

## Arquitectura Orientada a Servicios para el control de flujos de inventarios de un Ingenio Azucarero

Julio Cesar Bertani Carrera, Giner Alor Hernandez

División de Estudios de Posgrado e Investigación  
Instituto Tecnológico de Orizaba  
Email: jbertanic@yahoo.com.mx, galor@itorizaba.edu.mx  
Paper received on 31/07/08, accepted on 10/09/08.

**Resumen.** La Arquitectura Orientada a Servicios (SOA) se ha convertido en un nuevo paradigma para el desarrollo de aplicaciones distribuidas. SOA permite integrar sistemas heredados con nuevas aplicaciones de desarrollo y crear aplicaciones más flexibles y adaptables. Además, permite disminuir los tiempos de implementación y de mantenimiento. SOA utiliza la tecnología de servicios Web como solución tecnológica para la integración de aplicaciones empresariales. En este trabajo se presenta una arquitectura orientada a servicios que resuelve problemas de integración con sistemas heredados de control de inventarios de un Ingenio Azucarero. Así también, se describen los niveles de prestación de servicios de la arquitectura propuesta. Además, se presenta un caso de estudio donde se describe la funcionalidad y validez de la SOA propuesta. Finalmente, se discute el estado del arte y se enfatiza nuestra contribución.

### 1. Introducción

La mayoría de las empresas cuentan con software desarrollado anteriormente denominados software heredado. Para el desarrollo de este software las empresas invirtieron mucho dinero y tiempo por lo cual el pensar sustituirlos no es una decisión fácil de tomar. Bajo esta premisa y la importancia que estos sistemas tienen en sus respectivos negocios, las empresas buscan la manera de integrar sus nuevos desarrollos de software con mejores prácticas de calidad, mejores procesos administrativos y la utilización de tecnología de punta con los sistemas heredados. El llevar a cabo esta integración presenta problemas como la interoperabilidad, la dependencia de la arquitectura de trabajo, procesos de negocios, así como de los lenguajes de programación. Los servicios web surgieron para minimizar estos problemas y por la necesidad de estandarizar la comunicación entre distintas plataformas y lenguajes de programación. Los servicios web proporcionan un conjunto de protocolos que permiten a las aplicaciones exponer su funcionalidad y sus datos a otras aplicaciones a través de Internet [1]. Como una solución a la integración de sistemas heredados, surge en años recientes la arquitectura orientada a servicios (SOA, *Service-Oriented Architecture*). SOA define una nueva forma de entender el desarrollo de aplicaciones, orientándolas al entorno Web y permitiendo ensamblar componentes de softwa-



re para que sean reutilizables. SOA es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario [2]. Algunas beneficios que se obtienen al utilizar SOA son la reducción en el tiempo de desarrollo de aplicaciones, mejorar el control de los procesos de negocios, reducir los costos de desarrollo de nuevos proyectos, entre otros. En este trabajo se presenta el desarrollo de una arquitectura orientada a servicios que permitió al Ingenio San Nicolás integrar su sistema heredado de control de flujos de inventarios con una nueva aplicación bajo tecnología de servicios Web.

En la siguiente sección se discuten las características básicas de una arquitectura orientada a servicios.

## 2. Conceptos Básicos de SOA

Comúnmente, una empresa se constituye de un conjunto de aplicaciones y diversas fuentes de información que realizan una serie de funciones y almacenan innumerables datos. Por lo general, estas aplicaciones y recursos de información residen en diferentes organizaciones las cuales se construyeron utilizando diferentes tecnologías de información y ejecutándose en diferentes plataformas de software. En este contexto, una SOA provee un mecanismo de integración de aplicaciones existentes sin tener en cuenta la plataforma o el lenguaje de programación en que se desarrollaron, por lo que las soluciones basadas en una SOA se componen de servicios re-utilizables, bien definidos, públicos e interfaces basadas en estándares de Internet. Hay tres niveles de abstracción conceptual dentro de una SOA:

**1. Operaciones:** Son transacciones que representan unidades lógicas de trabajo. La ejecución de una operación influye en la persistencia de la información ya que puede representar funciones de lectura, escritura y modificación. Las operaciones de una SOA se comparan directamente a los métodos de la programación orientada a objetos (OOP, *Object-Oriented Programming*). De igual forma que los métodos, las operaciones tienen una interfaz específica, estructurada y devuelven respuestas estructuradas. Así también, la ejecución de una operación específica puede involucrar la invocación de operaciones adicionales

**2. Servicios:** Representan agrupaciones lógicas de operaciones. Por ejemplo, suponga que hay un servicio de Registro de Clientes, entonces, capturar, verificar, validar y guardar la información representan las operaciones asociadas que constituyen al servicio

**3. Procesos de negocio:** Es un conjunto de acciones o actividades realizadas con metas comerciales específicas. En un escenario comercial típico, los procesos de negocio realizan múltiples invocaciones de servicios. Algunos ejemplos de procesos de negocio son: Venta de Productos o Servicios. Órdenes de Pago, entre otros. En términos de una SOA, un proceso comercial consiste de una serie de operaciones que se ejecutan en una cierta secuencia según un conjunto de reglas comerciales. El orden, la selección y la ejecución de las operaciones se denomina orquestación del proceso comercial. Comúnmente, los servicios orquestados se invocan para responder a eventos comerciales.

SOA permite abordar los nuevos retos empresariales, ser más competitivos y disponer de sistemas de información integrados. La flexibilidad que brinda este tipo de arquitectura a responder rápidamente y de manera efectiva a los cambios de las condiciones de mercado es el gran diferenciador de SOA y promete la agilidad tan esperada para el negocio. SOA simplifica las interconexiones y el reuso de los sistemas heredados.

La metodología OOAD (*Object-Oriented Analysis and Design*) propuesta por Booch y Jacobson, provee un excelente punto de inicio en la definición de una SOA. El principal problema con las técnicas de diseño orientado a objetos con respecto a SOA es que su nivel de granularidad se enfoca en el nivel de clase, el cual se sitúa demasiado abajo para un nivel de abstracción para la modelación de un servicio comercial. Las asociaciones como herencia crean un fuerte acoplamiento y por consiguiente, una dependencia entre las partes involucradas. Por el contrario, el paradigma SOA promueve flexibilidad y agilidad a través de un bajo acoplamiento. En la figura 1 se ilustran los niveles de visibilidad y enfoque dados por el diseño orientado a objetos, orientado a componentes y orientado a servicios.

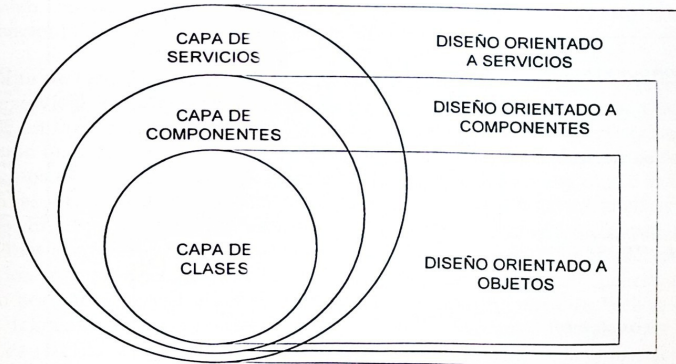


Fig. 1. Niveles de capas de diseño.

Una de las diferencias fundamentales entre SOA y el diseño Orientado a Objetos es la manera en la que ambas definen una aplicación. El diseño orientado a objetos determina que una aplicación se compone de clases interdependientes, mientras que SOA considera que una aplicación se compone por un conjunto de servicios autónomos [4]. SOA se implementa mediante servicios Web, y las aplicaciones se construyen como conjuntos de servicios re-utilizables y cooperantes, siendo cada uno de ellos responsable de una o más tareas de usuario, procesos de negocio o servicios de información perfectamente identificados y delimitados. SOA es un paradigma arquitectónico para crear y administrar "servicios comerciales" que tiene acceso a fun-

ciones, recursos, y piezas de información con una interfaz común sin tener en cuenta la localización o la composición técnica de la función o fragmento de datos [5]. En la siguiente sección se presenta una arquitectura orientada a servicios basada en capas que proporciona mecanismos de integración para el control de flujos de inventarios.

## 2.1. Una SOA basada en capas para el control de flujo de inventarios

Una vista abstracta de nuestra contribución se presenta en la figura 2. En esta figura, se presenta una arquitectura orientada a servicios basada en capas con diferentes niveles de prestación de servicios. Cada capa tiene una funcionalidad específica que a continuación se describe a detalle:

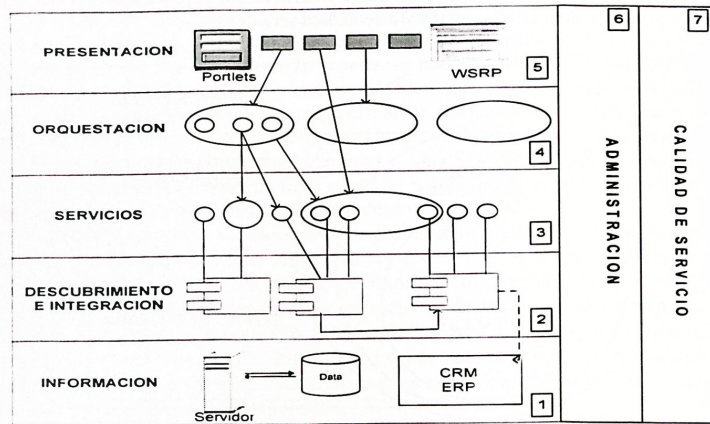


Fig. 2. Las diferentes capas de nuestra arquitectura orientada a servicios.

**Capa de Información.** Esta capa provee la información o datos provenientes de sistemas heredados. Esta capa consiste en aplicaciones existentes personalizadas, incluyendo aplicaciones empaquetadas como CRM y ERP, y sistemas orientados a objetos, así como las aplicaciones de inteligencia de negocio. Son sistemas que soportan la operación del negocio y que no están bajo el nuevo esquema de orientación de servicios.

**Capa de Descubrimiento e Integración.** En esta capa se realiza el descubrimiento e integración de servicios Web. Para realizar el proceso de descubrimiento

de servicios Web, se utilizan mecanismos que transforman los mensajes que se intercambian en un proceso de negocio, al formato correspondiente de la aplicación receptora. En este sentido, se tiene un diccionario común que contiene información de cómo cada componente o aplicación se comunica con otra, así como el significado de la información. Los mecanismos de transformación de mensajes incluyen análisis gramatical y métodos de emparejamiento que describen la estructura de cualquier formato de mensaje. Los formatos de mensajes se construyen por piezas de información que representan una solicitud de una funcionalidad encapsulada dentro del mensaje.

**Capa de Servicios.** En esta capa se encuentran los servicios Web creados para ser utilizados por las aplicaciones y su principal característica es que acceden directamente a los recursos, o sistemas legados, encapsulan las funcionalidades específicas de los sistemas existentes, dándoles así una interfaz que permita integrarlos. Esta interfaz utiliza WSDL el cual es un lenguaje basado en XML para describir la funcionalidad de un servicio Web. Estos servicios Web realizan operaciones atómicas de la empresa, principalmente funcionalidades de procesos de negocios simples, como por ejemplo verificar la existencia de un producto en el inventario. Los servicios Web simples de esta capa representan operaciones básicas de: (1) registro; (2) eliminación; (3) búsqueda; (4) meta-información.

**Capa de Orquestación.** En esta capa se definen las composiciones y coreografías de servicios expuestos en la capa de servicios. Los servicios se empaquetan dentro de un flujo a través de orquestación o coreografía, y de esta manera actúan juntos como una sola aplicación. Estas aplicaciones soportan específicos casos de uso y procesos de negocios. Los procesos BPM se implementan en esta capa e incorporan tareas interactivas (interacción participante), con actividades automatizadas (servicios). En esta capa también residen los servicios Web compuestos (formados por dos o más servicios simples), los cuales representan funcionalidades de procesos de negocios complejos, como por ejemplo la compra de un producto, el cual está compuesto por varios servicios simples. Para la orquestación de servicios Web se utilizan 2 lenguajes composicionales: *Business Process Execution Language for Web Services* (BPEL) el cual se enfoca tanto a la coreografía como a la orquestación y *Web Services Choreography Description Language* (WS-CDL) el cual se enfoca básicamente a la descripción y a la coreografía de servicios Web. Estos lenguajes permiten la representación y ejecución de flujos de trabajo que involucran actividades de procesos de negocio.

**Capa de presentación.** En esta capa reside el conjunto de interfaces que permiten acceder a la información a través de la invocación de los diferentes servicios Web provistos en las capas de servicio y de orquestación. En esta capa residen Servicios Web para Portlets Remotos (WSRP) y otras tecnologías como Java Server Faces, que desarrollan componentes de software que proveen una completa interfaz de usuario para la invocación de servicios Web. Es importante notar que en una SOA se debe de desacoplar la interfaz de usuario de los componentes.

**Capa de Administración.** Esta capa lleva a cabo la administración de servicios Web provistos en las diferentes capas de la SOA propuesta. *WS-Manageability* puede utilizarse en esta capa para la administración de servicios Web. Para realizar esto, es necesario usar la implementación JMX (*Java Management eXtension*). La arquitectura JMX consiste de tres niveles: instrumentación, agencia y servicios distribuidos. JMX proporciona interfaces y servicios adecuados para supervisar y administrar requerimientos de sistemas. Esta funcionalidad involucra abstracción de recursos usando componentes llamados MBeans (*Managed Beans*) y accesibilidad de recursos instrumentados remotamente a través de conectores JMX. Un MBean es un objeto de Java que representa un recurso administrado, como una aplicación, un servicio, un componente, o un dispositivo.

**Capa de Calidad de Servicio.** Esta capa proporciona las capacidades requeridas para supervisar, administrar y mantener métricas de calidad de servicio como seguridad, rendimiento y disponibilidad. Este es un proceso en segundo plano a través de mecanismos de solicitud/respuesta y herramientas que vigilan el buen funcionamiento de aplicaciones SOA, incluyendo todas las normas importantes de implementaciones de administración de Servicios Web y otros protocolos pertinentes y estándares que implementen la calidad en el servicio.

En la siguiente sección, se describe un caso de estudio donde se aplica SOA como arquitectura para desarrollar un sistema en ambiente web, utilizando servicios web, modelos de negocios y la integración con un sistema heredado.

### 3. Caso de Estudio: Ingenio San Nicolás. “Administración de Flujos de inventarios”

El departamento de Informática del ingenio San Nicolás, diseñó y desarrolló un sistema de control de producción, inventario y ventas, el cual se implementó en el año 2006. Este sistema permite capturar la producción de azúcar y de alcohol, administrar los movimientos de entradas y salidas a las bodegas, generar órdenes de embarque e impresión de sus documentos y control de ventas. El sistema opera bajo el esquema Cliente/Servidor en Visual Basic 6.0 y el administrador de la Base de Datos es Access. El sistema administrativo de control de producción, inventario y ventas del ingenio San Nicolás, administra de manera eficiente los movimientos de entrada y salida de las bodegas internas. Sin embargo la arquitectura de este sistema no permite el control de las bodegas externas ocasionando con esto problemas administrativos. El flujo de información de las bodegas externas hacia las oficinas administrativas localizadas en el Ingenio, ocasiona problemas en la generación de documentación de embarques y en los movimientos de entradas y salidas de los diferentes productos. Esto conlleva a un atraso en la integración de la información además de un alto grado de errores en la misma. Esta situación afecta a la generación de los diferentes informes requeridos por los usuarios del sistema.

Para resolver el problema de control de bodegas, se desarrollaron un conjunto de servicios Web que representan procesos de negocio tales como *Obtener datos del*

*Cliente*, el cual solicita una clave de cliente, realiza un proceso y devuelve los datos del cliente; tales como razón social, RFC, dirección, por mencionar algunos, otros servicios Web identificados son: Obtener datos de fletera, Obtener pedidos pendientes, Obtener datos de pedido, Obtener datos de productos. Estos servicios Web residen en la capa de servicios de nuestra arquitectura propuesta. A su vez, para resolver el problema de coordinación e integración se desarrollaron algunos servicios Web compuestos que residen en la capa de orquestación de nuestra propuesta. Ejemplo de estos servicios Web compuestos son: Generar orden de carga; el cual genera el documento necesario para realizar el proceso de embarque de los diferentes productos almacenados, en éste documento se incluyen los datos de clientes, fleteras, productos, cantidades, entre otros. Con este conjunto de servicios Web desarrollado se automatiza el control de flujos de inventarios de las bodegas externas vía Web, y facilita la administración de los movimientos de entrada y salida de azúcar y se controlan las existencias de las diferentes bodegas externas. Cada uno de estos servicios tienen funcionalidades adicionales que permiten integrarse con otras sistemas con el fin de actualizar datos de clientes, de fleteras, consultar pedidos, documentar embarques, generar movimientos de entrada y salida y realizar los reportes necesarios desde la ubicación de la bodegas externas. En la figura 3, se ilustra el diagrama BPM desarrollado.

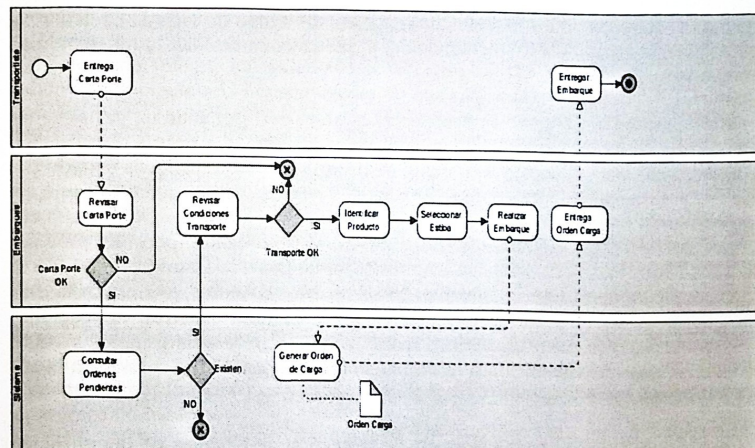


Fig. 3. Diagrama BPM (Proceso de Orden de Carga)

Este diagrama BPM tiene como objetivo principal proporcionar una notación que sea entendible por todos los usuarios empresariales, desde los analistas de negocios que crean el proyecto inicial de los procesos, hasta los desarrolladores técnicos responsables de implementar la tecnología que ejecutará y supervisará esos procesos [6]. La figura 3 ilustra principalmente tres bloques que participan en la realización

de un embarque a través del documento orden de carga. El transportista entrega un documento llamado carta porte y el personal de embarques revisa el documento y verifica que este correcto, posteriormente consulta en el sistema si existen Pedidos (Órdenes México) pendientes para este cliente, si existe se procede a revisar las condiciones del transporte y al estar éstas en buen estado se procede a embarcar, después de terminar el embarque se genera en el sistema la Orden de Carga, principalmente con los datos del producto y bodega. Terminando la orden de carga se imprime y se le entrega al transportista, él cual procede a retirarse a entregar la mercancía al lugar indicado. En este momento se termina el proceso. Existen cortes al proceso cuando no se cumplen algunas condiciones.

La coreografía de procesos de negocios se centra en la transición de los estados. El modelo de diagramas de interacción representan la forma en cómo interactúa el sistema con los servicios identificados. Estos diagramas permiten enfatizar la comunicación entre servicios y se enfocan en especificar los mensajes entre los servicios y cómo el conjunto de estos mensajes permiten obtener una funcionalidad. En la figura 4 se ilustra el flujo y las interacciones que tiene el sistema con los servicios que participan en la generación del documento Orden de Carga. En este diagrama, la aplicación Web se refiere al software que el usuario va a manipular para realizar las acciones necesarias para generar una orden de carga. La coreografía muestra los servicios que participan en la generación de la orden de carga. En la parte derecha de la figura se presentan los modelos de servicios, el tiempo de vida de cada uno de ellos y la forma en cómo van interactuando, y que en conjunto dan la funcionalidad a la aplicación.

La arquitectura de integración propuesta se validó desarrollando un portal de información de la empresa, el cual es el punto de partida para el acceso a la aplicación Web, utilizando una validación de nombre de usuario y contraseña, cada usuario tiene un perfil de permisos de acceso según la tarea a realizar, con lo que se obtiene acceso restringido a la información. Para llevar a cabo esto, se utilizó ColdFusion como servidor Web y lenguaje de programación. ColdFusion permite procesar peticiones de páginas Web, no es un administrador de Base de Datos pero interactúa de manera fácil e intuitiva con diferentes bases de datos (Access, MySQL, SQL Server), además es un lenguaje de programación que permite construir aplicaciones Web sofisticadas, se integra con HTML para utilizar interfaces de usuarios y XML para intercambio de información. Por último se utilizó Microsoft Access para la administración de datos.

En la figura 5 se muestra la pantalla de la gestión de Clientes de la aplicación Web desarrollada.

#### 4. Trabajos relacionados

Existen diversos trabajos que hacen énfasis al uso de los servicios web y de las arquitecturas orientadas a servicios con el fin de resolver problemas de integración de aplicaciones heredadas. Chung et. al. [7] propone un sistema llamado eXFlow

para la integración de procesos de negocio en el comercio electrónico B2B (*Business-to-Business*) y EAI (*Enterprise Application Integration*). Las ventajas de la adopción de los procesos de negocios como solución son las siguientes: 1) Aprovechar la reingeniería de procesos de negocio (BPR), 2) Aprovechar *Enterprise Application Integration* (EAI), 3) Mejorar el control de la gestión de procesos. Arpinar et. al. [8] propone una arquitectura para la orquestación semiautomática de servicios Web tanto para un enfoque centralizado como para uno punto-a-punto. Esta investigación de servicios compuestos apunta en la reducción de la complejidad y tiempo necesitado para generar, y ejecutar una composición y mejorar su eficacia seleccionando los mejores servicios posibles disponibles en determinado momento. En [9] se investigan y analizan los distintos enfoques y tecnologías a fin de proporcionar un modelo común de referencia para la integración de diferentes sistemas con la adopción de la Arquitectura Orientada a Servicios (SOA) definida sobre estándares abiertos. En [10] se analiza un modelo Sandwich respecto a BOA (*Business Oriented Architecture*) que aporta un conocimiento más profundo sobre el problema actual de la integración de los negocios. Tiene un marco de trabajo con características concurrentes con SOA y va mas allá de sus bases porque lo hace mas convenientemente y rápido para combinar servicios e integrar negocios. En [11] se centra en el perfeccionamiento de los servicios Web y de las arquitecturas orientadas a servicios para mejorar la disponibilidad de los servicios Web en la configuración de las empresas. Se propone usar la redundancia de servicios Web como un acercamiento alternativo para mejorar la fiabilidad y disponibilidad de servicios Web. En [12] se analiza la integración de software heredado en una SOA. En [13] se presenta una arquitectura de integración orientada a servicios. Dicha arquitectura comprende 3 módulos principales; 1) Un contenedor de *Enterprise Java Beans* (EJB); 2) Un contenedor Web; 3) Servicios J2EE. En [14] se define que un sistema de Gestión de Flujo de trabajo (SGFT) es capaz de definir, conducir y ejecutar flujos de trabajo o procesos de negocio mediante un software cuya secuencia de ejecución está basada en una representación informatizada de dichos procesos de negocio. La SOA propuesta en este trabajo considera a las aplicaciones como servicios que se pueden descubrir y acceder desde una red, los cuales ofrecen una funcionalidad para el negocio ocultando los detalles de su implementación. Finalmente, en [16] se define la Administración de Procesos de Negocios (*Business Process Management, BPM*) como una disciplina administrativa que ve los procesos de negocio como activos susceptibles de modelarse, automatizarse, integrarse, monitorizarse y optimizarse de forma continua mediante el modelado de procesos y la aportación de métricas, identificando tareas innecesarias y cuantificando los servicios IT en términos de plazos y consumos de recursos.

## 5. Conclusiones

SOA es una arquitectura que permite integrar software heredado con nuevas tecnologías de desarrollo de software. Sin embargo, es importante tener definida la arquitectura que permita tener un marco de referencia que muestre cómo interactúan los componentes con los usuarios del negocio y con los sistemas heredados. En este trabajo, se presentó una arquitectura orientada a servicios basada en capas que per-

mite la integración de aplicaciones. Dicha arquitectura propuesta se validó mediante la implementación de un sistema de administración de flujos de inventarios. Con el desarrollo de esta arquitectura, el sistema desarrollado permite mejorar los procesos internos de administración de flujos de inventarios y agiliza la emisión de informes y la documentación necesaria para los embarques y/o recepción de productos, logrando con esto mayor eficiencia en la administración de las bodegas y generando reportes en tiempo real disminuyendo el tiempo de integración de la información.

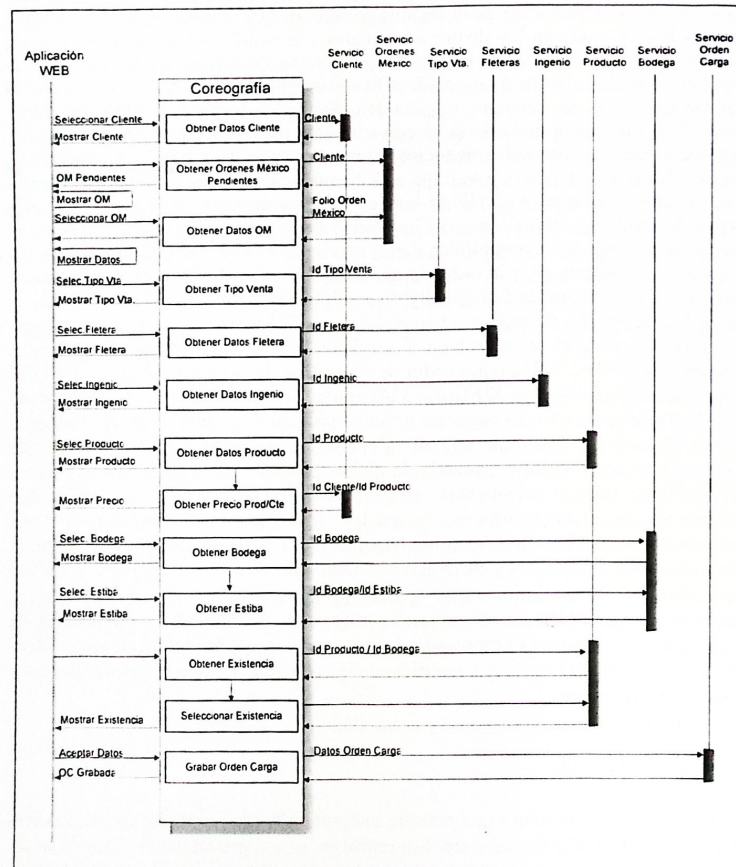


Fig. 4. Diagrama de Interacción Orden de Carga

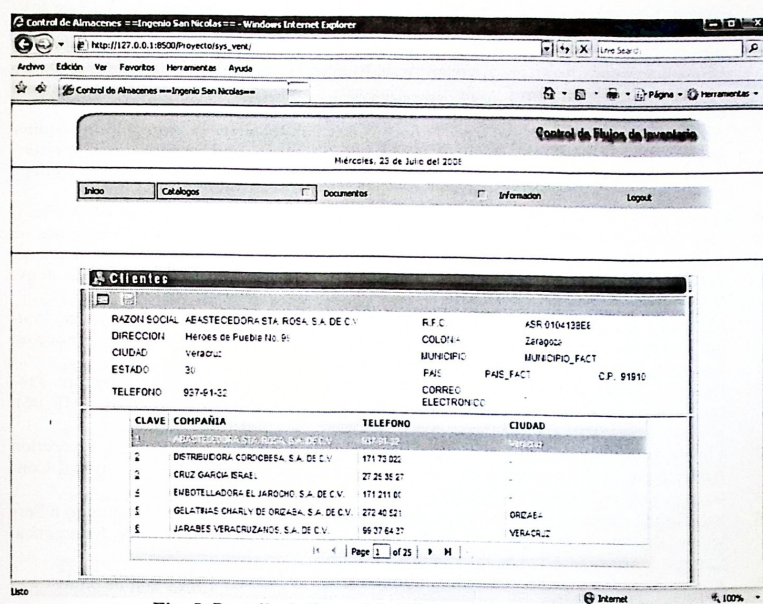


Fig. 5. Pantalla de Control de Clientes de la aplicación Web.

## Agradecimientos

Este trabajo fue financiado económicamente por la Dirección General de Educación Superior Tecnológica (DGEST) y por el Programa de Mejoramiento del Profesorado (PROMEP) de la Secretaría de Educación Pública (SEP).

## Referencias

1. Jeremy Allaire (2002) Macromedia MX: Componentes y servicios web. [http://www.adobe.com/es/devnet/components\\_ws.pdf](http://www.adobe.com/es/devnet/components_ws.pdf) (acceso diciembre 2007)
2. Arquitectura Orientada a Servicios. <http://www.es.wikipedia.org> (acceso diciembre 2007)
3. Olaf Zimmermann et al. Elements of Service-Oriented Analysis and Design (2004)
4. Ariel Schwindt [http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_3505/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3505/default.aspx)
5. TIBCO, Enabling Real-Time Business Through A Service-Oriented and Event-Driven Architecture
6. Stephen A. White, IBM Corporation, <http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf> (acceso mayo 2008)

7. Nathan Chung-Nin Chung, Wen-Shih Huang, Tse-Ming Tsai and Seng-cho T. Chou. eX-Flow: A Web Services-Compliant System to Support B2B Process Integration. Proceedings of the 37th Hawaii International Conference on System Sciences 2004. IEEE Press.
8. I. Budak Arpinar, Boanerges Aleman-Meza, Ruoyan Zhang and Angela Maduko. Ontology-Driven Web Services Composition Platform. Proceedings of the IEEE International Conference on E-Commerce Technology. IEEE Press.
9. Aurélie Aurilla, Bechina Arntzen. Services Oriented Architecture: integration requirements. 31st Annual International Computer Software and Applications Conference (COMPSAC 2007) IEEE 2007. ØSIR, Faculty of computer Science, College University I Hedmark. IEEE Press.
10. Juan Zhu, Li Zhen Zhang. A Sandwich Model for Business Integration in BOA (Business Oriented Architecture). Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06) IEEE 2006. Department of Computer Applications, School of Computer Science & Engineering, South China University of Technology. IEEE Press.
11. Michael Jiang, Allan Willey. Service-Oriented Architecture for Deploying and Integrating Enterprise Applications. Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05). IEEE Press.
12. Harry M. Sneed. Integrating legacy Software into a Service oriented Architecture. Proceedings of the Conference on Software Maintenance and Reengineering (CSMR'06). Vienna, Austria. IEEE Press
13. Thilina Gunasinghe, Tim Kelly. Establishing a Standard Business Process Execution Architecture for Integrating Web Services. Proceedings of the IEEE International Conference on Web Services (ICWS'05). IEEE Press.
14. Ing. Jose O. Hurtado Miranda. Sistemas de Gestión de Flujo de trabajo Orientado a Servicios. Doctorado en Ingeniería de Software. Universidad Pontificia de Salamanca, Campus Madrid, Facultad de Informática
15. Andrea Vega. : Lo bueno y lo malo. Information Week Mexico 161. Mayo 5, 2007