

## **EJECUCIÓN RECURSIVA DE SERVICIOS WEB**

<sup>1</sup>Ismael Armando Zúñiga Félix, <sup>2</sup>Zindi Sánchez Hernández y <sup>3</sup>David Jesús

Hernández Rodríguez

<sup>1,2</sup>División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Nogales. Ave. Instituto Tecnológico # 911 Cp. 84065 H. Nogales Sonora, México

<sup>3</sup>Departamento de Ciencias Básicas, Instituto Tecnológico de Acapulco.  
Ave. Instituto Tecnológico de Acapulco s/n, Crucero de Cayaco, C.P. 39905,  
Acapulco, Guerrero, México

<sup>1</sup>iz@depiitn.edu.mx, <sup>2</sup>zindi.sanchez@depiitn.edu.mx y <sup>3</sup>djhr99@hotmail.com

### **RESUMEN**

Se presenta un prototipo por medio del cual se permite ejemplificar el llamado recursivo de servicios Web, como una pequeña aplicación Web que toma la forma de una calculadora constituida en base a servicios, donde cada operación aritmética es un servicio Web, los cuales son independientes entre sí y pueden ser accedidos de forma independiente e incluso recursivamente por otros y por sí mismos. Componentes Web en base a servicios Web recursivos.

**PALABRAS CLAVE:** Servicios Web, Recursividad, SOA, SOAP y SaaS.

### **INTRODUCCIÓN**

En términos generales se puede decir que los sistemas involucran un conjunto de funcionalidades que de forma coordinada se interrelacionan prestando sus servicios para cumplir objetivos sistémicos preestablecidos. De tal forma que “servicios” básicos (simples) se pueden conformar (componer), en conjunto, como un sistema que brinda un servicio más complejo; el cual a su vez puede prestar sus servicios (visto como un subsistema) a uno o más sistemas.

Lo antes expuesto, es la extrapolación de lo que sucede en la vida real, donde sistemas se integran a partir de sistemas. Que en principio pueden ser independientes y autónomos, y en coordinación cooperan para la consecución de objetivos y metas de interés común (previo acuerdo).

Los servicios pueden ser solicitados en número y frecuencia en base a necesidades, oferta, disposición, recursos y contratos (acuerdos preestablecidos). Donde a un servicio, incluso, le pueden ser solicitadas sus “funcionalidades” (servicios) por sí mismo; a lo cual se le conoce como un “llamado recursivo”.

Los servicios Web se implementan por medio de un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Pueden considerarse como Componentes de Software que intercambian datos en red; que mediante la adopción de estándares abiertos logran interoperabilidad. [1]

En términos generales pueden implementar un sistema (siendo el sistema en sí mismo) o ser el front-end (implementando la interfaz) del mismo, el cual puede estar basado en otra tecnología-plataforma.

El presente artículo presenta el resultado del análisis de las implicaciones al conformar servicios Web como sistemas que son integrados a su vez por otros sistemas y que pueden usar sus servicios de forma recursiva (como parte de los resultados del proyecto “Viabilidad e implicaciones en el desarrollo de Servicios Web en software libre”, con registro DGEST: NOG-MS-2009-101).

Otra razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad, con el tiempo, será cada vez más importante dado que la

tendencia es construir grandes aplicaciones a partir de componentes distribuidos más pequeños.

Por otro lado, antes de que existiera SOAP (Simple Object Access Protocol), no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red. Las que había eran ad hoc y poco conocidas, tales como EDI (Electronic Data Interchange), RPC (Remote Procedure Call), u otras APIs.

Un ejemplo del uso de servicios es SaaS (Software as a Service). Del cual en [2] se dice que en la medida que las organizaciones continúen adoptando modelos de outsourcing para la automatización de los procesos críticos de negocio, SaaS es cada vez más atractiva para muchos tipos diferentes de prestadores de servicios así como los Vendedores de Software Independientes (en inglés: ISV). A través de este enfoque los proveedores de servicio reducen los costos internos de operación y ayudan a reducir el coste total de apropiación por parte de los clientes. El tiempo de implementación es más corto y se logra mayor aceptación por los usuarios.

Aunado a la tendencia del acceso remoto en diferentes plataformas, al basar aplicaciones SaaS en browser se permite el “preferido” acceso móvil, ya que seguridad, privacidad y portabilidad son los objetivos primordiales [3].

SaaS aprovecha a SOA (Service Oriented Architecture) permitiendo que aplicaciones de software se comuniquen entre sí. Cada servicio de software puede actuar como un proveedor de servicios, exponiendo su funcionalidad a otras aplicaciones a través de “corredores” (brokers) públicos, y también puede actuar como un cliente de servicios, incorporando funcionalidad y datos de otros servicios. [4]

## **PROTOTIPO**

De acuerdo a la naturaleza del proyecto, buscamos el “empaparnos” de la tecnología y analizar sus implicaciones; para usarlas de base uno siguiente, el cual pretende hacer uso de los resultados para proponer un modelo de integración de servicios haciendo uso del concepto de los Servicios Web, más que de sus detalles de implementación. Donde se tratarán a los servicios Web sólo como servicios; y de forma análoga a la vida real, determinar las interrelaciones que pudieran establecerse para solucionar un problema complejo, donde un servicio complejo se basa a partir de dos o mas servicios básicos.

Para nuestros fines el analizar las implicaciones de la concurrencia sobre servicios Web nos delimita y nos da pauta para el desarrollo de sistemas conformados en base a servicios Web.

## **RESULTADOS**

Para los fines del proyecto se implementó un “sistema ejemplo” como un prototipo en base a servicios Web que implementa una calculadora con operaciones aritméticas básicas (suma, resta, multiplicación y división), accesible en línea (ver figura 3) en: <http://tec.site90.net/re/cliente.php> (archivos que implementan la arquitectura básica del prototipo en: <https://sites.google.com/site/iazfbc/proyectos/modifi.rar>).

Experimentando con el concepto base de servicios Web, se materializó sobre el lenguaje de script server PHP, para plantear estructuras de relación para la conformación de servicios en base a servicios básicos como se muestra en las figuras 1 y 2. La figura 1 muestra un ejemplo de la interdependencia física que puede llegar a presentarse entre servicios Web y clientes de los mismos y la figura 2 muestra un ejemplo de interdependencia lógica que se puede presentar al

conformar servicios interrelacionados unos con otros; donde la recursividad puede presentarse, de forma natural de acuerdo al sistema. Con lo anterior se busca ilustrar la complejidad al conformar sistemas en base a servicios Web análogamente a lo que sucede en cualquier ámbito, al hablar del concepto de “servicio” interactuando con otros servicios. Aunado a lo anterior, la recursividad es una práctica común en la conformación de servicios, la cual implica la administración y control de “instancias” (como “versiones”) de un servicio en el llamado recurrente del mismo.

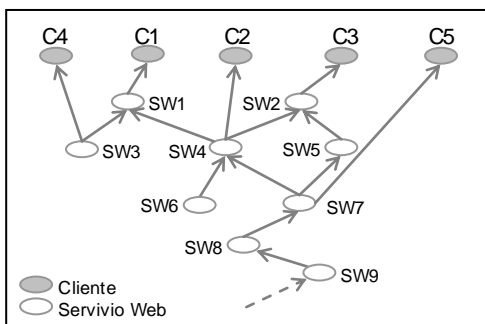


Figura 1. Ejemplo de interdependencia física en la conformación de servicios Web.

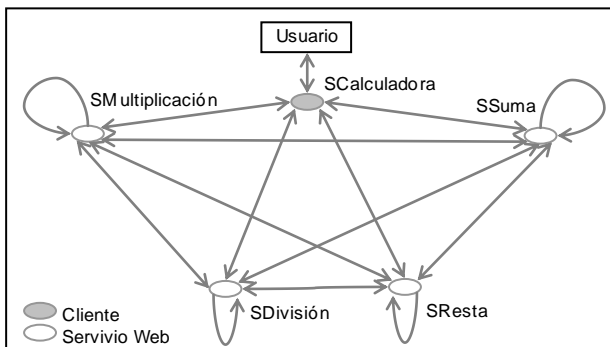


Figura 2. Ejemplo de interdependencia lógica entre servicios Web.

La figura 1 muestra las principales implicaciones del caso, destacamos la conformación de dependencias en cuanto a número y profundidad virtualmente infinita que pudiera establecerse. En la figura 2 se plantea un ejemplo de

interdependencia lógica entre Servicios Web, de tal forma que lo que se ilustra para el servicio de “SMultiplicación” pudiera ocurrir a los otros “Sx” (SW); enfatizando incluso donde la arista de SMultiplicación que apunta a SMultiplicación ilustra el llamado recursivo a sí mismo (a su(s) servicio(s)).

Un posible escenario como ejemplo ilustrativo podría ser el de “Usuario” (en la figura 2) el cual envía solicitudes de servicio a “SCalculadora” (servicio de calculadora de operaciones aritméticas básicas) de la forma (por ejemplo): “(5\*(4\*(2+1)))” (o de la forma “(6+((8\*3)/4))”), el cual las analiza y llama a los Servicios Web correspondientes (SSuma, SResta, SDivisión y SMultiplicación; ver figura 4, donde cada “Sx” se encuentra en un subdirectorío diferente y tiene un juego de tres archivos (servicio.php, nusoap.php y auxx.php) con el mismo nombre (pero diferente implementación) ubicado en <http://tec.site90.net/re/cliente.php>.

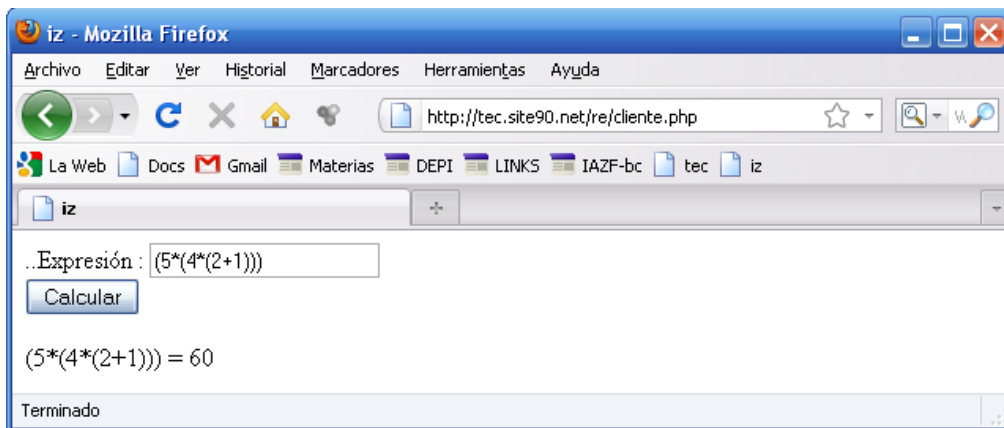


Figura 3. Ejecución en línea.

Un posible escenario de ejecución (ver figura 3).- “Usuario” con la expresión “(5\*(4\*(2+1)))” hace uso del servicio calculadora (SCalculadora), quién determina que “(5\*(x))” es una multiplicación y llama a SMultiplicación (instancia 1), quién a su vez determina que “(x)” es una multiplicación “(4\*(y))” y llama a SMultiplicación

(a sí misma, instancia 2), de la cual “(y)” es una suma “(2+1)” y llama a SSuma. De tal forma que, SSuma resuelve que “(2+1)=3”, retornando “3” a SMultiplicación (instancia 2) la cual resuelve “(4\*(3))=12”, y retorna a su vez “12” a SMultiplicación (instancia 1) el cual resuelve “(5\*(12))=60”, el cual retorna “60” a SCalculadora quien a su vez como coordinador del proceso y quien implementa el servicio de “calculadora” (concentra y orquesta los “Sx”) y retorna como resultado “60” a “Usuario” cliente del SCalculadora.

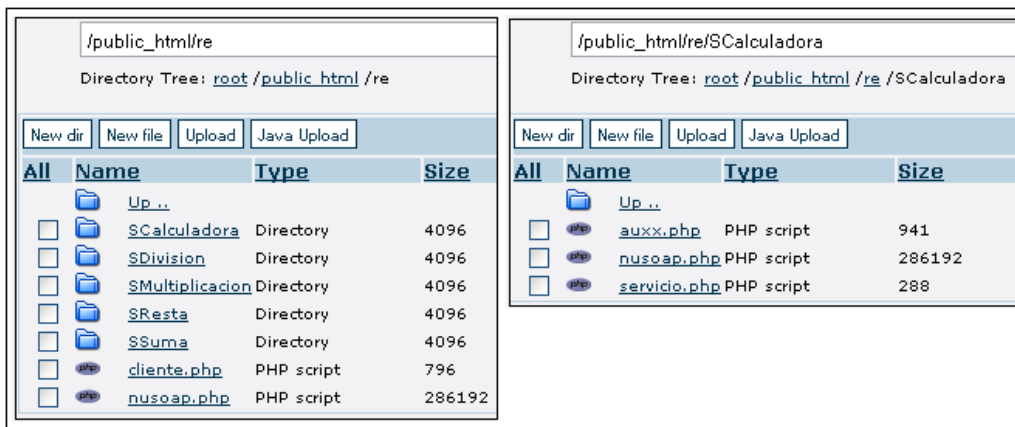


Figura 4. Estructura de archivos involucrados.

## CONCLUSIONES OBSERVACIONES

El desarrollo de Servicios Web sobre software libre es viable y robusto, las implicaciones que encontramos nos han permitido determinar que la plataforma PHP es conveniente para la investigación en curso “Implementación de servicios a partir de servicios básicos usando Servicios Web en software libre, para verificar las implicaciones de soportar el concepto de Componentes de Software” (en proceso de registro ante DGEST, “Registro de proyecto de investigación CR-01/2010” con fecha de elaboración 23 de Agosto de 2010) y para futuras investigaciones en el área de sistemas distribuidos, conformados como servicios.

Ya que nos abstrae de los detalles de implementación del modelo de servicios Web, lo cual nos permite concentrarnos en su uso, sin detenernos en los detalles técnicos, por otro lado, para los fines que buscamos, el hecho de que el mismo navegador Web interprete el código, también nos conviene por que elimina la complejidad y tiempo de desarrollo.

La consecución del presente proyecto ha permitido sentar las bases en el conocimiento requerido para realizar un proyecto continuación al presente, donde se buscará proponer un “modelo” a seguir para conformar “servicios integrados”, servicios bajo arquitectura SOA (Service Oriented Architecture) a través de Servicios Web, como la unidad básica para conformar servicios mas complejos que a su vez son Servicios Web.

En la actualidad se ha iniciado con las investigaciones y prototipos preliminar para plantear la problemática y consecuencias vislumbradas en el desarrollo de sistemas basados en servicios interdependientes, que llamamos “conformación de servicios Integrados” en base a un middleware implementado con servicios Web bajo una arquitectura P2P (Peer-to-Peer).

## **REFERENCIAS**

[1]. [http://es.wikipedia.org/wiki/Servicio\\_web](http://es.wikipedia.org/wiki/Servicio_web) (27 Octubre 2010).

[2]. [saas.com](http://www.saas.com),

<http://www.saas.com/ta/inside.jsp?page=SolutionsOverview&type=solutions&type2=overview> (27 Octubre 2010).

[3]. Biddick Michael, Time for a SaaS strategy, InformationWeek, Ed. 1, 254 (18 Enero 2010), p. 28.

[4]. [http://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Software_as_a_service) (27 Octubre 2010).



## **AUTORES**

**Ismael Armando Zúñiga Félix**, Ingeniero en Sistemas Computacionales (Inst. Tec. de Huatabampo ITHua), Maestro en Ciencias en Ciencias Computacionales (cenidet), docente en licenciatura y posgrado y Jefe del departamento académico de Sistemas y Computación del Inst. Tec. de Agua Prieta ITAP (1999-2006), docente en licenciatura y posgrado (División de Estudios de Posgrado e Investigación DEPI) Inst. Tec. de Nogales ITN (2006-a la fecha).

**Zindi Sánchez Hernández**, Ingeniero en Sistemas Computacionales (Instituto Tecnológico del Istmo), Maestro en Ciencias en Ciencias de la Computación (Instituto Tecnológico de Nogales). Docente de Ciencias Básicas ITN (2005-2006), Jefe de la División de Estudios de Posgrado e Investigación ITN (2008). Actualmente docente en la carrera de Ingeniería en Sistemas Computacionales y del programa de posgrado M. Sist. Computacionales del ITN.

**David Jesús Hernández Rodríguez**, Ingeniero en sistemas Computacionales (Inst. Tec. de Acapulco). Docente en licenciatura del Instituto Tecnológico de Acapulco (2001- a la fecha).