevados a cabo en la invocación de un método en WOX es la siguiente: 1) El programa cliente WOX invoca un método sobre una referencia remota (la forma en la que el cliente invoca un método sobre una referencia remota es exactamente la misma forma que la que utiliza para un objeto local); 2) El proxy dinámico WOX toma la solicitud, la serializa a XML y la manda a través de la red al servidor WOX; 3) El servidor WOX toma la solicitud y la deserializa a un objeto WOX; 4) El servidor WOX carga el objeto y ejecuta el método sobre él; 5) El resultado de la invocación del método es regresada al servidor WOX; 6) El servidor WOX serializa el resultado a XML y es regresado al cliente, ya sea el resultado real o una referencia a éste (el resultado es guardado en el servidor en caso de que una referencia haya sido enviada); 7) El proxy dinámico WOX recibe el resultado y lo deserializa al objeto apropiado (objeto real o referencia remota); 8) El proxy dinámico WOX regresa el resultado al programa cliente WOX.

Desde el punto de vista del programa cliente WOX, éste solamente realiza la invocación del método y obtiene el resultado de regreso de forma transparente. Las bibliotecas cliente WOX llevan a cabo el proceso de serializar la solicitud y mandarla al servidor WOX, así como recibir el resultado de la invocación del método y deserializarla. Las siguientes secciones presentan la herramienta web que permite almacenar y visualizar objetos WOX a través de un navegador web.

## 4. Herramienta web propuesta

En esta sección se describe la funcionalidad de la herramienta web desarrollada, su estructura e interfaz.

### 4.1. Funcionalidad

La funcionalidad de la herramienta web propuesta se describe a continuación, a través de una serie de acciones que se pueden llevar a cabo en ella.

**Acceso a un objeto WOX específico a través de la red.** La herramienta web permite a sus usuarios acceder a un objeto WOX almacenado en el sistema a partir de

un URL único generado por el mismo sistema. Con dicho URL los usuarios pueden acceder directamente al objeto sin necesidad de entrar al repositorio y buscar su objeto deseado entre todos los existentes en el mismo repositorio.

**Visualización de objetos de forma gráfica.** Los usuarios pueden ver, a través de una interfaz, los atributos de un objeto WOX determinado. Se proporcionan dos vistas: la primera es una tabla que representa al objeto y dentro de ella se visualizan todos sus atributos sin importar su tipo de dato; la segunda es una vista del código XML que representa al objeto, el cual es visualizado con un formato que permite al usuario entender cada etiqueta del código XML.

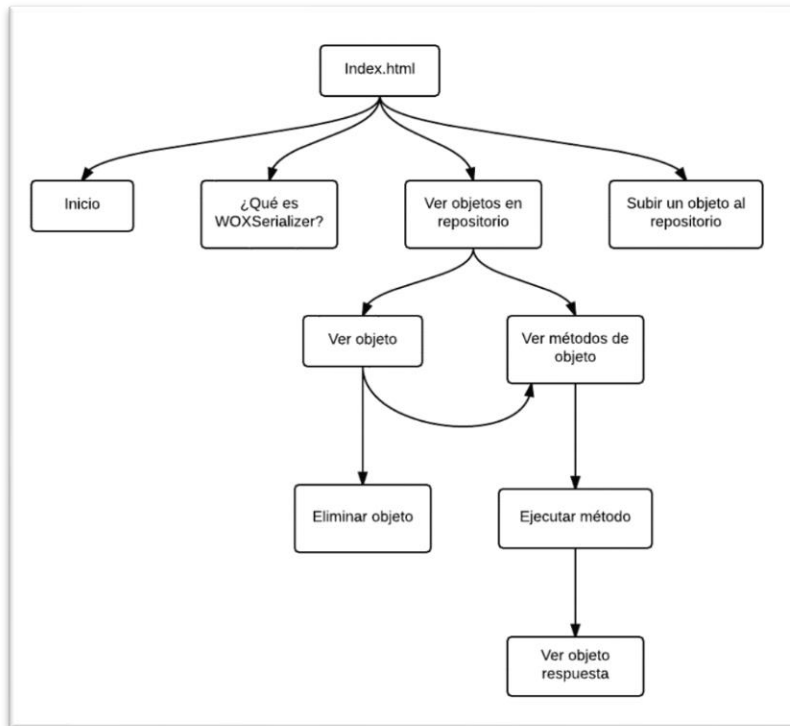


Fig. 2. Mapa de navegación de la herramienta web.

**Visualización de métodos de cualquier objeto WOX.** Los usuarios pueden visualizar todos los métodos que pertenecen a un objeto en particular, sin importar los parámetros que cada método requiera para ser invocado. Cada método cuenta con un espacio en el que se muestra la respuesta que devuelve el mismo al ser invocado.

**Ejecución de cualquier método perteneciente a la clase de un objeto WOX.** Mediante la herramienta web es posible invocar cualquier método de un objeto WOX; también se cuenta con una verificación de parámetros para evitar que el usuario introduzca valores de parámetros erróneos y así poder asegurar una correcta invocación de un método. Además, el sistema puede regresar una respuesta de la invocación de un método sin importar su tipo de dato. Es importante mencionar que para que la herra-

mienta web pueda ejecutar un método sobre un objeto, debe contar con la clase a la que pertenezca tal objeto.

**Almacenamiento de objetos WOX en el servidor.** Se proporciona a los usuarios la posibilidad de que ellos mismos puedan subir objetos WOX al servidor para su posterior manipulación, ya sea como parámetro para la invocación de un objeto o simplemente para almacenarlo en el repositorio para su posterior utilización.

## 4.2. Estructura

La herramienta web consiste de una interfaz web dinámica que permite a los usuarios el acceso a objetos WOX alojados en un repositorio. El usuario puede visualizar cada objeto existente en el repositorio de una forma gráfica, además de que se le permite ejecutar los métodos de cada objeto a través de un navegador web.

La Figura 2 muestra el mapa de navegación, en el que se representa la estructura de visualización que se emplea en la herramienta web.

## 4.3. Interfaz

La página de inicio de la herramienta web se muestra en la Figura 3, donde puede observarse el menú con las siguientes cuatro opciones: 1) Inicio; 2) ¿Qué es WOX serializer?; 3) Ver objetos en el servidor; y 4) Subir objeto al servidor.

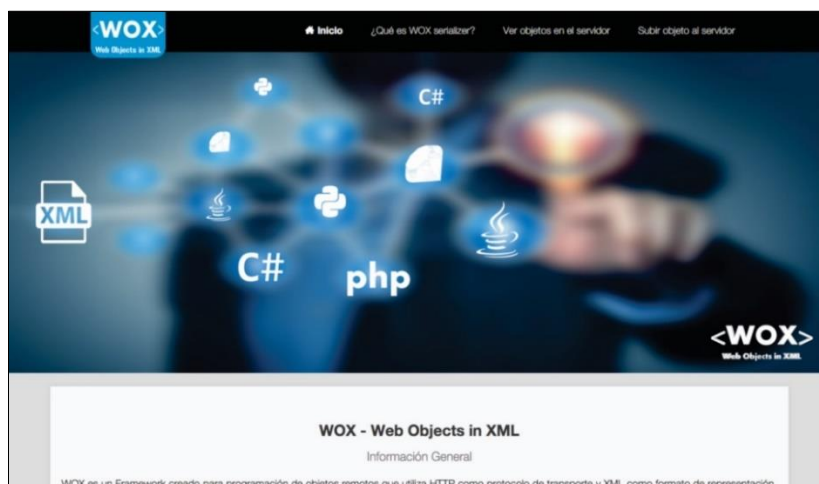


Fig. 3. Página de inicio de la herramienta web.

## 5. Herramienta web en funcionamiento

En esta sección se presenta la herramienta web en funcionamiento. Por restricciones de espacio, solamente se describe cómo se almacenan objetos en la herramienta web y cómo se visualizan a través del repositorio.



### 5.1. Almacenamiento de objetos

La Figura 4 muestra el funcionamiento de la herramienta web para almacenar objetos WOX, mediante una serie de pasos que se llevan a cabo entre cliente y servidor. El paso 1 es la serialización de un objeto con el serializador WOX, para lo cual puede utilizarse cualquiera de los lenguajes de programación soportados por WOX [11]; como segundo paso el usuario procede a subir el objeto al servidor al seleccionar la opción *Subir un objeto al servidor* del menú principal de la herramienta web; en el paso 3 el servidor se encarga del almacenamiento del objeto en el directorio llamado *UploadTemp* del repositorio de la herramienta web; en el paso 4 se procede a verificar el objeto; se deserializa para verificar que no contenga errores o que ya exista y dependiendo del resultado, se mueve el objeto a la papelera si éste tenía errores (paso 5.1), o se registra en el repositorio en caso de que sea un objeto sin errores (paso 5.2); finalmente, en el paso 6 se le comunica al usuario el resultado del almacenamiento del objeto WOX en el repositorio.

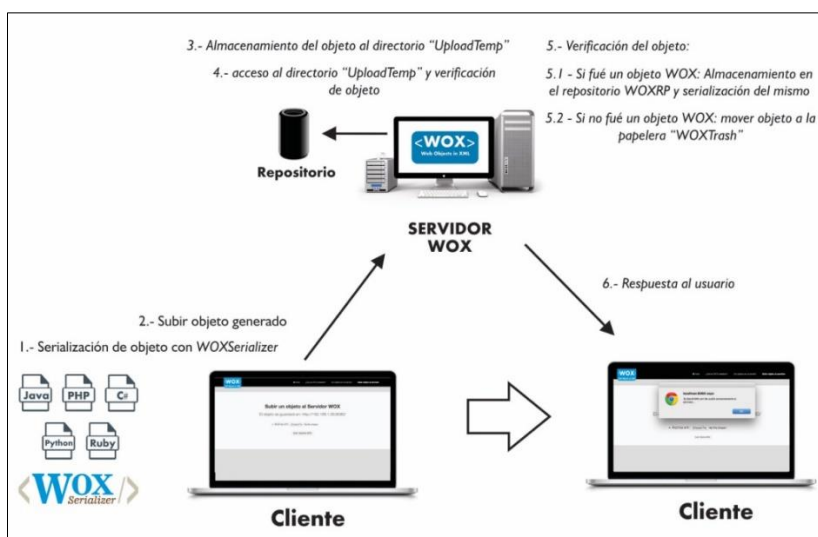


Fig. 4. Almacenamiento de un objeto WOX.

### 5.2. Visualización de objetos

La Figura 5 muestra el funcionamiento del repositorio para visualización de objetos WOX, mediante una serie de pasos que se llevan a cabo entre cliente y servidor. En el paso 1 el usuario accede al repositorio al seleccionar la opción *Ver objetos en el servidor* del menú principal de la herramienta web; en el segundo paso se deserializa el objeto *WOXRP.xml*, el cual es una lista de objetos de la clase *WOXObject* que contiene la información de todos los objetos almacenados en el repositorio; en el paso 3, una vez deserializado el objeto *WOXRP.xml*, se puede acceder a la información de los objetos registrados, por lo que se genera una tabla que contendrá el URL y la clase de

cada objeto registrado, así como dos opciones, una para visualizar el objeto y otra para acceder a sus métodos (paso 4); finalmente, en el paso 5 se muestra al usuario la tabla generada, para que éste elija el objeto deseado para su visualización.

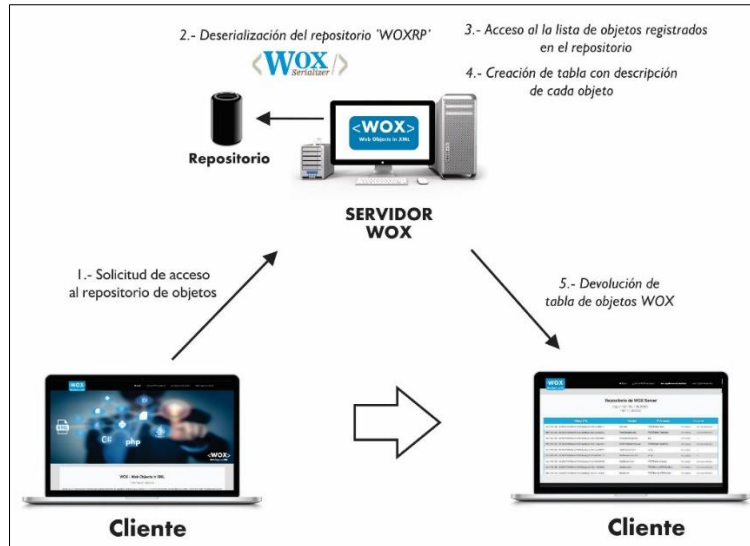


Fig. 5. Acceso al repositorio de objetos para su visualización.

WOX Web Objects in XML

Inicio ¿Qué es WOX serializer? Ver objetos en el servidor Subir objeto al servidor

Repositorio de WOX Server

http://192.168.1.90:8080/

Hay 11 objetos:

Object URL	Nombre	Referencia	Acciones	
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=312953317	libro.xml	WOXTester.Libro	Ver objeto	Ver sus métodos
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=334958425	TestComputer.xml	WOXTester.Computer	Ver objeto	Ver sus métodos
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=395534305	TestListCourses.xml	list	Ver objeto	---
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=735476557	TestPrimitiveArrays.xml	WOXTester.TestArray	Ver objeto	Ver sus métodos
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=554794375	TestProducts.xml	array	Ver objeto	---
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=662240113	TestReferences.xml	array	Ver objeto	---
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=695452496	TestStudent.xml	WOXTester.Student	Ver objeto	Ver sus métodos
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=140430062	visualizer.xml	WOXServer.WOXVisualizer	Ver objeto	Ver sus métodos
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=617257542	invoker.xml	WOXServer.WOXInvoker	Ver objeto	Ver sus métodos
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=242283985	WOXSer.xml	WOXServer.WOXServer	Ver objeto	Ver sus métodos
http://192.168.1.90:8080/WOXServer/WOXObject.jsp?objectId=057452956	B_GandhiMX.xml	WOXTester.Biblioteca	Ver objeto	Ver sus métodos

Fig. 6. Repositorio de objetos de la herramienta web.

En la interfaz de la herramienta web para acceder al repositorio de objetos el usuario debe seleccionar la opción *Ver objetos en el servidor*, la cual lo llevará a la página que se muestra en la captura de pantalla de la Figura 6. El repositorio de objetos tiene cuatro columnas: el URL del objeto en el servidor, el nombre del objeto, la referencia remota al objeto y las acciones que se pueden ejecutar sobre el objeto (*Ver objeto* y

Ver sus métodos). A través del hipervínculo *Ver objeto* se visualiza la representación gráfica del objeto y su representación en XML, como se ilustra en la Figura 7.



Fig. 7. Visualización de un objeto gráficamente y en XML.

## 6. Conclusiones y trabajo futuro

En este artículo se presentó una herramienta web que permite almacenar objetos, así como visualizar su estado y métodos. Esta herramienta web es un complemento de *Web Objects in XML (WOX)*, el cual es un *framework* para programación de aplicaciones distribuidas basadas en objetos.

La herramienta web cumplió con los objetivos planteados al inicio de su desarrollo, ya que se logró exitosamente diseñar e implementar el almacenamiento de objetos-WOX y proveer un repositorio de objetos a partir del cual es posible visualizar cada objeto que ha sido creado en éste, de manera gráfica y en su representación XML; todo ello a través de un navegador web. Asimismo, se implementó exitosamente la funcionalidad para la visualización de los métodos de cualquier objeto WOX que se encuentra en el repositorio, así como la ejecución de métodos de un objeto WOX a través de una interfaz donde se proporcionan los valores para cada uno de sus parámetros.

Como trabajo futuro es necesario afinar la implementación de la ejecución de métodos de un objeto WOX, ya que en esta versión de la herramienta web solamente es posible ejecutar métodos cuyos parámetros son de tipo primitivo. La ejecución de métodos que reciben objetos como parámetros no está finalizada.

## Referencias

1. Jaimez-González, C., Lucas, S.: Implementing a State-based Application Using Web Objects in XML. In: Proceedings of the 9th International Symposium on Distributed Objects, Middleware, and Applications (DOA 2007), Lecture Notes in Computer Science, 4803, 577–594 (2007)

2. Jaimez-González, C., Lucas, S., López-Ornelas, E.: Easy XML Serialization of C# and Java Objects. In: Proceedings of the Balisage: The Markup Conference 2011, Balisage Series on Markup Technologies, 7 (2011)
3. Jaimez-González, C., Lucas, S., Luna-Ramírez, W.: A Web Tool for Monitoring HTTP Asynchronous Method Invocations. In: Proceedings of the 7th IEEE International Conference for Internet Technology and Secured Transactions, (ICITST-2012), 127–132 (2012)
4. Merle, P., Gransart, C., Geib, J.: CorbaWeb: A Generic Object Navigator. Disponible en: [http://www.lifl.fr/~merle/papers/96\\_WWW5/paper/Overview.html](http://www.lifl.fr/~merle/papers/96_WWW5/paper/Overview.html)
5. Chang, S., Kim, H.: SOPView+: An Object Browser Which Supports Navigating Database by Changing Base Object. In: Proceedings of the 21st International Conference on Computer Software and Applications Conference, COMPSAC 97 (1997)
6. Carey, M., Haas, L., Maganty, V., Williams, J.: PESTO: An Integrated Query/Browser for Object Databases. In: Proceedings of the 22th International Conference on Very Large Data Bases, Mumbai, India (1996)
7. Kumar, G., Jalote, P.: A Browser Front End for CORBA Objects. In: 10th International World Wide Web Conference (2001)
8. Apache Software Foundation. Web Services - Apache Axis, Disponible en: <http://ws.apache.org/axis/>
9. Tork, M., Arya, M., Haas, L., Carey, M., Cody, W., Fagin, R., Schwarz, P., Thomas, J., Wimmers, E.: The Garlic Project. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (1996)
10. Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, Disponible en: [http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm\(2000\)](http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm(2000))
11. Web Objects in XML. Disponible en: <http://woxserializer.sourceforge.net/>