

Universidad Central “Marta Abreu” de Las Villas
Facultad Matemática, Física y Computación
Licenciatura en Ciencia de la Computación



TRABAJO DE DIPLOMA

*Arquitectura Orientada a Servicios para la gestión automatizada del
proceso de edición de una revista arbitrada*

AUTOR: Yuniesky Abreu Collazo

TUTOR: Gheisa Ferreira Lorenzo

CONSULTANTE:

“Año del 50 Aniversario del Triunfo de la Revolución”

Dictamen

Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor:

Firma del jefe del Laboratorio:

“He hecho de mi lo que no sabía,
y lo que podía hacer de mi no lo he hecho”

Fernando Pessoa

A mis padres por su confianza y apoyo.

A la familia y amigos por mostrarme su apoyo incondicional.

Yuniesky Abreu Collazo.
Julio 2009.

El presente trabajo esta basado en la modelación y descripción de las tareas fundamentales involucradas en la edición y publicación de una revista arbitrada. Modelar estos procesos para lograr una Gestión de los Procesos de Negocio e integrar los mismos a nuevas tecnologías como las Arquitecturas Orientadas Servicios.

En este documento se presentan los conceptos necesarios a tener en cuenta para crear dicha arquitectura, para el proceso de negocio de una revista arbitrada. Considerando que el problema principal de los responsables del diseño de aplicaciones, radica en cómo utilizar todos estos conceptos para encapsular funcionalidades del negocio dentro de servicios. De esta manera se puede obtener un producto de alto rendimiento, disponibilidad y escalabilidad.

Se escogió para realizar la modelación de los procesos de negocio la herramienta de representación de flujo de trabajo YAWL (Yet Another Workflow Language) que posee una fuerte validación estructural y matemática del proceso. A partir del modelo de negocio se crea una arquitectura orientada a servicios para el propio proceso con el objetivo de lograr interoperabilidad.

The present work is based on the modeling and description of the fundamental tasks involved in the edition and publication of an arbitrated magazine.

To shape these processes to achieve a Business Processes Management and to integrate the same ones to new technologies as the Service Oriented Architecture.

In this document the necessary concepts appear to bearing in mind to create the above mentioned architecture, for the process of business of an arbitrated magazine. Considering that the principal problem of the design responsible ones of applications, it is represented in how use all this concepts agrupate functionalities of bussines inside the services. This way, it is obtained a product with a high performance, availability and scalability.

It was chosen to realize the modeling of the processes of business the tool of representation of workflow YAWL (Yet Another Workflow Language) that possesses a strong structural and mathematical ratification of the process. From the model of business there is created an architecture faced to services for the proper process with the target to achieve interoperability.

ÍNDICE

Introducción	1
Objetivo General.....	2
Preguntas de Investigación	2
Objetivos Específicos	2
CAPITULO I: REFERENTES TEORICOS ACERCA DE LA GESTION DE PROCESOS DE NEGOCIO Y LAS ARQUITECTURAS ORIENTADAS A SERVICIOS.	4
1.1 Gestión de procesos de negocio.....	4
1.1.1 Flujo de trabajo y Sistemas de Administración de Flujos de Trabajo	4
1.1.2 Modelo de referencia de flujo de trabajo	6
1.1.3 Especificación de un flujo de trabajo	8
1.1.4 Proceso de negocio	10
1.1.5 Enfoque de Gestión de Procesos de Negocio	10
1.2 SOA: Arquitectura Orientada a Servicios.....	14
1.2.1 Surgimiento e importancia	14
1.2.2 Diseño basado en SOA.....	17
1.2.3 Componentes fundamentales en una arquitectura orientada a servicios	19
1.2.4 Entidades en una arquitectura orientada a servicios.....	20
1.2.5 Relación entre servicios y comunicación	22
1.2.6 Principios básicos para el diseño de los servicios	24
1.2.7 Capas principales en una arquitectura orientada a servicios	25
1.2.8 Diseño Orientado a Servicios	26
1.3 Tecnologías asociadas a la coordinación de servicios web	28

1.3.1 Lenguajes y Estándares	29
1.4 Consideraciones Finales del Capítulo.....	31
CAPITULO II: MODELADO DEL NEGOCIO	33
2.1 Lineamientos generales para el descubrimiento y diseño de los procesos de negocio.....	33
2.2 Organización del trabajo editorial en una revista arbitrada	34
2.2.1 Personal involucrado en el trabajo editorial.....	35
2.2.2 Procesos de negocio fundamentales en el trabajo editorial.....	37
2.3 Modelación de los procesos de negocio	38
2.3.1 Modelo general de procesos del negocio	39
2.3.2 Especificación de los procesos generales del negocio	41
2.3.3 Especificación de los subprocesos del negocio.....	43
2.4 Conclusiones del capítulo	51
CAPITULO III: ARQUITECTURA ORIENTADA A SERVICIOS PARA LA GESTION AUTOMATIZADA DEL PROCESO DE EDICION DE UNA REVISTA ARBITRADA.	52
3.1 Análisis orientado a servicios	52
3.2 Diseño orientado a servicios.....	57
3.3 Desarrollo de servicios web.....	58
3.3.1 Definición del WSDL para el servicio Aceptación Preliminar de los Artículos.....	61
3.3.2 Definición del WSDL para el servicio Manejo Datos Autor	65
3.4 Conclusiones del capítulo	69
CONCLUSIONES Y RECOMENDACIONES	70
BIBLIOGRAFIA	71

Anexo	73
Anexo I. Patrones de Flujo de Trabajo	73
Anexo II. Pasos del Análisis Orientado a Servicios	74
Anexo III. Pasos del Modelo Orientado a Servicios	75
Anexo IV. Diseño Orientado a Servicios	76
Anexo V. Código de descripción de servicios	77

Introducción

Los procesos de revisión y edición en las revistas digitales e impresas, en especial aquellas que necesitan de un arbitraje para la publicación de sus artículos, se realizan con un alto costo en recursos y en tiempo. Someter estos procesos a la automatización con el apoyo de herramientas computacionales puede conducir a obtener ganancias y minimizar las pérdidas de la organización.

Tal es el caso de la Revista Cubana de Ciencias Informáticas (UCI, 2009) y la Revista Ciencias Agrícolas (Agrícola, 2009), entre otras, que tienen a su cargo la divulgación de trabajos de varias temáticas.

Estas revistas, como otras que pueden encontrarse en línea en Internet (Zelkowitz, 2009) (ACM, 2009a) (ACM, 2009b) (ACM, 2009c) (ACM, 2009d), se caracterizan por mantener un sitio web donde se publican las características generales de la revista, las posibilidades de subir artículos a ser analizados para publicar, y otros servicios fundamentalmente de divulgación. Sin embargo, no se localizan acciones encaminadas a establecer un proceso de gestión y control de las publicaciones, es decir, administrar, de alguna manera, como es que internamente se desempeñan las actividades relacionadas con la publicación final de un número de determinada revista.

Durante años, el principal uso de las computadoras en las organizaciones ha sido la automatización de las actividades individuales dentro de las mismas, por lo que el hombre ha sentido la necesidad de adaptarse a los cambios en los procesos internos que experimentan las organizaciones, produciendo como resultado un cambio de orientación que apunta hacia los procesos organizacionales o procesos de negocio, mejorándose de una forma u otra la toma de decisiones.

La gestión de los procesos de negocio a través de la automatización, vista en términos de capturar, modelar, ejecutar y monitorizar los procesos de negocio, como un conjunto de procedimientos o actividades enlazadas, cuya realización permite alcanzar un cierto objetivo o meta en el contexto de una organización, marca en buena medida la competitividad.

En el presente trabajo el modelado de procesos de negocio (*BPM – Business Process Modeling*) y las tecnologías asociadas, tales como la Arquitectura Orientada a Servicios (*SOA – Service Oriented Architecture*), brindan un marco adecuado para abordar el problema de la gestión automatizada de los procesos de negocio de una revista arbitrada, para lo cual se plantea el objetivo general siguiente:

Objetivo General

Diseñar una arquitectura orientada a servicios que permita administrar el proceso de edición de una revista arbitrada.

Para orientar la labor investigativa, en este trabajo se plantean como preguntas de investigación:

Preguntas de Investigación

- ¿Cuáles son las tareas fundamentales que están involucradas en la aceptación y publicación de artículos en una revista arbitrada?
- ¿En qué medida las herramientas de flujo de trabajo disponibles son factibles para realizar la modelación de los procesos que intervienen en la revista arbitrada y cuál puede ser la más apropiada en el problema que se plantea?
- ¿Cómo definir una arquitectura orientada a servicios que de respuesta a los procesos de negocio involucrados?
- ¿Qué tecnologías de desarrollo de servicios web pueden ser utilizadas para la implementación de los servicios para la arquitectura a desarrollar?

Partiendo de estas interrogantes, los objetivos específicos a emprender son las siguientes:

Objetivos Específicos

- Identificar las tareas que aparecen en la aceptación y publicación de artículos en una revista arbitrada.

- Modelar los procesos de negocio que se definen en la revista utilizando una herramienta de software libre.
- Proponer una arquitectura orientada a servicios para la gestión automatizada de los procesos de la revista arbitrada.
- Implementar en un lenguaje de descripción de servicios web (WSDL) la interfaz de comunicación de los servicios propuestos.

La temática abordada posee gran importancia y actualidad.

Los modelos de procesos sirven como medio de comunicación entre analistas de negocio e ingenieros en sistemas, permite cambiar procesos de negocio sin modificar el código de los sistemas que soportan las tareas de los procesos, además, la representación explícita de los procesos permite que los mismos puedan ser ejecutados y automatizados a partir de la integración de los procesos de negocio con las arquitecturas orientadas a servicios. Desde el punto de vista de los procesos de negocio los servicios web en el marco de las arquitecturas orientadas a servicios suponen un mecanismo de comunicación idóneo para la integración de aplicaciones heterogéneas en los procesos de negocio internos de una organización.

El presente informe incluye, además de esta introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos. En el capítulo I Se introducen las ideas básicas de la evolución del flujo de trabajo y los sistemas de administración de flujos de trabajo hacia la gestión de los procesos de negocio de una organización y se enlaza este enfoque con las actuales arquitecturas orientadas a servicios. El capítulo II aborda las ideas básicas del descubrimiento del modelo de negocio en el trabajo editorial de una revista arbitrada, el cual constituye la base para el diseño de una arquitectura orientada a servicios. En el capítulo III se presenta una propuesta de arquitectura orientada a servicios para la gestión automatizada de los procesos de negocio de una revista arbitrada y se desarrolla la descripción de la interfaz de cada uno de los servicios identificados.

CAPITULO I: REFERENTES TEORICOS ACERCA DE LA GESTION DE PROCESOS DE NEGOCIO Y LAS ARQUITECTURAS ORIENTADAS A SERVICIOS.

El presente capítulo se organiza en un conjunto de secciones cada una de las cuales presenta el estudio de un concepto, definición o tecnología de particular interés para el desarrollo de este trabajo. Se introducen las ideas básicas de la evolución del flujo de trabajo y los sistemas de flujos de trabajo hacia la gestión de los procesos de negocio de una organización (*BPM - Business Process Management*) y se enlaza este enfoque con las actuales arquitecturas orientadas a servicios (*SOA - Service Oriented Architecture*). Por último se discuten algunas tecnologías de interés asociadas a la coordinación de servicios.

1.1 Gestión de procesos de negocio

El término BPM aparece en múltiples ocasiones asociado a la tecnología de flujo de trabajo (*workflow*) ya que de manera general se puede ver como una evolución natural de esta tecnología, con el fin de adaptarse a las necesidades actuales de los procesos de negocio. En los siguientes epígrafes se desarrolla el concepto de *workflow* y uno de los principales modelos que se encuentran detrás de él, el Modelo de Referencia, para posteriormente establecer un enfoque asociado a BPM.

1.1.1 Flujo de trabajo y Sistemas de Administración de Flujos de Trabajo

Workflow Management Coalición (WfMC), una de las principales organizaciones en el mundo que trabaja en temas de flujo de trabajo, lo define de la siguiente manera: “un conjunto de uno o más procedimientos o actividades directamente ligadas, que colectivamente realizan un objetivo del negocio, normalmente dentro del contexto de una estructura organizacional que define roles funcionales y relaciones entre los mismos”.

Un sistema de administración de flujo de trabajo (*WMS - Workflow Management System*) es “un sistema que define, crea y gestiona la ejecución de flujos de trabajo mediante el uso de software, siendo capaz de interpretar la definición del proceso, interactuar con los participantes y, siempre que se requiera, invocar el uso de herramientas y aplicaciones”. Algunos ejemplos claros de utilización de los sistemas de este tipo pueden ser: la gestión documental, los servicios de gestión de personal (solicitud de las vacaciones, hoja de gastos de desplazamiento, etc.), el control de la producción o los procesos colaborativos. Los WMS pueden ser implementados de diferentes formas como resultado de la utilización de diferentes tecnologías y plataformas. Pueden ir desde un pequeño grupo de trabajo a una gran organización. No obstante, todos los WMS exhiben ciertas características comunes. En el nivel más alto, los WMS pueden ser caracterizados por proporcionar tres áreas de funcionalidad (Figura 1):

- *Funciones de tiempo de Construcción (Build-time functions)*, dedicadas a la definición y modelado de un proceso junto con todas sus actividades concernientes.
- *Funciones de control en tiempo de ejecución (Run-time control functions)*, las cuales controlan el proceso en el ambiente de ejecución, llevando a cabo cada tarea (o actividad) definida como parte del proceso.
- *Funciones de interacción en tiempo de ejecución (Run-time interaction)*, que interactúan con los usuarios o aplicaciones externas para que los participantes del proceso puedan llevar a cabo sus tareas.

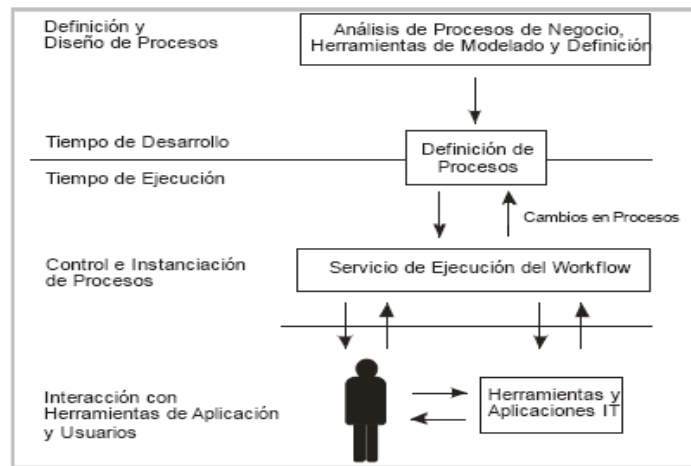


Figura 1. Caracterización de los WMS por la WfMC

Existen diferentes líneas relacionadas con los WMS, que abarcan desde temas relacionados tanto con la definición de flujos de trabajo como con su ejecución (Members., 2009).

1.1.2 Modelo de referencia de flujo de trabajo

En 1995 la WfMC publica un documento titulado "The Workflow Reference Model" que define conceptos básicos asociados a la arquitectura de los WMS.

El Modelo de Referencia propuesto por la WfMC (Figura 2) intenta reunir las características comunes a cualquier WMS, para poder alcanzar la interoperabilidad entre este tipo de sistemas, a través de estándares comunes para cada una de las funciones que se puedan realizar.

Para lograr esto identifica las distintas áreas funcionales para luego desarrollar las especificaciones para la implementación de las mismas, lo que asegura tanto la interoperabilidad entre WMS y su integración con otras aplicaciones informáticas.

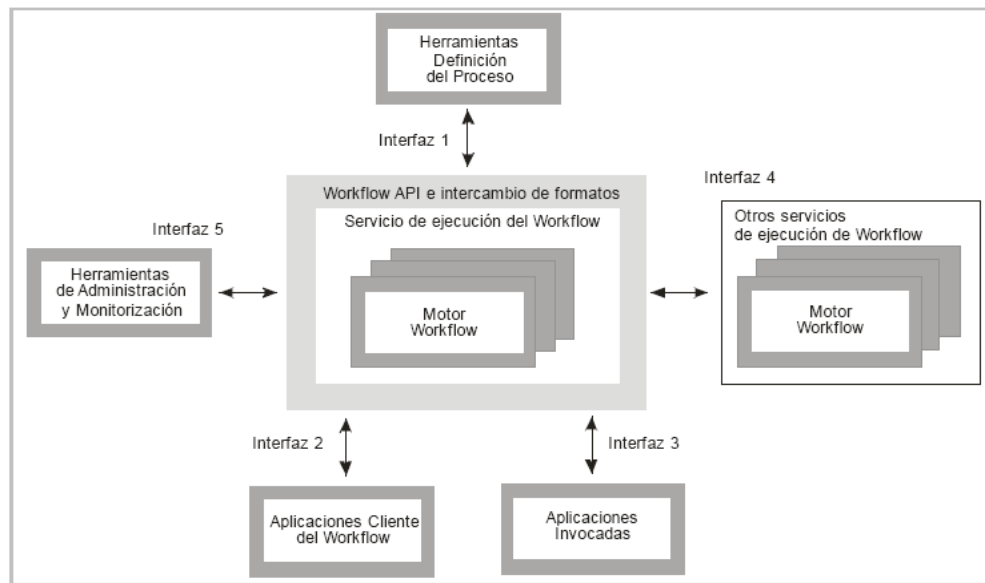


Figura 2. Modelo de referencia al WfMC

El componente central de este modelo de referencia es el **Servicio de Ejecución del Workflow** que se encarga de crear, gestionar y ejecutar cada una de las instancias del modelo de flujo de trabajo. En este componente se encuentra el motor del WMS que proporciona la ejecución de cada instancia del flujo de trabajo.

Otro de los componentes es el de **Herramientas de Definición del Proceso** o flujo de trabajo, que permiten modelar, describir y documentar un determinado flujo de trabajo.

La **Interfaz 1** permite la comunicación entre este componente y el servicio de ejecución del flujo de trabajo. La distinción de ambos componentes aporta beneficios, entre los que se destacan la separación entre el entorno de ejecución y el de definición.

La **Interfaz 1** se encarga por tanto del intercambio de información entre el componente que permite la definición del proceso y el propio servicio de ejecución del flujo de trabajo.

El componente denominado **Aplicaciones Cliente del Workflow** representa las entidades de software utilizadas por el usuario final en aquellas actividades que requieren participación humana para su realización. Si este componente se separa de

lo que es el propio componente de ejecución, es necesaria una interfaz (**Interfaz 2**) que defina y maneje claramente el concepto de lista de trabajos, como una cola de trabajo asignada a un usuario o a un grupo de usuarios por el propio motor de ejecución del flujo de trabajo.

El componente **Aplicaciones Invocadas** representa software o aplicaciones ya existentes que un WMS puede utilizar para la realización de ciertas actividades, teniendo en cuenta que, en principio, dichas aplicaciones se pueden encontrar en cualquier plataforma o lugar de la red. La **Interfaz 3** permite la comunicación entre este componente y el servicio de ejecución del flujo de trabajo, no sólo a nivel de invocación del mismo, sino de transformación de datos en formatos entendibles por ambos componentes.

La interoperabilidad entre WMS está representada por el componente denominado **Otros Servicios de Ejecución de Workflow**, siendo la **Interfaz 4** la que permite dicha comunicación. En este caso, la WfMC ha desarrollado un conjunto de escenarios de interoperabilidad que van desde la conexión a nivel de actividades simples hasta un completo intercambio de definición de procesos y datos.

Finalmente, el componente **Herramientas de Administración y Monitorización** permite que distintos servicios de ejecución de flujo de trabajo compartan las mismas funciones de administración y monitorización del sistema, como pueden ser, por ejemplo, la gestión de usuarios, el control de los recursos y la supervisión del estado de todo el proceso.

1.1.3 Especificación de un flujo de trabajo

La aparición masiva de numerosos WMS puso en evidencia aun más la falta de un modelo formal universal para la especificación de un proceso de negocio.

La especificación de un flujo de trabajo puede ser explicada en sentido general desde diferentes perspectivas:

Perspectiva de Control de Flujo: describe actividades y su orden de ejecución mediante diferentes constructores que permiten controlar el flujo de ejecución (joins,

splits, secuencias, paralelismo, etc.). Estas actividades se pueden ver como unidades atómicas de trabajo.

Perspectiva de Datos: describe los datos (documentos, objetos, etc.) que fluyen entre las diferentes actividades. Estos datos también pueden ser variables locales que definen pre y post condiciones en la ejecución de tareas.

Perspectiva de Recursos: muestra una visión más orientada al negocio, describiendo el proceso en función de las responsabilidades que tienen las diferentes personas o dispositivos en la ejecución de una determinada tarea.

Perspectiva Operacional: muestra las acciones elementales que se realizan dentro de las actividades, tales como invocar un determinado servicio de una aplicación con determinados datos.

Todas las perspectivas presentan una visión diferente del mismo sistema. La perspectiva de control de flujo provee un mejor panorama para la especificación de un flujo de trabajo describiendo más ampliamente el proceso en sí mismo. La perspectiva de datos sólo se apoya en la mencionada anteriormente, mientras que la operacional es más bien complementaria. Es por esto que los estudios se enfocan en la perspectiva de control de flujo (Gramaje, 2002).

Investigaciones actuales apuntan a describir los requisitos de un flujo de trabajo mediante patrones (Aalst et al., 2007). Los patrones son independientes de las tecnologías y permiten abstraer los requerimientos específicos del dominio al que se están enfocando. Como resultado se han identificado más de veinte patrones que describen el comportamiento de los procesos de negocios. Estos patrones van desde patrones sencillos hasta muy complejos y se pueden organizar en seis grupos.

- Patrones de control básico
- Patrones de bifurcación y sincronismo avanzados
- Patrones que involucran múltiples instancias
- Patrones estructurales

- Patrones basados en el estado
- Patrones de cancelación

El contar con una especificación de alto nivel como esta permite comparar diferentes productos y tecnologías de modelado de procesos, analizando el soporte que cada una de ellas provee para la modelación de los procesos de negocio.

1.1.4 Proceso de negocio

Un proceso de negocio es el conjunto estructurado de tareas que contribuyen a lograr los objetivos de una organización. Los mismos son parte indispensable de la base operativa de una empresa, por lo que el éxito de la misma depende de la eficiencia con que sean gestionados. Una gestión inadecuada de los procesos de negocio puede conducir a altos costos, baja productividad e inadecuados tiempos de respuesta tanto a las oportunidades como a las amenazas.

1.1.5 Enfoque de Gestión de Procesos de Negocio

Si bien la sigla BPM puede tener diferentes significados según el contexto en que se esté manejando, a los efectos del epígrafe que se presenta a continuación se referirá a la Gestión de Procesos de Negocios (*BPM - Business Process Managment*).

En general, el ciclo de vida para la Gestión de los Procesos de Negocio puede dividirse en las siguientes etapas (agenda, 2009):

- Estudio: análisis de los objetivos y organización de la compañía para poder convertir el conjunto de actividades en procesos de negocio.
- Modelar los procesos del negocio: usar el software y hardware para crear un modelo que sea lo más realista posible.
- Implementación de una solución: implementar una solución BPM vinculada al sistema de información de la compañía (aplicaciones y bases de datos existentes).
- Ejecución: es la etapa operacional durante la cual se implementa la solución BPM.

- Control: consiste en analizar el estado de los procesos con indicadores que muestren la eficacia de los mismos.
- Optimización: ofrecer soluciones para mejorar el funcionamiento de los procesos de negocio.

Se define Gestión de Procesos de Negocio como el conjunto de actividades que realizan las organizaciones para optimizar o adaptar sus procesos de negocio a las nuevas necesidades organizacionales. Involucra el descubrimiento, diseño y distribución de procesos de negocio, así como el control ejecutivo, administrativo y supervisión de dichos procesos. Tiene que ver entonces con manejar el cambio para mejorar los procesos de negocio, que por años han sido gestionados con distintas técnicas y herramientas, pero sin estándares definidos y ciclo de vida completo para diseñarlos y ejecutarlos. El manejo del cambio requiere control y entendimiento de los procesos, y para eso son necesarios estándares de modelado y ejecución de procesos.

Business Process Management Initiative (BPMI) ha promovido el uso de BPM en las últimas décadas y ha desarrollado especificaciones abiertas para el diseño, ejecución, mantenimiento y optimización de procesos de negocio, es decir, para cubrir todas las expectativas del Modelo de Referencia. El modelo de referencia promueve estándares para el modelado, definición y ejecución de procesos del Negocio: *Business Process Modeling Notation* (BPMN), para modelado y especificación de procesos; *Business Process Modeling Language* (BPML), para ejecución de procesos y *Business Process Query Language* (BPQL), para distribución y ejecución de procesos, como interface de gestión estándar. Los procesos de Negocio especificados en BPMN y traducidos a BPML serán entonces ejecutados por motores de procesos en *Business Process Management Systems* (BPMS).

BPMN es una notación estándar para modelar visualmente flujos de procesos que tiene como objetivo proveer notación común para analistas del negocio que crean los flujos iniciales de los procesos y desarrolladores de software responsables por la tecnología e implementación de los procesos. Está basado en diagramas de actividad

del Lenguaje de Modelado Unificado UML y diagramas de flujo actividad-decisión. Especifica un único tipo de diagrama, *Business Process Diagram* (BPD).

BPML es una notación estándar para lenguajes de ejecución de procesos basado en XML, que establece un formato estándar para expresión e intercambio de procesos independiente de la implementación. BPMN mapea directamente, sobre BPML y otros como BPEL4WS. El lenguaje desarrollado tiene una base matemática rigurosa con el objetivo de que los sistemas construidos sobre éste puedan ser igual de resistentes que los construidos sobre bases de datos. Para esto se emplea semántica declarativa basada en cálculo de procesos y modelo de procesamiento concurrente. BPML define un estándar para procesos, cubriendo aspectos como actividades del negocio de complejidad variable, transacciones de negocio y sus compensaciones, manejo de datos del proceso, concurrencia, manejo de excepciones y semántica operacional. Este estándar de modelado formal de procesos, deberá ser soportado por los sistemas de administración de flujos de procesos de negocio (*BPMS- Business Process Management System*) para su ejecución y exposición al negocio de los procesos vía lenguaje de consulta de procesos y herramientas de modelado de procesos. Estas herramientas deberán permitir realizar el modelado de los procesos con BPMN que será traducido directamente a BPML para su ejecución, como se muestra en la Figura 3.

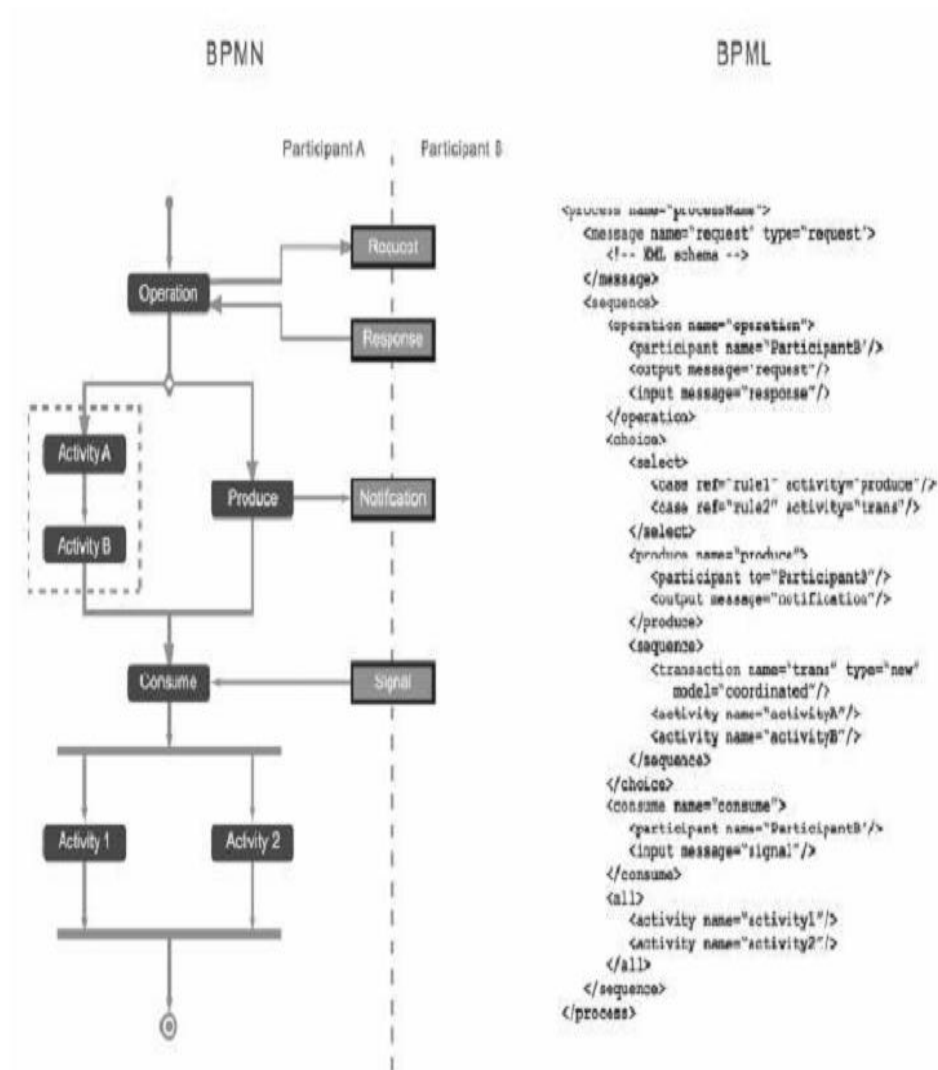


Figura 3. Ejemplo de mapeo entre BPMN y BPML

Aún cuando esta y otras organizaciones como la OMG (*Object Group Management*) han centrado el desarrollo de estándares de este tipo, se nota una apreciable apertura hacia el desarrollo basado en software libre de este tipo de herramientas, como se establece en el epígrafe 1.3.

Aunque en múltiples ocasiones los conceptos de *workflow* y BPM se confunden pues ambos contienen una representación de un flujo de trabajo, es decir, un flujo de trabajo desde el punto de vista lógico puede ser representado por ambos en algún

lenguaje de notación y de especificación, entre ambos conceptos existe una diferencia esencial.

La diferencia fundamental radica en que, en general, los WMS no cumplen con los requisitos de escalabilidad que requieren las organizaciones hoy en día y en general están limitados a interactuar con aplicaciones dentro de la organización. Por el contrario, BPM no sólo define las tareas y el seguimiento de las mismas, sino que tiene como objetivo solucionar el ciclo de vida completo de un proceso de negocio.

1.2 SOA: Arquitectura Orientada a Servicios

La Arquitectura Orientada a Servicios (*SOA - Service Oriented Architecture*), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. En los epígrafes que continúan se tratan algunos aspectos generales referentes a este tipo de arquitectura y su vínculo directo con BPM.

1.2.1 Surgimiento e importancia

La arquitectura orientada a servicios es el siguiente paradigma arquitectónico luego del advenimiento del modelo cliente/servidor y las aplicaciones basadas en Internet. La diferencia con esos paradigmas es que en lugar de focalizarse en la tecnología, hace foco en cómo organizar los negocios. Al encontrarse ahora la gente de tecnología hablando en el mismo lenguaje que la gente de negocio, se resuelve un gran problema de comunicación. No existen estándares que brinden un modelo y alcance a SOA (López, 2008).

Un servicio web es una pieza de software que conforma una serie de estándares de intercambio de información. Estos estándares permiten el intercambio de operaciones entre diferentes tipos de computadoras, apartándose del problema del hardware que utilicen, de los sistemas operativos que estén corriendo en dichos equipos, o de los lenguajes de programación en los que estén escritos. Para mantener su independencia, los servicios web, encapsulan la lógica dentro de un contexto. Este contexto puede ser

una tarea de negocio, una entidad de negocio o alguna otra agrupación lógica (Erl, 2005).

Cuando se construye una solución con arquitectura orientada a servicios, cada servicio puede encapsular una tarea realizada por una rutina individual, o un subproceso compuesto de varias rutinas. Un servicio puede además encapsular la lógica completa del proceso, tal como indica la Figura 4.

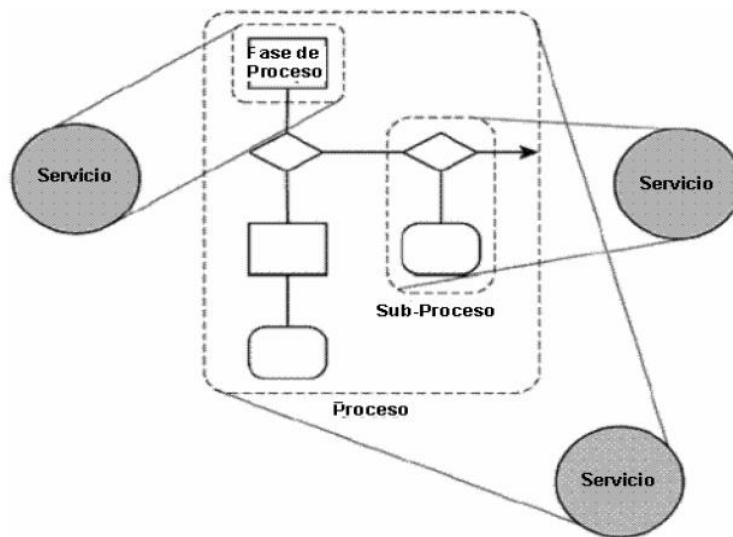


Figura 4. Encapsulamiento de procesos

El tamaño y alcance de la lógica representada por los servicios puede variar. Además, la lógica de un servicio web puede requerir de la lógica provista por otros servicios web. En este caso un servicio web se valdrá de otros para resolver su problema. Una arquitectura orientada a servicios representa una metodología para lograr interoperabilidad entre aplicaciones y servicios web de manera tal que permita reutilizar tecnologías existentes.

Generalmente, los servicios incluyen tanto lógica de negocios como manejo de estado (datos), relevantes a la solución del problema para el cual fueron diseñados. La manipulación del estado es gobernada por las reglas de negocio.

Un servicio funciona como una aplicación independiente, teniendo sus propias reglas de negocio, datos, procedimientos de administración y operación. Expone toda su funcionalidad utilizando una interfaz basada en mensajes.

La comunicación hacia y desde el servicio, es realizada utilizando mensajes y no llamadas a métodos. Estos mensajes deben contener o referenciar toda la información necesaria para ser entendidos. Al contrario de las arquitecturas orientadas a objetos, las arquitecturas orientadas a servicios están formadas por servicios de aplicación débilmente acoplados y altamente interoperables.

Visto del lado del negocio, una arquitectura orientada a servicios es una técnica de desarrollo de aplicaciones de alto nivel que le permite focalizar los procesos de negocio, en la infraestructura, para alcanzar una ventaja competitiva. Las industrias que implementan satisfactoriamente SOA, poseen una ventaja competitiva importante sobre las industrias que no, porque las que tienen sus servicios alineados con los negocios, pueden reaccionar más rápido a los cambios en los requerimientos de negocio que las que no los tienen. SOA es una arquitectura en la cual una aplicación se constituye de servicios que se exponen y servicios que se consumen; difiere del tradicional enfoque cliente/servidor haciendo énfasis en el bajo acoplamiento entre los componentes de software.

Dado que estos servicios pueden ser consumidos por diferentes sistemas y plataformas, las características de los servicios web son ideales para implementar esta solución.

Para dar solución a algunos de los problemas que en sus inicios presento la tecnología de flujo de trabajo, la tecnología BPM surge y se desarrolla apoyada fuertemente en el concepto de arquitectura orientada a servicios. Exponer procesos de negocio como servicios es la clave de la flexibilidad de esta arquitectura. La misma define los servicios de los cuales estará compuesto el sistema, sus interacciones, y con qué tecnologías serán implementados. Las interfaces que utiliza cada servicio para exponer su funcionalidad son gobernadas por contratos, que definen claramente el conjunto de mensajes soportados, su contenido y las políticas aplicables.

1.2.2 Diseño basado en SOA

La metodología de modelado y diseño para aplicaciones SOA se conoce como análisis y diseño orientado a servicios (Erl, 2005). Para que un proyecto SOA tenga éxito, los desarrolladores de software deben orientarse a crear servicios comunes que son orquestados por clientes o por un módulo intermedio para implementar los procesos de negocio. El desarrollo de sistemas usando SOA requiere un compromiso con este modelo en términos de planificación, herramientas e infraestructura.

Por otro lado, la implementación ideal de un servicio exige resolver algunos inconvenientes técnicos inherentes a su modelo:

- Los tiempos de llamado no son despreciables, gracias a la comunicación de la red, tamaño de los mensajes, etc. Esto necesariamente implica la utilización de mensajería confiable.
- La respuesta del servicio es afectada directamente por aspectos externos como problemas en la red, configuración, etc. Estos deben ser tenidos en cuenta en el diseño, desarrollándose los mecanismos de contingencia que eviten la parálisis de las aplicaciones y servicios que dependen de él.
- Comunicaciones no confiables, mensajes impredecibles, reintentos, mensajes fuera de secuencia, etc.

A su vez, cuando se usan múltiples servicios para implementar un sistema, es muy fácil que la comunicación entre estos se salga de control. Por ejemplo, se puede tener un servicio que llama a otros seis servicios, algunos de los cuales llaman a otros servicios, y de esta manera, muy fácilmente el sistema se vuelve inmanejable. De esta forma, un sistema grande puede terminar con múltiples dependencias por lo que detectar un problema de rendimiento o funcionalidad se puede volver muy complicado. Según lo expresado, si no se cuenta con una *estrategia* adecuada, se puede llegar a una implementación donde exista una explosión de dependencias entre los diferentes servicios.

En la bibliografía consultada (Erl, 2005) aparece una clasificación que establece tres estrategias fundamentales para desarrollar arquitecturas orientadas a servicios.

La **estrategia top-down** es utilizada para construir soluciones SOA y genera productos de alta calidad. La arquitectura resultante con este enfoque será óptima debido a que se comienza analizando el flujo de negocio de manera integral, para luego bajar el nivel de detalle hasta los servicios a implementar. Nótese que en este tipo de estrategia, el modelado de procesos de negocios constituye el primer paso en la definición de una arquitectura orientada a servicios. La principal desventaja de este enfoque es el presupuesto y tiempo. Realizar un análisis global del negocio y del conjunto completo de servicios a implementar implica un gran costo inicial que no todas las organizaciones se encuentran dispuestas a enfrentar.

La **estrategia bottom-up** establece una perspectiva diferente durante el análisis. La misma propone comenzar a construir los servicios a partir de requerimientos puntuales, como por ejemplo, establecer canales de integración punto a punto entre sistemas, o reemplazar soluciones de comunicación remota de aplicaciones por un protocolo multiplataforma como SOAP (*Simple Object Access Protocol*). Muchas veces estos requerimientos pueden resolverse simplemente implementando servicios sobre módulos de un sistema ya existente. Las organizaciones podrían ver ventajoso a este modelo ya que les permite integrar sus sistemas utilizando nuevas tecnologías a bajo costo. A pesar de que las implementaciones de este tipo podrían resultar exitosas, es decir, lograr su objetivo de integración puntual, no se encontrarían enmarcadas en una arquitectura diseñada para aprovechar la orientación a servicios en su máxima expresión. Las soluciones desarrolladas bajo este modelo no están concebidas para soportar un gran número de servicios de forma consistente, robusta o ágil.

La **estrategia ágil** permite incorporar los principios de arquitectura orientada a servicios en los ambientes de negocio, sin necesidad de esperar que se haya finalizado el proceso en toda la organización. La modalidad de trabajo de este modelo difiere de las anteriores ampliamente ya que se ocupa de ejecutar el análisis del negocio en paralelo al diseño de servicios y desarrollo. Esta forma de trabajo tiene una

componente de esfuerzo adicional, con el lógico costo asociado. Esto se debe a la necesidad de tener que ajustar los servicios construidos para alinearlos con los modelos de negocio que pueden ir cambiando a medida que se avanza con el análisis.

1.2.3 Componentes fundamentales en una arquitectura orientada a servicios

La siguiente figura muestra los componentes fundamentales de una arquitectura orientada a servicios (Figura 5).

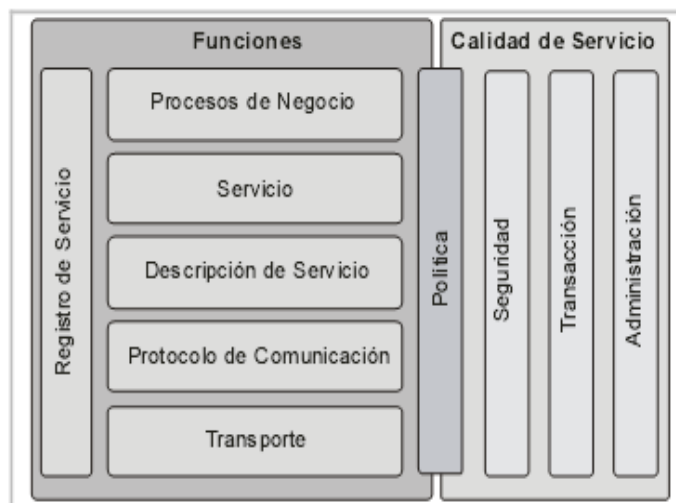


Figura 5. Componentes de una arquitectura orientada a servicios

Como se puede observar, se diferencian dos zonas, una que abarca los aspectos funcionales de la arquitectura y otra que abarca aspectos de calidad de los servicios.

A continuación se describen los elementos brevemente:

- **Funciones**
 - **Transporte:** mecanismo utilizado para llevar las demandas de servicio desde un consumidor de servicio hacia un proveedor de servicio, y las respuestas desde el proveedor hacia el consumidor.
 - **Protocolo de comunicación de servicios:** mecanismo acordado a través del cual un proveedor de servicios y un consumidor de servicios comunican qué está siendo solicitado y qué está siendo respondido.

- Descripción de servicio: esquema acordado para describir qué es el servicio, cómo debe invocarse, y qué datos requiere el servicio para invocarse con éxito.
- Servicios: describe un servicio actual que está disponible para utilizar.
- Procesos de Negocio: colección de servicios, invocados en una secuencia particular con un conjunto particular de reglas, para satisfacer un requerimiento de negocio.
- Registro de Servicios: repositorio de descripciones de servicios y datos que pueden utilizar proveedores de servicios para publicar sus servicios, así como consumidores de servicios para descubrir o hallar servicios disponibles.
- Calidad de servicio
 - Política: condiciones o reglas bajo las cuales un proveedor de servicio hace el servicio disponible para consumidores.
 - Seguridad: reglas que pueden aplicarse para la identificación, autorización y control de acceso a consumidores de servicios.
 - Transacciones: atributos que podrían aplicarse a un grupo de servicios para entregar un resultado consistente.
 - Administración: de atributos que podrían aplicarse para manejar los servicios proporcionados o consumidos.

Una descripción mas detalla puede encontrarse en (Cubillos, 2004).

1.2.4 Entidades en una arquitectura orientada a servicios

Las colaboraciones en SOA siguen el paradigma *find, bind and invoke*, donde un consumidor de servicios realiza la localización dinámica de un servicio consultando el registro de servicios para hallar uno que cumpla con un determinado criterio. Si el servicio existe, el registro proporciona al consumidor la interfaz de contrato y la dirección del servicio proveedor.

El siguiente diagrama (Figura 6) ilustra las entidades (*roles, operaciones y artefactos*) en una arquitectura orientada a servicios donde estas colaboran.

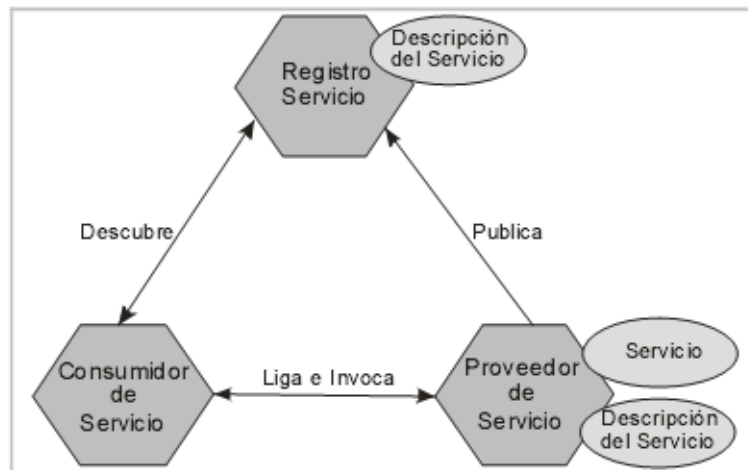


Figura 6. Colaboraciones en SOA

Cada entidad puede tomar el rol de consumidor, proveedor y/o registro:

- Un **consumidor** de servicios es una aplicación, un módulo de software u otro servicio que requiere un servicio, y ejecuta el servicio de acuerdo a un contrato de interfaz.
- Un **proveedor** de servicios es una entidad direccionable a través de la red que acepta y ejecuta consultas de consumidores, y publica sus servicios y su contrato de interfaces en el registro de servicios para que el consumidor de servicios pueda descubrir y acceder al servicio.
- Un **registro** de servicios es el encargado de hacer posible el descubrimiento de servicios, conteniendo un repositorio de servicios disponibles y permitiendo visualizar las interfaces de los proveedores de servicios a los consumidores interesados.

Las operaciones son:

- **Publicar.** para poder acceder a un servicio se debe publicar su descripción para que un consumidor pueda descubrirlo e invocarlo.
- **Descubrir.** un consumidor de servicios localiza un servicio que cumpla con un cierto criterio consultando el registro de servicios.
- **Ligar e Invocar.** una vez obtenida la descripción de un servicio por parte de un consumidor, éste lo invoca de acuerdo a la información en la descripción del servicio.

Finalmente, los artefactos en una arquitectura orientada a servicios son:

- Servicio. un servicio que está disponible para el uso a través de una interfaz publicada y que permite ser invocado por un consumidor de servicios.
- Descripción de servicio. especifica la forma en que un consumidor de servicio interactuará con el proveedor de servicio, especificando el formato de consultas y respuestas desde el servicio. Esta descripción también puede especificar el conjunto de pre-condiciones, pos-condiciones y/o niveles de calidad de servicio.

1.2.5 Relación entre servicios y comunicación

Los servicios pueden ser usados por otros servicios u otros programas. Para poder interactuar, los servicios deben estar al tanto de la existencia de cada uno. Esto se logra a través del uso de descripciones de servicios cumpliendo con la siguiente arquitectura de servicios como aparece en la Figura 7:

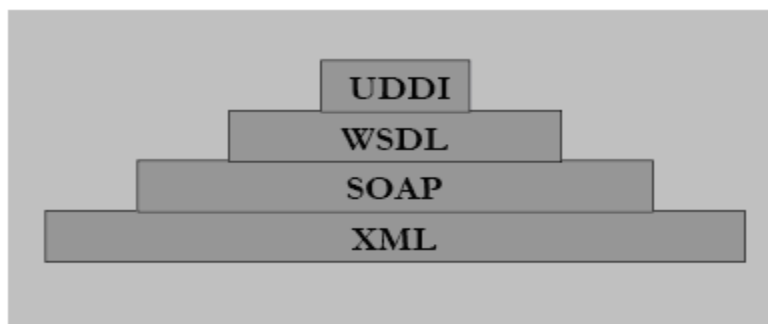


Figura 7. Arquitectura de desarrollo de servicios

XML (*Extensible Markup Language*): Es el formato estándar para los datos que se vayan a intercambiar.

SOAP (*Simple Object Access Protocol*): Es el protocolo sobre el que se establece el intercambio.

WSDL (*Web Services Description Languages*): Es el lenguaje de la interfaz pública para los servicios web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación.

UDDI (*Universal Description, Discovery and Integration*): Es el protocolo para publicar la información de los servicios web. Permite comprobar qué servicios web están disponibles.

Una descripción de servicio establece, en su forma básica, el nombre del servicio y los datos esperados y retornados por el servicio.

WSDL satisface esta necesidad al definir una gramática XML para definir servicios de red como colecciones de agentes de comunicación capaces de intercambiar mensajes.

El formato general de un archivo WSDL tiene la estructura que muestra la Figura 8:

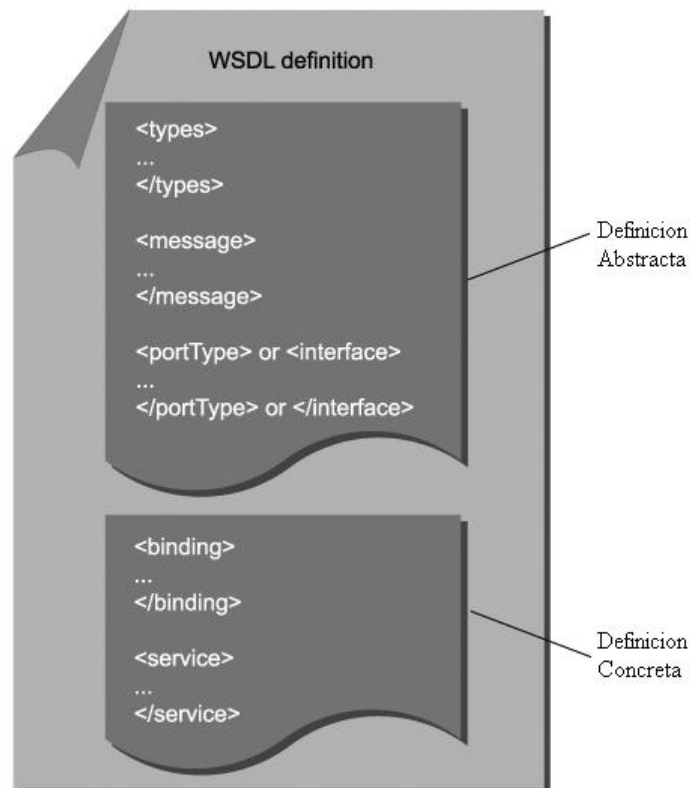


Figura 8. Estructura de archivo WSDL

La forma en que los servicios usan las descripciones de servicios resulta una relación desacoplada. Por ejemplo un servicio **Uno** conoce al servicio **Dos** porque el servicio **Uno** posee la descripción del servicio **Dos**. De esta forma el servicio **Uno** tiene toda

la información necesaria para comunicarse con el servicio **Dos**. Esta relación está reflejada en la Figura 9.



Figura 9. Relación entre Servicios

Luego de que un servicio envía un mensaje, pierde el control de lo que le sucede al mensaje. Es por esto que los mensajes deben ser autónomos e incluir suficiente inteligencia de forma tal de poder manejar su lógica interna de procesamiento ver Figura 10.

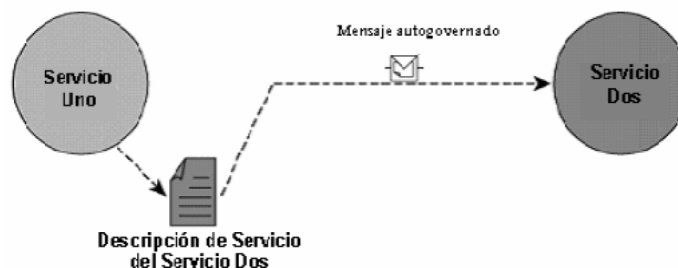


Figura 10. Mensajes Autónomos o Auto gobernados

1.2.6 Principios básicos para el diseño de los servicios

Para definir una arquitectura orientada a servicios apropiada, es necesario apoyarse en un conjunto de principios:

- **Desacoplamiento:** los servicios mantienen una relación que minimiza las dependencias y sólo se quiere que unos conozcan la existencia de los otros.
- **Contrato de servicio:** los servicios se adhieren a un acuerdo de comunicación el cual está definido por las descripciones de servicios.

- **Autonomía:** los servicios tienen el control sobre la lógica que ellos encapsulan.
- **Abstracción:** los servicios ocultan su lógica de procesamiento. Sólo especifican lo descrito en el contrato de servicio.
- **Reutilización:** la lógica se encuentra dividida en servicios con la intención de hacerla reutilizable.
- **Composición:** una colección de servicios puede ser coordinada y unida de forma tal de componer servicios.
- **Ausencia de estado:** los servicios minimizan la información retenida sobre una tarea específica.
- **Descubrimiento:** los servicios se diseñan de forma tal de poder ser descritos con el objeto de poder ser encontrados y evaluados a través de mecanismos de descubrimiento.

1.2.7 Capas principales en una arquitectura orientada a servicios

Además de seguir una estrategia para la definición de la arquitectura orientada a servicios como se explicó en epígrafes anteriores, es importante también definir las capas básicas de la arquitectura que permitan la conexión de los servicios que se diseñen con los procesos definidos en el negocio.

Existen tres capas básicas de servicios (Figura 11):

- La **capa de servicios de aplicación** establece los diferentes niveles que existen para expresar el funcionamiento de una tecnología específica. Los servicios que residen dentro de esta capa pueden ser referidos como los servicios de aplicación. Su propósito es proporcionar funciones reusables relacionadas con el procesamiento de los datos dentro del ambiente de aplicación.
- La **capa de servicios de negocio** introduce un servicio que se interesa solamente con representar la lógica de negocio, nombrado servicio de negocio. Incluye la creación de dos modelos de servicio de negocio

adicionales: servicios de negocio centrado en tareas y servicios de negocio centrado en entidades.

- La **capa de servicios de orquestación** es la de mayor nivel de abstracción y se encarga de manejar los detalles de la interacción entre los demás servicios exigiendo y garantizando que el funcionamiento se ejecute en una sucesión específica. Dentro de la capa de servicios de orquestación, los servicios del proceso componen otros servicios que proporcionan juegos específicos de funciones, independientes de las reglas de negocio.

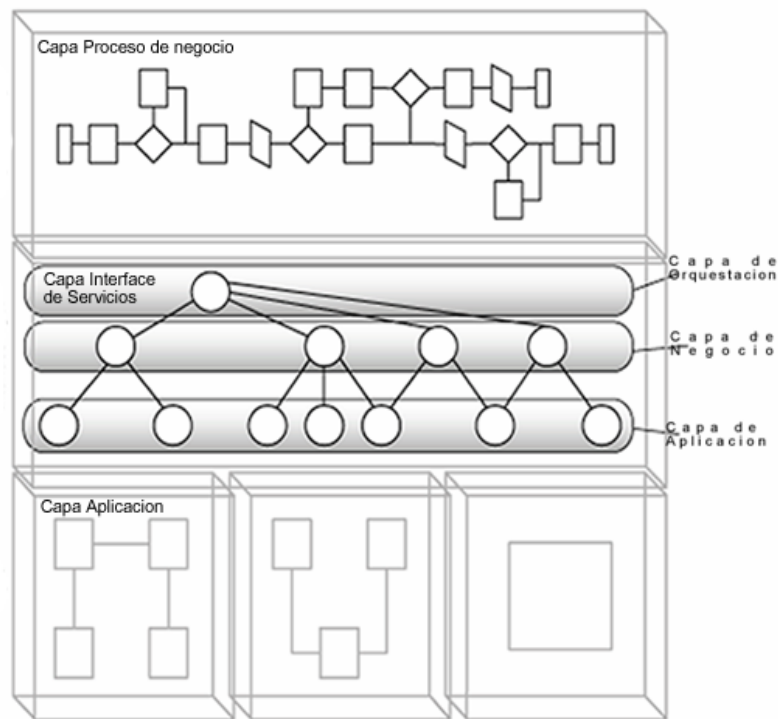


Figura 11. Capas básicas de servicios

1.2.8 Diseño Orientado a Servicios

El diseño orientado a servicios es el proceso en el cual se definen las relaciones y la lógica de representación de los servicios web, a partir de los servicios candidatos creados en el proceso de análisis orientado a servicios y de la forma en que se encapsulan los servicios para poner en marcha el proceso de negocio.

El proceso de diseño de servicios comienza con la definición de las tecnologías que establecen un entorno de desarrollo para la arquitectura orientada a servicios SOA, que abarca desde la definición de las capas de servicios hasta la elección de la extensión SOA a utilizar.

El protocolo SOA está basado en XML para el intercambio de información y tiene la siguiente estructura Figura 12:

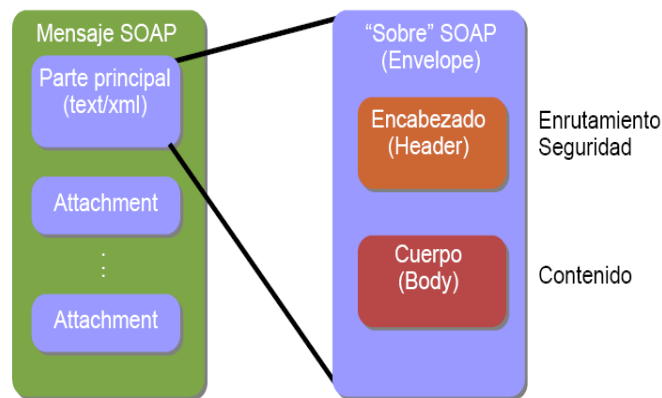


Figura 12. Estructura mensaje SOAP

Luego, el diseño de servicios basado en las tres capas de servicio incluye la identificación de los tres tipos principales de servicios:

Servicios de negocio centrados en entidades: representan la capa de servicios menos influenciada por las demás capas. El propósito es representar exactamente la correspondencia existente entre los datos de las entidades y el propio modelo de negocio. Estos servicios son una solución para lograr que distintas aplicaciones puedan acceder o administrar información particular de cierta entidad.

Servicios de aplicación: representan la parte baja de las capas de servicio, responsable de poner en práctica cualquier demanda de proceso dictada por la capa de negocio o la capa de orquestación.

Servicios de negocio centrados en tareas: el diseño de este tipo de servicios por lo general requiere menos esfuerzo que el diseño de los dos anteriores, ya que el reuso

no es una consideración importante. Sólo las operaciones de los servicios candidatos es lo que se tiene en cuenta.

El diseño orientado a servicios concluye con el diseño del proceso de negocio orientado a servicios (Erl, 2005).

1.3 Tecnologías asociadas a la coordinación de servicios web

La coordinación de servicios web para la implementación de procesos de alto nivel, requiere de diversos estándares que permitan modelar las posibles interacciones entre los servicios. Se introducen de esta forma los términos de orquestación y coreografía, que tratan de describir aspectos relacionados con la creación de modelos de negocio que involucren servicios web.

1. Orquestación

Un proceso se puede considerar una orquestación de servicios cuando es controlado totalmente por una única entidad. Este proceso define completamente las interacciones con los servicios componentes y la lógica requerida para conducir correctamente esas interacciones. Este tipo de proceso puede entenderse como privado y ejecutable ya que sólo la entidad que está orquestando el proceso conoce el flujo de control e información que sigue el proceso que se está orquestando. De esta manera se crea un proceso que puede utilizar diferentes servicios manipulando la información que fluye entre ellos, convirtiendo, por ejemplo, los datos de salida de algunos servicios en datos de entrada de otro. Aquí, cada entidad que participa implementa y controla su propio proceso.

2. Coreografía

Un proceso es una coreografía de servicios cuando define las colaboraciones entre cualquier tipo de aplicaciones componentes, independientemente del lenguaje o plataforma en el que estén definidas las mismas. Un proceso de coreografía no es controlado por uno solo de los participantes. A diferencia de la orquestación, la coreografía puede verse como un proceso público y no ejecutable. Público porque define un comportamiento común que todas las entidades participantes deben

conocer, y no ejecutable porque está pensado para verse más bien como un protocolo de negocio que dicta las reglas para que dichas entidades puedan interactuar entre sí.

1.3.1 Lenguajes y Estándares

Existen actualmente lenguajes y estándares específicos para la coordinación de servicios web. Aquí se presentarán los más conocidos, y principalmente aquellos que permiten especificar la orquestación de servicios.

✓ XLANG

XLANG es un lenguaje de orquestación implementado por Microsoft, con el fin de soportar los flujos de procesos secuenciales, paralelos y condicionales de su servidor BizTalk. Es un lenguaje propietario de procesos de negocio, estructurado en bloques con estructuras de flujo de control, que incluye mecanismos para la captura de excepciones y soporte de transacciones de larga duración. En particular, los procesos dinámicos de negocio son compuestos como un flujo de control que invoca las interfaces WSDL de los participantes involucrados (ebPML.org, 2009).

✓ WSFL

Web Service Flow Language (WSFL) es un lenguaje de orquestación, desarrollado por IBM. Diseñado sobre WSDL, su objetivo es la colaboración externa entre participantes del proceso de negocio, como también los procesos internos. En contraste con XLANG, WSFL no está limitado a estructuras de bloque y permite grafos direccionados. La composición de flujo de servicios es implementada permitiendo especificaciones recursivas del control y flujo de datos entre los servicios web (Pages, 2001).

✓ BPEL

Business Process Execution Language for Web Services (BPEL4WS, BPELWS o comúnmente BPEL), define una notación estándar para especificar el comportamiento de un proceso de negocio basándose en servicios web. Posibilita una

mezcla de modelos de proceso de estructuras de bloque y de grafo, haciendo el lenguaje expresivo al costo de ser complejo (IBM, 2002).

✓ YAWL

Yet Another Workflow Language (YAWL) es un sistema de procesamiento de Workflow. Este lenguaje toma como punto de partida las redes de Petri y adiciona mecanismos para permitir un soporte más directo e intuitivo de los patrones de workflow (SourceForge.Net, 2009). La motivación que se plantea para utilizar las redes de Petri es que proporciona semántica formal, se basa en estados en lugar de eventos, y abundan técnicas de análisis. Por otro lado, mejora la transformación de datos complejos, y la integración de servicios web. Usa Java, Schema, Xpath Query, SOAP y WSDC.

✓ WSCI

Web Service Choreography Interface (WSCI) es un lenguaje de descripción de interfaces basado en XML, que describe el flujo de mensajes intercambiados por un servicio web que participa de una coreografía de servicios. WSCI trabaja en conjunto con WSDL y también con cualquier lenguaje de definición de servicios que exhiba las características de WSDL. WSCI describe el comportamiento observable de un servicio web, expresado en términos de dependencias lógicas y temporales entre los mensajes intercambiados, ofreciendo reglas de secuencias, correlación, manejo de excepciones y transacciones. WSCI también describe el intercambio colectivo de mensajes entre los servicios web que interactúan, brindando así una vista global de las interacciones orientadas a mensajes (W3C, 2002).

✓ WS-CDL

WS-CDL (*Web Services Choreography Description Language*) surge como la continuación del WSCI. Es un lenguaje que describe el comportamiento de participantes desde una perspectiva de modelo global (W3C, 2003). Estos participantes pueden ser servicios web implementados en base a lenguajes ejecutables de procesos de negocio, servicios web implementados en lenguajes como Java, o

agentes de software controlados por humanos. WS-CDL no es un lenguaje ejecutable de descripción de procesos de negocio.

✓ ebXML

ebXML fue publicado en 1999 como una iniciativa del UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) y OASIS (Organization for the Advancement of Structured Information Standards). Es una familia basada en el estándar XML auspiciado por OASIS y UN/CEFACT para los cuales la misión es proveer una infraestructura abierta basada en XML que permita el uso global de información para negocios electrónicos en una forma interoperable, segura y consistente para todos los socios del negocio. ebXML no es un estándar, es un contenedor para varios estándares de especificación administrados por UN/CEFACT y OASIS: ebXML Servicios de Mensajes, ebXML Registro, ebXML Esquema de Especificación de Procesos de Negocios y ebXML Acuerdo y Perfil de Protocolos de Colaboración, los cuales fueron aprobados por la ISO (*International Organization for Standardization*) y son parte de la familia de estándares ISO 15000 (ebXML, 2006).

1.4 Consideraciones Finales del Capítulo

Las tecnologías BPM han surgido y se desarrollan, apoyadas en un concepto de reciente aparición, la arquitectura orientada a servicios, que proporciona un marco adecuado para solucionar el problema de distribución y heterogeneidad de sistemas con los que se debe interactuar para completar el proceso de negocio.

La propuesta de una arquitectura orientada a servicios, además de que debe cumplir con los principios básicos de diseño de los mismos, debe seguir una estrategia de elaboración. En este punto, aquella que establece un desarrollo de arriba hacia abajo, proporciona una idea adecuada para el desarrollo de la arquitectura en negocios como el de la revista arbitrada, donde la complejidad de los procesos no es elevada y los mismos están bien definidos.

Aún cuando se reportan múltiples herramientas que solucionan el problema de la coordinación de los servicios web, el empleo de herramientas libres como es el caso

de YAWL, es viable para la solución del problema que se plantea en la introducción de este trabajo. Esta elección no obedece solamente a la posibilidad de extensión de la misma, sino también a los consolidados mecanismos de soporte de patrones y validación semántica que presenta, lo que posibilita una modelación segura de los procesos del negocio.

CAPITULO II: MODELADO DEL NEGOCIO

El proceso de aceptación y publicación de artículos en las revistas arbitradas puede gestionarse de manera automatizada combinando técnicas BPM y SOA de manera adecuada. Una de las primeras etapas en la definición de la arquitectura orientada a servicios es la de análisis, en la que un eslabón fundamental, lo constituye el modelado de procesos de negocio. El presente capítulo aborda estas ideas desde la organización del trabajo editorial hasta la definición de los procesos de negocio fundamentales y su especificación y modelación.

2.1 Lineamientos generales para el descubrimiento y diseño de los procesos de negocio

Para conseguir sus objetivos, una revista organiza su actividad por medio de un conjunto de procesos de negocio. Cada uno de ellos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (por ejemplo, trabajadores o departamentos) participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan las políticas y la estructura de la información de la revista. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio.

- **Tareas:** cada tarea es un conjunto de acciones o actividades manejadas como una sola unidad. Generalmente son desempeñadas por una única persona dentro de los roles que pueden realizar dicha tarea. Las tareas surgen del análisis del flujo del trabajo, donde se define quienes la ejecutan.
- **Usuarios:** las tareas son realizadas en un orden definido por determinadas personas (o agentes automatizados tomando el rol de las personas) basados en las condiciones o reglas del negocio.
- **Roles:** cada rol determina las distintas competencias potenciales que existen en el sistema. Se definen independientemente de las personas físicas a las

cuales se les van a asignar dichos roles. Una persona puede tener más de un rol.

- Reglas de transición: son reglas lógicas que determinan la navegación del documento dentro del sistema. Expresan que acción se va a tomar dependiendo del valor de expresiones lógicas. La definición de las reglas puede ser muy complicada, con múltiples opciones, variaciones, y excepciones.

Pueden identificarse dos momentos fundamentales en el modelado del negocio: capturar los procesos de negocio de la organización bajo estudio y determinar los agentes involucrados en su realización.

En primer lugar, se pueden considerar los objetivos estratégicos de la organización. Teniendo en cuenta que estos objetivos van a ser muy complejos y de un nivel de abstracción alto, serán descompuestos en un conjunto de subobjetivos más concretos, que deberán cumplirse para conseguir el objetivo estratégico. Estos subobjetivos pueden a su vez ser descompuestos en otros, de manera que se defina una jerarquía de objetivos. En este estudio, se ha experimentado que dos o tres niveles de descomposición son suficientes.

Una vez que se han identificado los procesos de negocio, es preciso encontrar los agentes involucrados en su realización. Cada uno de estos agentes o actores del negocio desempeña cierto papel (juega un rol) cuando colabora con otros para llevar a cabo las actividades que conforman el proceso del negocio.

2.2 Organización del trabajo editorial en una revista arbitrada

El trabajo editorial en una revista arbitrada puede desarrollarse, de manera general, por un conjunto de personas que se encuentran involucradas en diferentes procesos de negocio.

2.2.1 Personal involucrado en el trabajo editorial

La revisión y aprobación de artículos en la revista está presidida por un director y es apoyada por un consejo editorial. Además, la revisión de los artículos es desarrollada por un grupo de árbitros.

El director representa oficial y legalmente a la publicación, dirige orienta y controla el trabajo en general de la publicación y propone acciones para el mejoramiento de la misma.

El consejo editorial es el órgano encargado de llevar a cabo el trabajo editorial de la revista. Está compuesto por el director, el editor jefe, el editor ejecutivo, los editores y el asistente de edición.

Las funciones del comité editorial son las siguientes:

- Definir la política científica y editorial, así como velar por su cumplimiento.
- Establecer las normas editoriales de la publicación, incluidos los preceptos éticos del material publicado, así como velar por su cumplimiento.
- Definir la política de distribución y comercialización de la revista.

Además del director de la revista, cuyas funciones se explicaron anteriormente, los demás miembros del consejo editorial tienen determinadas responsabilidades y desarrollan un grupo de funciones. De esta manera:

Editor Ejecutivo

Asiste al director en el manejo diario de la publicación ya que interactúa con los editores, árbitros y autores, para garantizar el flujo continuo del proceso editorial.

Las funciones del editor ejecutivo son las siguientes:

- Velar por el estricto cumplimiento de la política trazada por el consejo.
- Coordinar y controlar el trabajo del consejo editor.
- Orientar los trabajos aceptados hacia el editor, según la temática.

- Garantizar la información necesaria al equipo de trabajo y los árbitros para que puedan realizar con calidad sus funciones.
- Proponer las medidas y acciones necesarias para el desarrollo de la publicación.

Editor Jefe

Colabora directamente con el editor ejecutivo, forma parte del consejo editorial. Interactúa con los editores para garantizar una adecuada clasificación temática, además de participar directamente en el análisis de la No aceptación de un artículo.

Entre sus labores esta:

- Velar por una correcta clasificación temática.
- Enviar notificación al autor de un artículo en el caso de la no aceptación.

Editores

Son miembros del consejo editor y su número debe cubrir las diferentes especialidades dentro de la(s) temática(s) que cubre la revista.

Entre sus tareas se destacan:

- Participar en la dirección colegiada de la revista.
- Aprobar y hacer cumplir la política editorial de la revista.
- Velar por lo que se publique en la temática que es responsable.
- Son los responsables de la calidad de los artículos que se publican, y se apoyan en el sistema de arbitraje. Deben proponer a los árbitros de cada artículo y sugerir un dictamen para cada uno de ellos, según los informes de arbitraje recibidos.
- Considerar, con énfasis particular el cumplimiento de las normas éticas, los trabajos sometidos a su consideración.
- Editar y gestionar la publicación de artículos de revisión.

- Proponer al consejo editor, sobre la base de su merito científico, la información no arbitrada que se incluirá en la publicación.

Asistente de Edición

Colabora con el director en la conducción diaria de la publicación.

Entre sus funciones están:

- Ejecutar y controlar el flujo de los manuscritos.
- Conocer las normas de estilo científico y asesorar al consejo editor acerca de su cumplimiento.
- Tramitar y controlar todo el flujo de trabajo de la revista.
- Asistir al editor ejecutivo en sus diferentes tareas.

Por otra parte:

Árbitros

Tienen la función de revisar y evaluar los artículos para impresión.

Como reglas deben cumplir:

- Los árbitros no deben conocer la identidad de los demás árbitros.
- No deben conocer el nombre de los autores.
- Revisar integralmente (contenido y forma, incluída la redacción, palabras claves, estructura de los resúmenes, adecuación del lenguaje utilizado, etc.) los artículos sometidos a su consideración y proponer, mediante un informe de arbitraje por escrito, las medidas y modificaciones que se entiendan necesarias, de acuerdo con la política editorial, normas y requisitos de la revista.

2.2.2 Procesos de negocio fundamentales en el trabajo editorial

Pueden definirse a partir de un estudio exhaustivo tres procesos de negocio fundamentales en la revista:

- Aceptacion Preliminar de los Artículos.
- Proceso de Arbitraje.
- Confección de la revista.

Aceptación Preliminar de los Artículos:

Proceso de negocio que incluye varios subprocesos en los que participan el editor jefe, editor ejecutivo y el asistente de edición. Inicia con un subproceso en el cual se verifican y se hacen cumplir las normas editoriales, seguido de otros dos subprocesos en los cuales se elige un editor especialista y se clasifica inicialmente la temática del artículo; estos dos subprocesos culminan con la aceptación de la temática propuesta por el autor y establecida como política de la revista.

Proceso de arbitraje:

Incluye entre sus actores principales a los árbitros, al editor jefe, editor ejecutivo y al editor especialista. Proceso de negocio que comienza con una designación de árbitros por parte del editor ejecutivo y el editor jefe según las temáticas en las que están agrupados los árbitros ya que su elección debe ser aleatoria; le sigue un proceso de negocio que se encarga del arbitraje propiamente dicho, en el que sólo intervienen los árbitros y que incluye otros subprocesos, seguido por un subproceso en el que se evalúan los artículos dando paso a otro subproceso en que se notifica al autor por las condiciones de su trabajo y se envía el artículo para impresión.

Confección de la Revista:

Proceso en el que participan los editores y el consejo editorial, inicia con la confección de la propuesta del número de la revista por un grupo de editores, seguido por la aprobación del número de la revista por parte del consejo editorial.

2.3 Modelación de los procesos de negocio

A continuación se presenta una propuesta general de modelado de las principales tareas involucradas en el proceso de edición de una revista arbitrada y se especifica cada uno de los modelos.

Se realizó la modelación utilizando la herramienta YAWL (*Yet Another Workflow Language*), software para modelar flujos de trabajo, con fuerte argumento matemático basado en Redes de Petri, lo cual permite obtener una sólida validación del modelo.

2.3.1 Modelo general de procesos del negocio

El modelado del negocio Aceptación de los Artículos, puede organizarse sobre la base de cuatro procesos fundamentales. Comienza con la Aceptación Preliminar de los Artículos. Este proceso se bifurca hacia un Análisis de la No Aceptación o al Proceso de Arbitraje en caso de que el artículo sea inicialmente aceptado, seguido por el proceso Confección de la Revista, para de esta forma poner fin al proceso de negocio (Figura 13).

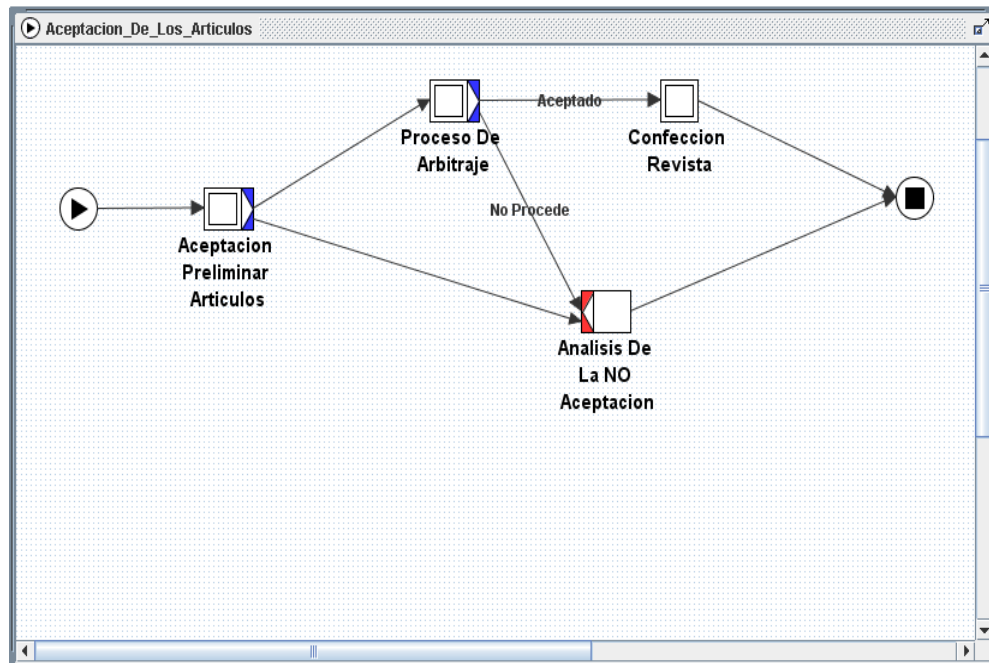


Figura 13. Proceso de Negocio Aceptación de los Artículos

Los principales roles a asignar son:

- Arbitro 1
- Arbitro 2
- Arbitro 3

- Asistente de Edición
- Autor Artículo
- Consejo Editorial
- Editor Ejecutivo
- Editor Especializado
- Editor Jefe
- Editores

A los cuales se les pueden asignar los siguientes privilegios:

- Un rol puede suspender el comienzo de una tarea
- Un rol puede reasignar el trabajo de una tarea a otro rol que puede reiniciar el estado de la tarea
- Un rol puede reasignar el trabajo de una tarea a otro rol que retiene el estado
- Un rol puede desasignarse a un trabajo de una tarea
- Un rol puede delegar su trabajo de esta tarea en otro rol
- Un rol puede saltar un trabajo de esta tarea

Nótese que en el proceso mostrado anteriormente se hace uso de los siguientes patrones de flujo de trabajo. Para más detalle en la notación, ver Anexo I:

- Patrones de control básico
- Patrones de bifurcación y sincronismo avanzados
- Patrones que involucran múltiples instancias
- Patrones estructurales
- Patrones basados en el estado

2.3.2 Especificación de los procesos generales del negocio

La especificación de los procesos de negocio (Tablas 1, 2 y 3), así como de cada uno de los subprocesos que componen los mismos, se realiza en lo adelante, atendiendo a un conjunto de características que proporciona la herramienta sobre la que se realiza el modelado, además de presentarse una descripción del proceso Análisis de la No Aceptación en la Tabla 4:

- Nombre y clasificación del proceso
- Descripción del proceso
- Roles que intervienen en el proceso
- Variables de entrada al proceso
- Variables de salida del proceso
- Flujo hacia donde se dirige el proceso
- Reglas de transición que dictan la manera en que de un proceso se pasa a otro proceso.

Tabla 1: Proceso de negocio Aceptación Preliminar de Artículos

<i>Múltiples Tareas</i>	<i>Aceptación Preliminar Artículos</i>
<i>Descripción</i>	<i>Proceso de negocio que inicia con el cumplimiento de las normas editoriales y culmina con la aceptación de la temática inicial.</i>
<i>Roles</i>	<i>Editor Jefe, Editor Ejecutivo y Editor Especializado</i>
<i>Variables Entrada</i>	<i>Datos Autor, Observaciones</i>
<i>Variables Salida</i>	<i>Artículo Arbitraje</i>
<i>Flujo</i>	<i>Proceso Arbitraje</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 2: Proceso de negocio Arbitraje

<i>Múltiples Tareas</i>	<i>Proceso Arbitraje</i>
<i>Descripción</i>	<i>Proceso que se inicia con la aceptación preliminar de los artículos y culmina con un artículo listo para publicar en la revista.</i>
<i>Roles</i>	<i>Editor Jefe, Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Artículo Arbitraje</i>
<i>Variables Salida</i>	<i>Evaluación, Artículo Listo</i>
<i>Flujo</i>	<i>Proceso Arbitraje</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 3: Proceso de negocio Confección de la Revista

<i>Múltiples Tareas</i>	<i>Confección Revista</i>
<i>Descripción</i>	<i>Comienza al finalizar el proceso de arbitraje y culmina con la aprobación del número de la revista.</i>
<i>Roles</i>	<i>Editor Jefe, Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Evaluación, Artículo Arbitrado</i>
<i>Variables Salida</i>	<i>Artículo</i>
<i>Flujo</i>	<i>Fin del Proceso</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 4: Proceso de negocio Análisis de la No Aceptación

<i>Tarea</i>	<i>Análisis de la No Aceptación</i>
<i>Descripción</i>	<i>En este proceso se analizan las causas por las que no ha sido aceptado el artículo y se notifica al autor. El artículo pasa por un proceso de revisión que culmina con una notificación y recomendaciones al autor.</i>

<i>Roles</i>	<i>Editor Jefe y Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Evaluación, Datos Autor</i>
<i>Variables Salida</i>	<i>Notificación</i>
<i>Flujo</i>	<i>Fin del Proceso</i>
<i>Regla de transición</i>	<i>Tiempo</i>

2.3.3 Especificación de los subprocessos del negocio

A continuación se realiza la especificación de los subprocessos que se definieron a partir de los procesos generales descritos en el epígrafe anterior.

2.3.3.1 Modelado del proceso de negocio Aceptación Preliminar de Artículos

Los subprocessos que aparecen al detallar el proceso de negocio Aceptación Preliminar de Artículos se muestran en la Figura 14:

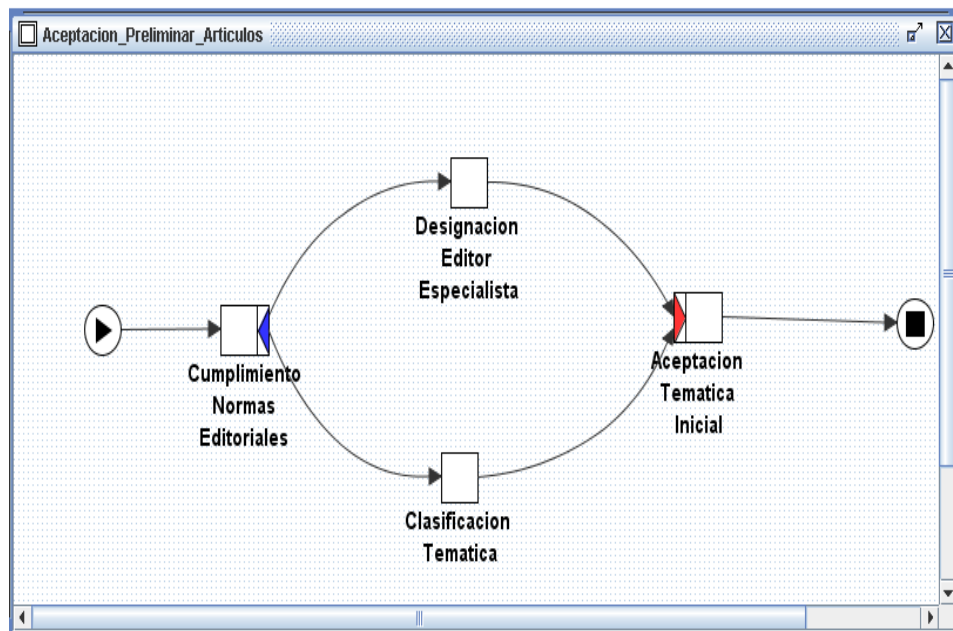


Figura 14. Subproceso Modelado del proceso de Negocio Aceptación Preliminar Artículos

Especificación de los subprocessos de Aceptación Preliminar de Artículos

Las Tablas 5, 6, 7, 8 muestran la especificación de cada uno de los subprocesos de la Figura 14.

Tabla 5: Subproceso Cumplimiento de las Normas Editoriales

<i>Tarea</i>	<i>Cumplimiento de las Normas Editoriales</i>
<i>Descripción</i>	<i>Actividad en la cual el Asistente de Edición garantiza el cumplimiento de un conjunto de reglas y normas que debe seguir el autor para publicar un artículo.</i>
<i>Roles</i>	<i>Asistente de Edición</i>
<i>Variables Entrada</i>	<i>Artículo, Datos Autor, Observaciones</i>
<i>Variables Salida</i>	<i>Artículo, Datos Autor, Observaciones</i>
<i>Flujo</i>	<i>Designación Editor Especialista y Clasificación Temática</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 6: Subproceso Designación del Editor Especialista

<i>Tarea</i>	<i>Designación del Editor Especialista</i>
<i>Descripción</i>	<i>Actividad que garantiza que se asigne al artículo al personal calificado para su revisión con el fin de aceptar una temática inicial.</i>
<i>Roles</i>	<i>Editor Jefe, Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Artículo, Datos Autor</i>
<i>Variables Salida</i>	<i>Artículo, Información</i>
<i>Flujo</i>	<i>Aceptación Temática Inicial</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 7: Subproceso Clasificación Temática

<i>Tarea</i>	<i>Clasificación Temática</i>
--------------	-------------------------------

<i>Descripción</i>	<i>Actividad en la cual el Editor Jefe y Editor Ejecutivo clasifican el artículo al mismo tiempo que el Editor Especialista, para al pasar a la siguiente actividad mostrar una clasificación inicial de la temática.</i>
<i>Roles</i>	<i>Editor Jefe, Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Artículo, Información</i>
<i>Variables Salida</i>	<i>Artículo, Información</i>
<i>Flujo</i>	<i>Aceptación Temática Inicial</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 8: Subproceso Asignación Temática Inicial

<i>Tarea</i>	<i>Aceptación Temática Inicial</i>
<i>Descripción</i>	<i>Se llega a un consenso sobre la temática presentada por el autor y en que sección debería ser incluida.</i>
<i>Roles</i>	<i>Editor Jefe, Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Artículo, Información</i>
<i>Variables Salida</i>	<i>Artículo arbitraje</i>
<i>Flujo</i>	<i>Proceso de Arbitraje</i>
<i>Regla de transición</i>	<i>Tiempo</i>

2.3.3.2 Modelado del proceso de negocio Proceso_de_Arbitraje

Los subprocesos que aparecen al detallar el proceso de negocio Arbitraje se muestran en la Figura 15:

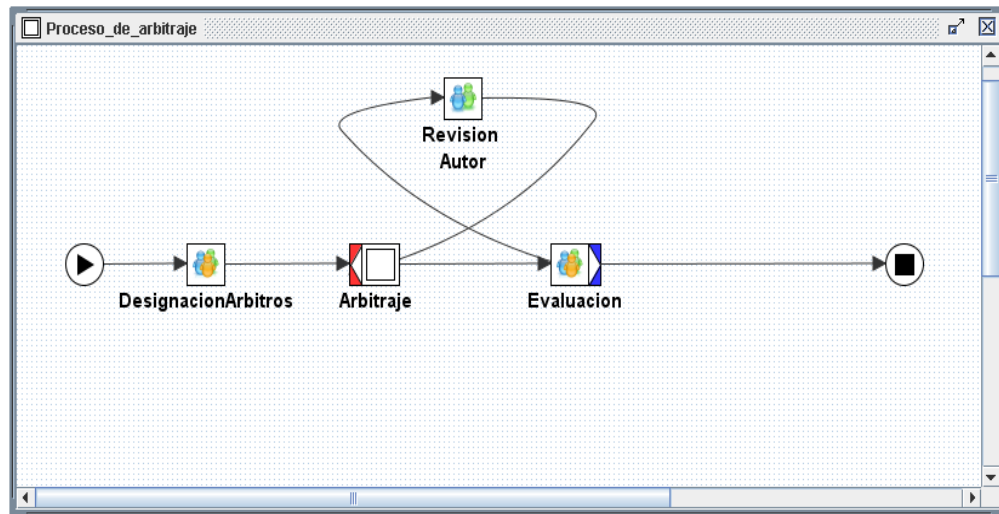


Figura 15. Modelado del proceso de negocio Proceso de Arbitraje

Especificación de los subprocessos del proceso Arbitraje

Las Tablas 9, 10, 11 y 12 muestran la especificación de cada uno de los subprocessos de la Figura 15.

Note que el subprocesso que se describe en la Tabla 10 se declara como “Múltiples tareas” lo que indica que contiene a su vez otros subprocessos que deben ser modelados y especificados.

Para ello se muestra en la Figura 16 la modelación del subprocesso Arbitraje que se especifica en las Tablas 13, 14 y 15.

Tabla 9: Subproceso Designación de Árbitros

<i>Tarea</i>	<i>Designación de Árbitros</i>
<i>Descripción</i>	<i>En esta tarea se asignan a cada artículo tres árbitros como mínimo de un conjunto de árbitros agrupados por temáticas. Los árbitros son asignados aleatoriamente y no deben conocer el nombre del autor.</i>
<i>Roles</i>	<i>Editor Jefe y Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Artículo Arbitraje, Artículo Listo</i>

<i>Variables Salida</i>	<i>Artículo Arbitraje, Artículo Listo</i>
<i>Flujo</i>	<i>Arbitraje</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 10: Subproceso Arbitraje

<i>Múltiples Tareas</i>	<i>Arbitraje</i>
<i>Descripción</i>	<i>Comienza con la designación de los árbitros y culmina con una conciliación entre los mismos.</i>
<i>Roles</i>	<i>Editor Jefe, Editor Ejecutivo, Árbitros</i>
<i>Variables Entrada</i>	<i>Artículo Arbitraje, Artículo Arbitrado</i>
<i>Variables Salida</i>	<i>Artículo Arbitraje, Artículo Arbitrado</i>
<i>Flujo</i>	<i>Evaluación</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 11: Subproceso Evaluación

<i>Tarea</i>	<i>Evaluación</i>
<i>Descripción</i>	<i>Se evalúa cada artículo dada una conciliación antes expresada por los árbitros los cuales revisan el artículo por separado. Se evalúan los artículos en bueno, muy bueno, regular y malo y se decide si proceden o no a la impresión.</i>
<i>Roles</i>	<i>Editor Jefe y Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Artículo Arbitraje</i>
<i>Variables Salida</i>	<i>Evaluación</i>
<i>Flujo</i>	<i>Revisión del Autor, Fin</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 12: Subproceso Revisión del Autor

<i>Tarea</i>	<i>Revisión Del Autor</i>
<i>Descripción</i>	<i>Se le envía al autor el artículo con la evaluación y recomendaciones en el caso de que se necesite modificar el artículo para impresión o se le envían los datos de su artículo impreso o publicado.</i>
<i>Roles</i>	<i>Editor Jefe y Editor Ejecutivo</i>
<i>Variables Entrada</i>	<i>Evaluación, Artículo Arbitrado</i>
<i>Variables Salida</i>	<i>Artículo Arbitrado</i>
<i>Flujo</i>	<i>Arbitraje</i>
<i>Regla de transición</i>	<i>Tiempo</i>

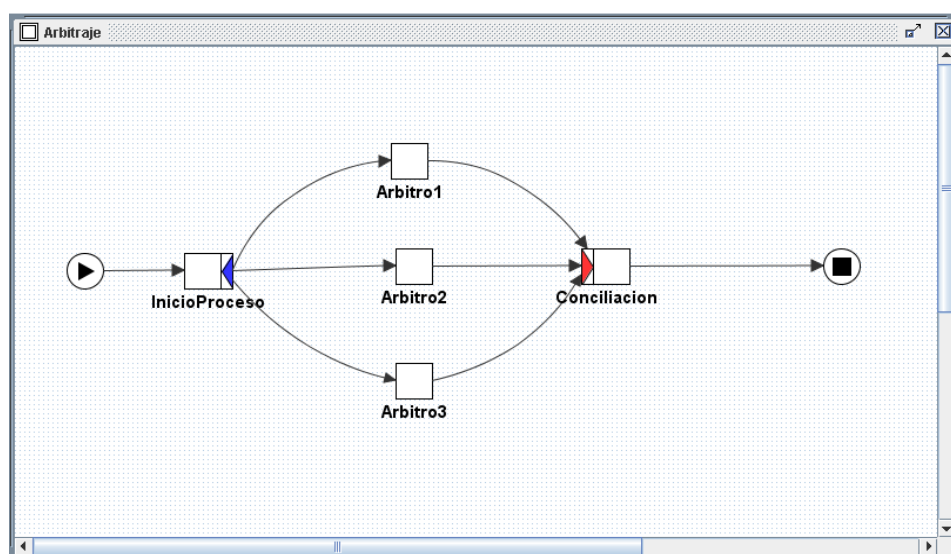


Figura 16. Modelado del subproceso de negocio Arbitraje

Tabla 13: Subproceso Inicio Proceso

<i>Tarea</i>	<i>Inicio Proceso</i>
<i>Descripción</i>	<i>En esta actividad se envía el artículo a árbitros escogidos aleatoriamente.</i>
<i>Roles</i>	<i>Editor Jefe, Editor Ejecutivo</i>

<i>Variables Entrada</i>	<i>Artículo Arbitraje</i>
<i>Variables Salida</i>	<i>Artículo Arbitraje</i>
<i>Flujo</i>	<i>Arbitro 1, Arbitro 2, Arbitro 3</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 14: Subproceso Arbitro 1, Arbitro 2, Arbitro 3

<i>Tarea</i>	<i>Arbitro 1, Arbitro 2, Arbitro 3</i>
<i>Descripción</i>	<i>Este proceso es la revisión del artículo por parte de los árbitros con desconocimiento del autor y de los mismos árbitros.</i>
<i>Roles</i>	<i>Árbitros</i>
<i>Variables Entrada</i>	<i>Artículo Arbitraje</i>
<i>Variables Salida</i>	<i>Artículo Arbitraje</i>
<i>Flujo</i>	<i>Conciliación</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 15: Subproceso Conciliación

<i>Tareas</i>	<i>Conciliación</i>
<i>Descripción</i>	<i>Proceso mediante el cual se llega a una conciliación entre las recomendaciones y cambios propuestos por cada árbitro.</i>
<i>Roles</i>	<i>Editor Especialista</i>
<i>Variables Entrada</i>	<i>Artículo Arbitraje</i>
<i>Variables Salida</i>	<i>Artículo Arbitraje</i>
<i>Flujo</i>	<i>Evaluación</i>
<i>Regla de transición</i>	<i>Tiempo</i>

2.3.3.3 Modelado del proceso de negocio Confección de la Revista

Los subprocessos que aparecen al detallar el proceso de negocio Confección de la Revista se muestran en la Figura 17:

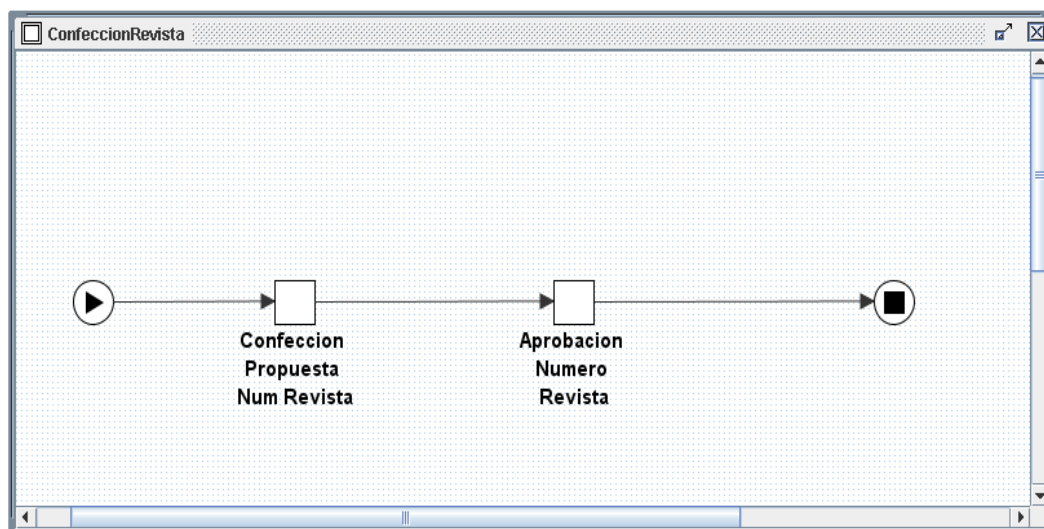


Figura 17. Modelado del proceso de negocio Confección Revista

Especificación de los subprocessos del proceso Confección de la Revista

Las Tablas 16 y 17 muestran la especificación de cada uno de los subprocessos de la Figura 17.

Tabla 16: Subproceso Propuesta Número de Revista

<i>Tareas</i>	<i>Confección Propuesta Número Revista</i>
<i>Descripción</i>	<i>Comienza cuando se tiene ya el artículo arbitrado y se realizan análisis para crear el número de la revista e incluir en el mismo el artículo.</i>
<i>Roles</i>	<i>Editores</i>
<i>Variables Entrada</i>	<i>Artículo Arbitrado</i>
<i>Variables Salida</i>	<i>Artículo Arbitrado</i>
<i>Flujo</i>	<i>Aprobación Número Revista</i>
<i>Regla de transición</i>	<i>Tiempo</i>

Tabla 17: Subproceso Aprobación Número de Revista

<i>Tareas</i>	<i>Aprobación Numero Revista</i>
<i>Descripción</i>	<i>Se aprueba el número de la revista y se le asigna un código al artículo para ser impreso.</i>
<i>Roles</i>	<i>Consejo Editorial</i>
<i>Variables Entrada</i>	<i>Artículo Arbitrado</i>
<i>Variables Salida</i>	<i>Artículo Arbitrado</i>
<i>Flujo</i>	<i>Fin del Proceso</i>
<i>Regla de transición</i>	<i>Tiempo</i>

2.4 Conclusiones del capítulo

El análisis del negocio ha permitido el descubrimiento de los procesos de gestión fundamentales que se manifiestan en una revista arbitrada.

Se ha podido determinar tres procesos a nivel macro, que a su vez pueden desglosarse en subprocesos propios.

Para la modelación se utilizó el YAWL en el que se presentan los patrones de diseño básicos para el modelado de flujo de trabajo. La validación que impone la herramienta dirige la especificación de cada uno de los procesos y subprocesos hacia un conjunto de características entre las que se destacan las variables de entrada y salida de cada proceso y las reglas de transición a ser definidas.

El modelado realizado sirve de base para el capítulo posterior en el que es utilizado para definir los servicios candidatos de las diferentes capas de la arquitectura SOA que se propone en el trabajo.

CAPITULO III: ARQUITECTURA ORIENTADA A SERVICIOS PARA LA GESTION AUTOMATIZADA DEL PROCESO DE EDICION DE UNA REVISTA ARBITRADA.

Las arquitecturas orientadas a servicios permiten la creación de sistemas altamente escalables que reflejan el negocio de la organización. Brindan una forma estándar de exposición e invocación de servicios web, lo cual facilita la interacción entre diferentes sistemas propios o de terceros. En este capítulo se presenta una propuesta de arquitectura orientada a servicios para la gestión automatizada de los procesos de negocio de una revista arbitrada y se desarrolla la descripción de la interfaz de cada uno de los servicios identificados.

3.1 Análisis orientado a servicios

El análisis orientado a servicios forma parte del ciclo de vida de las arquitecturas orientadas a servicios.

En este trabajo, como se señaló en el capítulo anterior se escoge una estrategia de arriba hacia abajo (*top-down*) con un enfoque de tres capas de acuerdo a los referentes teóricos expresados en el Capítulo I. Para el análisis de los servicios se desarrolla una secuencia de pasos según se muestra en el Anexo II.

Paso 1: Definición de los requerimientos del negocio.

Se plantea que el negocio requiere servicios que permitan almacenar la información de los autores de los artículos en una base de datos, que la misma sea interrogada en su momento para obtener los datos del autor, enviar notificaciones a los autores en los distintos momentos que el proceso de negocio lo requiera, así como asignar aleatoriamente árbitros a los artículos.

Paso 2: Identificación de los sistemas automatizados existentes.

En este proceso de negocio es recomendable tener en consideración la existencia de sistemas legados. En el proceso que se analiza, es probable que ya se trabaje con una base de datos que guarde información acerca de los artículos, la cual puede ser accedida desde los servicios web que sean modelados.

Paso 3: Modelado de los servicios

Forma parte del análisis orientado a servicios el modelado de servicios que se encuentra en los límites del análisis con el diseño, el cual es esencial para lograr la organización de la información del negocio.

El modelado de los servicios puede ser visto como una secuencia de 12 pasos que se llevan a la práctica, para mayor detalle puede consultar Anexo III.

El proceso de negocio ha sido modelado en el capítulo anterior utilizando una herramienta que asegura su validación estructural y matemática, logrando representación de la lógica del negocio mediante flujo de trabajo, teniendo en cuenta las reglas y políticas de la organización.

Entre las reglas del negocio hay que tener en cuenta que la secuencia de la lógica debe expresar que:

- Si el documento no cumple con las normas editoriales, se envía al proceso *Aceptación Temática Inicial* con observaciones realizadas por el editor jefe y el editor ejecutivo, para que sea enviado al proceso *Análisis de la No Aceptación*.
- Si el autor en el proceso *Revisión Del Autor*, demora más de 15 días en enviar notificación, enviar al proceso *Evaluación* con una caracterización, con el objetivo de que el artículo no proceda, el cual es enviado al proceso *Análisis de la No Aceptación*.
- Luego del proceso *Aceptación Temática Inicial*, se procede a enviar el artículo para el proceso *Arbitraje*, en el cual por regla del negocio no se debe conocer información referente a los autores.
- En el proceso de *Arbitraje* se le otorga una evaluación al artículo, Regular, Bueno, Muy Bueno, No Procede, en el último caso se le debe enviar

notificación al autor y hacer un análisis de la No Aceptación en un período de 5 días.

Para modelar los servicios se identificaron las operaciones candidatas, se crearon los servicios candidatos y posteriormente se definió la composición de los servicios.

Las operaciones de los posibles servicios candidatos para cada proceso se definen a continuación:

- *Cumplimiento de las Normas Editoriales*

Debe crearse un servicio web que permita obtener y guardar la información de los autores de los artículos, puede ser desarrollado teniendo en cuenta sistemas automatizados existentes.

- *Designación del Editor Especialista*

Actividad manual, realizada por el Editor Jefe y Editor Ejecutivo

- *Clasificación Temática*

Actividad manual, realizada por el Editor Jefe y Editor Ejecutivo

- *Aceptación Temática Inicial*

Actividad manual, realizada por el Editor Jefe y Editor Ejecutivo

- *Designación de Árbitros*

Debe crearse un servicio web que muestre los árbitros que se encuentran disponibles según la temática, los mismos son asignados por el Editor Jefe y Editor Ejecutivo

- *Evaluación*

Debe crearse un servicio que permita enviar notificación al autor, así como su evaluación y observaciones, actividad realizada por el Editor Jefe y Editor Ejecutivo

- *Revisión Del Autor*

Debe crearse un servicio que le permita al autor enviar su artículo nuevamente con los cambios propuestos por los mismos

- *Inicio Proceso*

Actividad manual, realizada por el Editor Jefe y Editor Ejecutivo

- *Arbitro 1, Arbitro 2, Arbitro 3*

Actividad manual, realizada por los árbitros

- *Conciliación*

Actividad manual, realizada por el Editor Especialista

- *Confección Propuesta Numero Revista*

Actividad manual, realizada por los editores

- *Aprobación Numero Revista*

Debe crearse un servicio web que, de parte del consejo editorial, de a conocer al autor la publicación del artículo en la revista

Luego de realizar un análisis de las operaciones propuestas anteriormente, planteándose la necesidad de encapsular los servicios se definen los siguientes servicios candidatos:

Manejo de Datos del Autor:

- Recibir y guardar en una base de datos la información del autor, nombre, apellidos, e-mail, dirección particular.
- Realizar consulta a la base de datos para obtener la información de los Autores.

Trabajo con los Documentos (Artículos):

- Guardar artículo en la red.
- Validar normas del documento.

Asignación del Arbitraje:

- Seleccionar de forma aleatoria los árbitros agrupados por temática.

Manejo de Notificaciones:

- Envío de notificaciones al autor con observaciones y caracterización de su artículo.

Con el objetivo de identificar los mejores escenarios para el desarrollo de los servicios web y maximizar los beneficios del negocio, se define la relación de los servicios candidatos de la capa aplicación con los servicios candidatos de la capa de negocio, creándose la composición de servicios que muestra la Figura 18:

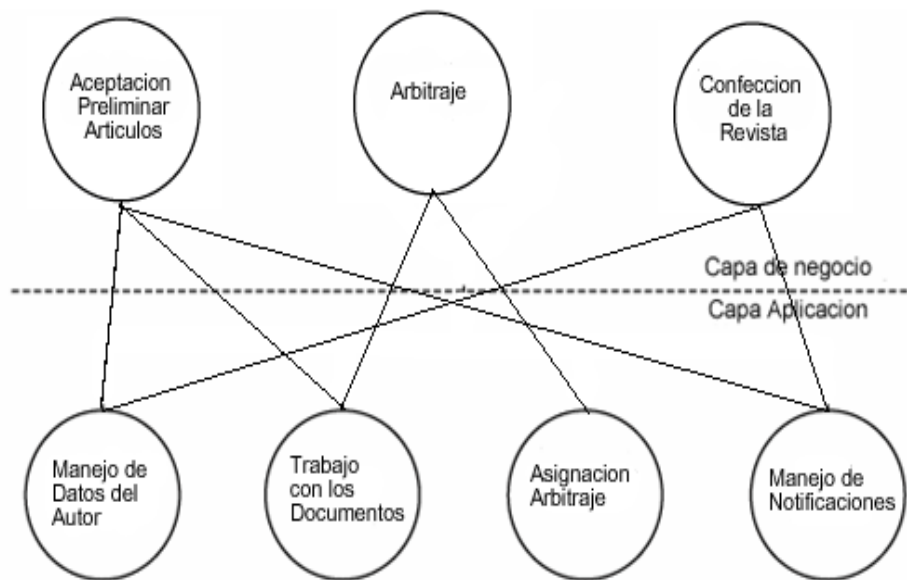


Figura 18. Relación entre servicios

El análisis orientado a servicios concluye con la definición de la relación entre los servicios dada las funciones que estos deben cumplir dentro de un determinado proceso de negocio.

3.2 Diseño orientado a servicios

Después del análisis orientado a servicios y de la revisión de los 12 pasos para el modelado de servicios resumidos en el epígrafe anterior, se establece una estructura para comenzar el diseño orientado a servicios.

El diseño orientado a servicios es el proceso concreto de diseño de servicios a partir de la lógica de los servicios candidatos y su interrelación con el proceso de negocio.

En el Anexo IV se ilustra el proceso iterativo de diseño de servicios para las distintas capas siguiendo un flujo de trabajo determinado.

De acuerdo a las capas de servicios necesarias para seguir una estrategia de arriba hacia abajo, la secuencia de diseño que se plantea está basada en la identificación de los servicios:

- Servicios de negocio centrados en tareas:
 1. Aceptación Preliminar de los Artículos (APA)
 2. Arbitraje (Arb)
 3. Confección de la Revista (CR)
- Servicios de aplicación:
 4. Datos Autor
 5. Manejo de Documentación
 6. Asignación Arbitraje
 7. Notificaciones

La solución propuesta de la arquitectura orientada a servicios para el modelo de negocio expuesto en el Capítulo II se ilustra a continuación (Figura 19):

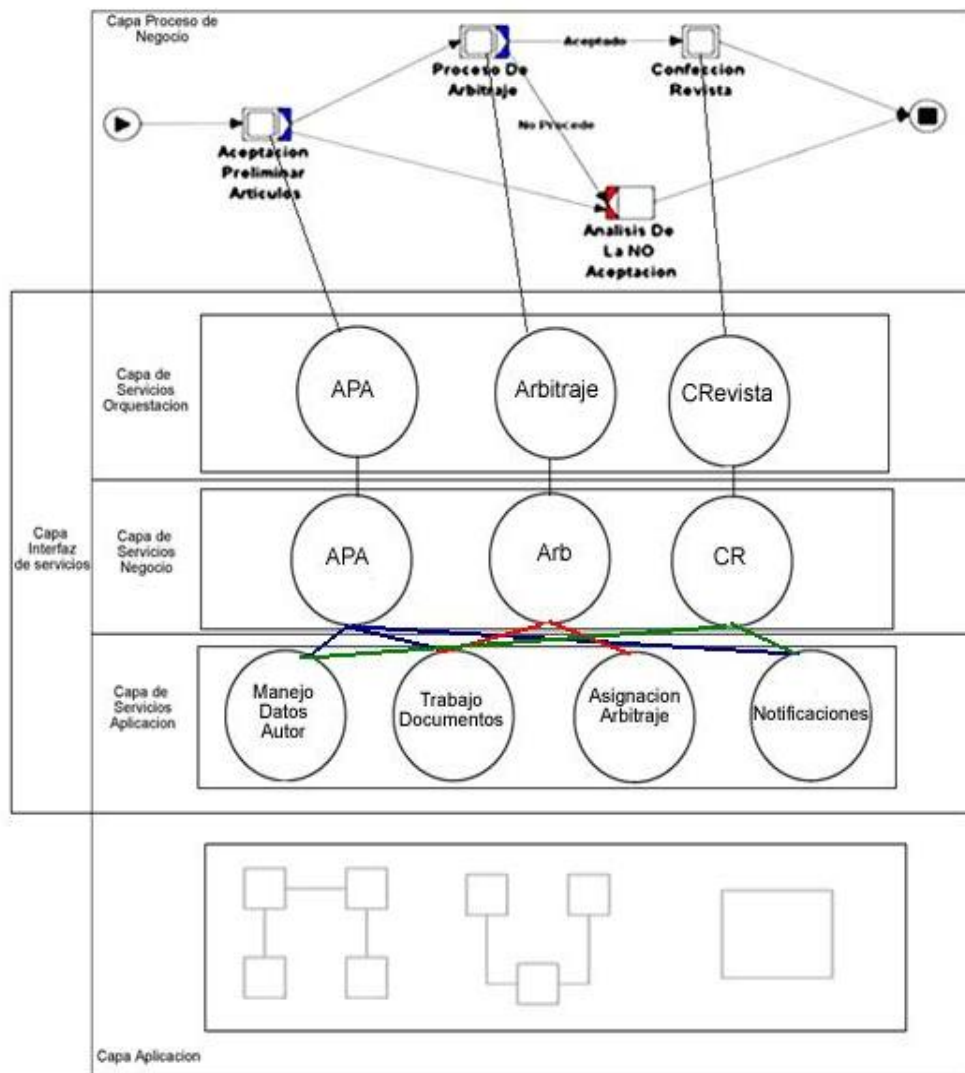


Figura 19. Propuesta de Arquitectura de Servicios

3.3 Desarrollo de servicios web

En una arquitectura orientada a servicios hay dos partes claramente diferenciadas, el modo de utilizar un servicio web y cómo desarrollarlo. Para lograr la implementación se combinan cuatro tecnologías:

- Hypertext Transfer Protocol (HTTP) como protocolo de red primario.
- SOAP/XML para el formato de carga.

- Universal Description, Discovery and Integration (UDDI) para el registro del servicio.
- Web Service Description Language (WSDL) para describir las interfaces del servicio.

Esta combinación se completa mediante los pasos:

- El programador desarrolla el servicio web.
- El programador describe el servicio web en un fichero WSDL.
- El programador publica el servicio web en un directorio como UDDI.
- La persona suscrita al directorio busca el servicio web.
- La persona suscrita al directorio invoca el servicio con SOAP.
- La persona suscrita al directorio recibe la respuesta mediante SOAP.

En este trabajo se le ha dado especial atención a la descripción de los servicios en ficheros WSDL.

Un servicio web es una clase cuya interfaz se puede hacer pública en un servidor web mediante un lenguaje de descripción de interfaces (WSDL). Dicha interfaz ofrece un conjunto de actividades que un cliente puede invocar. El cliente accede al servicio web usando los estándares de Internet. En este trabajo se desarrolla el código WSDL de los servicios de las capas de negocio y aplicación de la arquitectura propuesta.

WSDL es el lenguaje de descripción de servicios web basados en XML que permite describir la interfaz ofrecida por un servicio web, las operaciones que el servicio puede soportar y los parámetros de entrada y salida.

Un documento WSDL contiene un número de versión y un componente raíz *Definition*, que posee nombre, *Name* y un atributo *TargetNamespace*, que declara el espacio de nombres al que pertenecerán todos los nombres de componentes definidos en el documento.

El bloque de definiciones está compuesto a su vez por cinco bloques: *Types*, *Message*, *PortType*, *Binding* y *Service*.

- *Types*: Contiene las definiciones de los mensajes que pueden ser enviados o recibidos por un servicio.
- *Message*: Proporciona las asociaciones entre los mensajes y su definición en el esquema (.xsd).
- *PortType*: Define el conjunto de interfaces que ofrece el servicio web. Cada interfaz es asociada con uno o más mensajes. Se especifica una interfaz por cada uno de los métodos de acceso que se deseen utilizar SOAP, HTTPGET o HTTPPOST. En el siguiente ejemplo solamente se muestra para SOAP:
- *Binding*: Asocia la definición portType con un protocolo concreto.
- *Service*: Define una colección de puertos ofrecidos por el servicio web. Dichos puertos son cada elemento de Binding con su correspondiente URL.

La creación de los ficheros (.wsdl) se realizó utilizando como herramienta el XMLSpy, el cual permitió generar el código de descripción de los servicios.

Altova XMLSpy es el editor XML más utilizado en el entorno de desarrollo XML estándar. Permite modelar, editar, depurar y transformar todas las tecnologías XML. Es una herramienta de productividad para los desarrolladores que trabajan con las últimas tecnologías XML, servicios web y de base de datos.

XMLSpy permite:

- Editar WSDL y depurar SOAP en un entorno gráfico y amigable al usuario.
- Dispone del motor de validación más ajustado a estándares que existe.
- Incluye un motor XSLT premiado, un procesador XSLT 2.0 integrado con esquemas XML y un depurador XSLT potente.
- Permite abordar el desarrollo de servicios web en cualquiera de las plataformas principales, incluyendo Microsoft .NET, J2EE y Eclipse.

A continuación se muestra la descripción de los servicios web descubiertos para el negocio. Para la descripción de los servicios se ha utilizado la herramienta XMLSpy.

3.3.1 Definición del WSDL para el servicio Aceptación Preliminar de los Artículos

A continuación se muestra uno de los servicios centrados en tareas descubiertos para lograr un proceso de negocio orientado a servicios. Se hace notar que el servicio utiliza tres de los servicios de la capa de aplicación, que el tipo de protocolo de transporte es HTML, con comunicación RPC/Document basado en protocolo de codificación SOAP.

La interfaz del servicio Aceptación Preliminar Artículos (APA) aparece en la Figura 20:

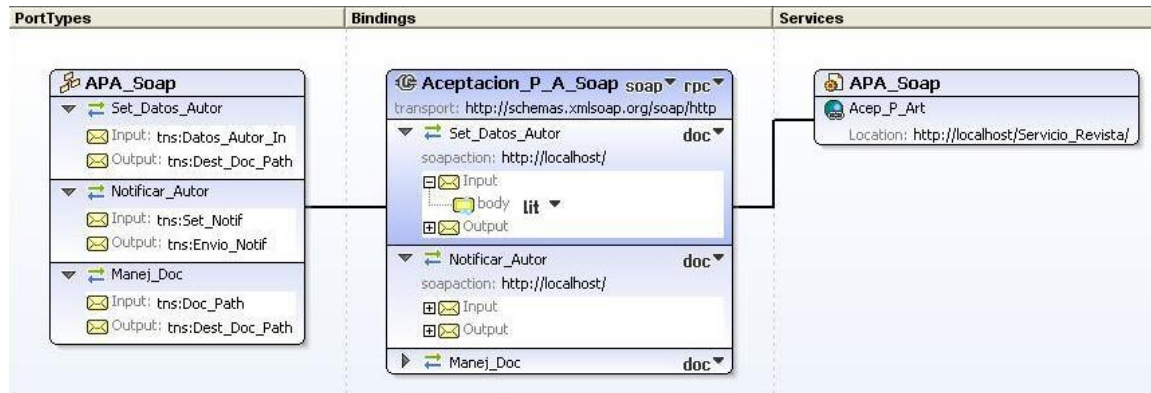


Figura 20. Interfaz de descripción del servicio web APA

El código de descripción del servicio web Aceptación Preliminar Artículos con sus tipos de datos aparece a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!-- edited with XMLSpy v2008 rel. 2 (http://www.altova.com) by mazuki (darksiderg) -->
```

```
<wsdl:definitions
                                xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                                xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                                xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
```

```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace">

    <wsdl:types>

        <xs:schema      targetNamespace="http://new.webservice.namespace"
elementFormDefault="qualified"/>

    </wsdl:types>

    <wsdl:message name="Set_Datos_Aut">

        <wsdl:part name="parameter" type="xs:string"/>

    </wsdl:message>

    <wsdl:message name="Dest_Doc_Path"/>

    <wsdl:message name="Set_Notif"/>

    <wsdl:message name="Envio_Notif"/>

    <wsdl:message name="Doc_Path"/>

    <wsdl:message name="Datos_Autor_In"/>

    <wsdl:portType name="APA_Soap">

        <wsdl:operation name="Set_Datos_Autor">

            <wsdl:input message="tns:Datos_Autor_In"/>

            <wsdl:output message="tns:Dest_Doc_Path"/>

        </wsdl:operation>

        <wsdl:operation name="Notificar_Autor">

            <wsdl:input message="tns:Set_Notif"/>

            <wsdl:output message="tns:Envio_Notif"/>

        </wsdl:operation>

    </wsdl:portType>

</wsdl:service>
```

```

        </wsdl:operation>

        <wsdl:operation name="Manej_Doc">

            <wsdl:input message="tns:Doc_Path"/>

            <wsdl:output message="tns:Dest_Doc_Path"/>

        </wsdl:operation>

    </wsdl:portType>

    <wsdl:binding name="Aceptacion_P_A_Soap" type="tns:APA_Soap">

        <soap:binding                                style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>

        <wsdl:operation name="Set_Datos_Autor">

            <soap:operation                            soapAction="http://localhost/"
style="document"/>

            <wsdl:input>

                <soap:body use="literal"/>

            </wsdl:input>

            <wsdl:output>

                <soap:body use="literal"/>

            </wsdl:output>

        </wsdl:operation>

        <wsdl:operation name="Notificar_Autor">

            <soap:operation                            soapAction="http://localhost/"
style="document"/>

            <wsdl:input>

                <soap:body use="literal"/>

```

```

        </wsdl:input>

        <wsdl:output>

            <soap:body use="literal"/>

        </wsdl:output>

    </wsdl:operation>

    <wsdl:operation name="Manej_Doc">

        <soap:operation                    soapAction="http://localhost/"
style="document"/>

        <wsdl:input>

            <soap:body use="literal"/>

        </wsdl: input>

        <wsdl:output>

            <soap:body use="literal"/>

        </wsdl:output>

    </wsdl:operation>

</wsdl:binding>

<wsdl:service name="APA_Soap">

    <wsdl:port                                name="Acep_P_Art"
binding="tns:Aceptacion_P_A_Soap">

        <soap:address location="http://localhost/Servicio_Revista/">

    </wsdl:port>

</wsdl:service>

</wsdl:definitions>

```

3.3.2 Definición del WSDL para el servicio Manejo Datos Autor

Aquí se describe uno de los servicios de aplicación descubierto para lograr un proceso de negocio orientado a servicios. Se hace notar que el protocolo de transporte es HTML, con comunicación RPC/Document basado en protocolo de codificación SOAP.

La interfaz del lenguaje para el servicio Manejo Datos Autores se detalla en la Figura 21:



Figura 21. Interfaz de descripción del servicio web Manejo de Datos

El código de descripción del servicio Manejo de Datos del Autor con sus tipos de datos (xsd) aparece a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!-- edited with XMLSpy v2008 rel. 2 (http://www.altova.com) by mazuki
(darksiderg) -->
```

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://localhost/"
xmlns:ns="http://new.webservice.namespace" targetNamespace="http://localhost/">
```

```
<wsdl:types>
```

```

        <xs:schema            xmlns:ns1="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace"
elementFormDefault="qualified">

            <xs:complexType name="Datos_Autor">

                <xs:attribute      name="nombre"      type="xs:string"
use="required"/>

                <xs:attribute      name="apellidos"    type="xs:string"
use="required"/>

                <xs:attribute      name="correo_electronico"
type="xs:string" use="required"/>

                <xs:attribute      name="Dir_particular" type="xs:string"
use="required"/>

            </xs:complexType>

            <xs:complexType name="Datos_Out">

                <xs:attribute      name="Tematica_Artículo"
type="xs:string" use="required"/>

                <xs:attribute      name="nombre"      type="xs:string"
use="optional"/>

                <xs:attribute      name="apellidos"    type="xs:string"
use="optional"/>

                <xs:attribute      name="correo"      type="xs:string"
use="required"/>

                <xs:attribute      name="dir_part"     type="xs:string"
use="required"/>

            </xs:complexType>

        </xs:schema>

```

```
</wsdl:types>

<wsdl:message name="Datos_Autor_In">
    <wsdl:part name="parameter" type="ns:Datos_Autor"/>
</wsdl:message>

<wsdl:message name="Datos_Autor_Out">
    <wsdl:part name="parameter" type="ns:Datos_Autor"/>
</wsdl:message>

<wsdl:message name="NewMessage"/>

<wsdl:message name="Datos_Out">
    <wsdl:part name="parameter" type="ns:Datos_Out"/>
</wsdl:message>

<wsdl:portType name="Datos_Autor_Soap">
    <wsdl:operation name="Set_Datos_Autor">
        <wsdl:input message="tns:Datos_Autor_In"/>
        <wsdl:output message="tns:Datos_Autor_Out"/>
    </wsdl:operation>
    <wsdl:operation name="Get_Datos_Autor">
        <wsdl:input message="tns:Datos_Autor_In"/>
        <wsdl:output message="tns:Datos_Out"/>
    </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="Manejo_Datos_Soap" type="tns:Datos_Autor_Soap">
    <soap:binding
                                style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
```

```
<wsdl:operation name="Set_Datos_Autor">
    <soap:operation                soapAction="http://localhost/"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Get_Datos_Autor">
    <soap:operation                soapAction="http://localhost/"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Datos_Soap">
    <wsdl:port name="Datos_Autor" binding="tns:Manejo_Datos_Soap">
        <soap:address location="http://localhost/Servicio_Revista/">
```

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

En el Anexo V se muestra el código de descripción de los demás servicios web identificados para el proceso de negocio de la revista, dada la relación que se establece entre los mismos en la propuesta de arquitectura definida en el epígrafe anterior.

3.4 Conclusiones del capítulo

El análisis orientado a servicios y diseño orientado a servicios ha permitido descubrir y modelar los servicios web a partir de los requerimientos del negocio, proponiéndose una solución que integra las arquitecturas orientadas servicios con la gestión de los procesos de negocio.

Se diseño una arquitectura de servicios para el proceso de edición de una revista arbitrada, se logra con la misma la interoperabilidad entre sistemas y plataformas, cumpliéndose objetivos basados en principios para el desarrollo de servicios web. La arquitectura se baso en una estrategia de arriba hacia abajo con enfoque de tres capas, haciéndose notar la importancia del modelo del proceso de negocio para el diseño de la arquitectura.

Se diseño la interfaz de descripción abstracta de los servicios web, lo que permite definir la funcionalidad abstracta y la forma de acceder a un servicio.

CONCLUSIONES Y RECOMENDACIONES

La tecnología BPM ha tomado fuerza en los últimos años y se desarrolla apoyada en el concepto de arquitectura orientada a servicios.

Con la realización de este trabajo se ha logrado:

- El descubrimiento y la especificación de los procesos de negocio que se desarrollan en el trabajo editorial de una revista arbitrada.
- La modelación de los procesos de negocio identificados, utilizando para ello una de las herramientas que actualmente aparecen libres para la gestión de procesos de negocio, YAWL (*Yet Another Workflow Language*) y que garantiza una validación estructural y matemática basada en redes de Petri.
- La definición de una arquitectura orientada a servicios, que se define de arriba hacia abajo, aprovechando las ventajas que ofrece la modelación del negocio, y que define los servicios de los cuales estará compuesto el sistema, sus interacciones, y con qué tecnologías de comunicación serán implementados.
- La definición de las interfaces que utiliza cada servicio para exponer su funcionalidad a través de WSDL son gobernadas por contratos, que definen claramente el conjunto de mensajes soportados, su contenido y las políticas aplicables.

Como trabajo futuro se recomienda:

- El desarrollo de los servicios web mediante plataforma Java, la cual brinda una interfaz amigable para la creación de los servicios. Java incorpora funcionalidad RPC basada en XML de acuerdo al estándar SOAP 1.1.
- Completar el ciclo de vida de la gestión de procesos de negocio para lograr la completa integración entre tecnologías de gestión de procesos y las arquitecturas orientadas a servicios.

BIBLIOGRAFIA

- AALST, W. M. P. V. D., BENATALLAH, B., CASATI, F., CURBERA, F. & VERBEEK, E. (2007) Business process management: Where business processes and web services meet. *Data Knowl. Eng.*, 61, 1-5.
- ACM (2009a) ACM TRANSACTIONS ON MODELING AND COMPUTER SIMULATION
- ACM (2009b) Design Automation of Electronic Systems.
- ACM (2009c) TRANSACTIONS ON COMPUTER SYSTEMS New York, NY 10036
- ACM (2009d) TRANSACTIONS ON COMPUTATIONAL LOGIC
- AGENDA, S. (2009) BPMS y Ciclo de Vida de los Procesos
- AGRÍCOLA, C. (2009) Revista sobre Agricultura Trópica. Editorial: Feijóo ed. Santa Clara.
- CUBILLOS, J. A. (2004) Composición Semántica de Servicios Web., Grupo de Ingeniería Telemática, Universidad de Cauca Colombia.
- EBPML.ORG (2009) XLANG.
- EBXML (2006) ebXML(Electronic Business using eXtensible Markup Language).
- ERL, T. (2005) *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR.
- GRAMAJE, C. P. (2002) Una Aproximación Metodológica al Desarrollo de Flujos de Trabajo. Valencia, Universidad Politécnica de Valencia.
- IBM, B. S., MICROSOFT, SAP AG, SIEBEL SYSTEMS (2002) Business Process Execution Language.
- LÓPEZ, G. J., I.; ECHEVERRÍA, A.; FIERRO, P. (PHD.) (2008) DISEÑO DE APLICACIONES CON ARQUITECTURA ORIENTADA A SERVICIOS.
- MODELOS DE CICLO DE VIDA ad-hoc.
- MEMBERS., W. M. C. (2009) *Workflow Client API Specifications (WAPI)*.
- PAGES, C. (2001) Web Services Flow Language (WSFL).
- SOURCEFORGE.NET (2009) YAWL.

UCI (2009) Revista Cubana de Ciencias Informáticas.

W3C (2003) Web Services Choreography Working Group.

W3C (2002) Web Service Choreography Interface.

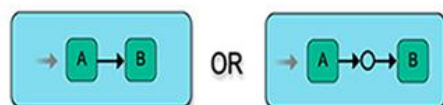
ZELKOWITZ, M. (2009) ADVANCES IN COMPUTERS. University of Maryland,
Department of Computer Science, College Park, USA.

Anexo

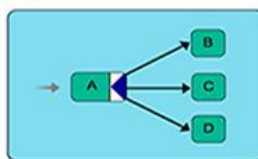
Anexo I. Patrones de Flujo de Trabajo

Patrones de Flujo de Trabajo

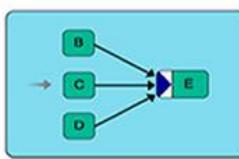
Secuencia



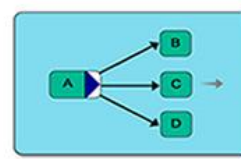
Split



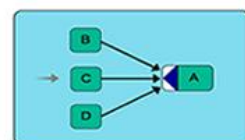
Sincronizacion



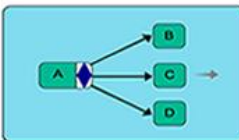
Eleccion Eclusiva



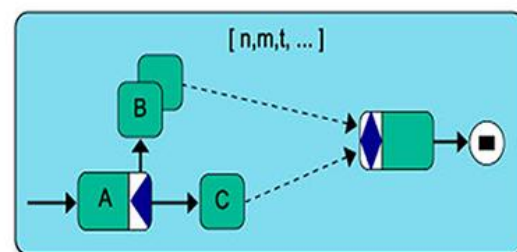
Union Simple



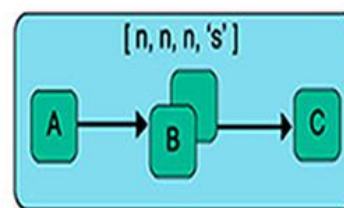
Eleccion Multiple

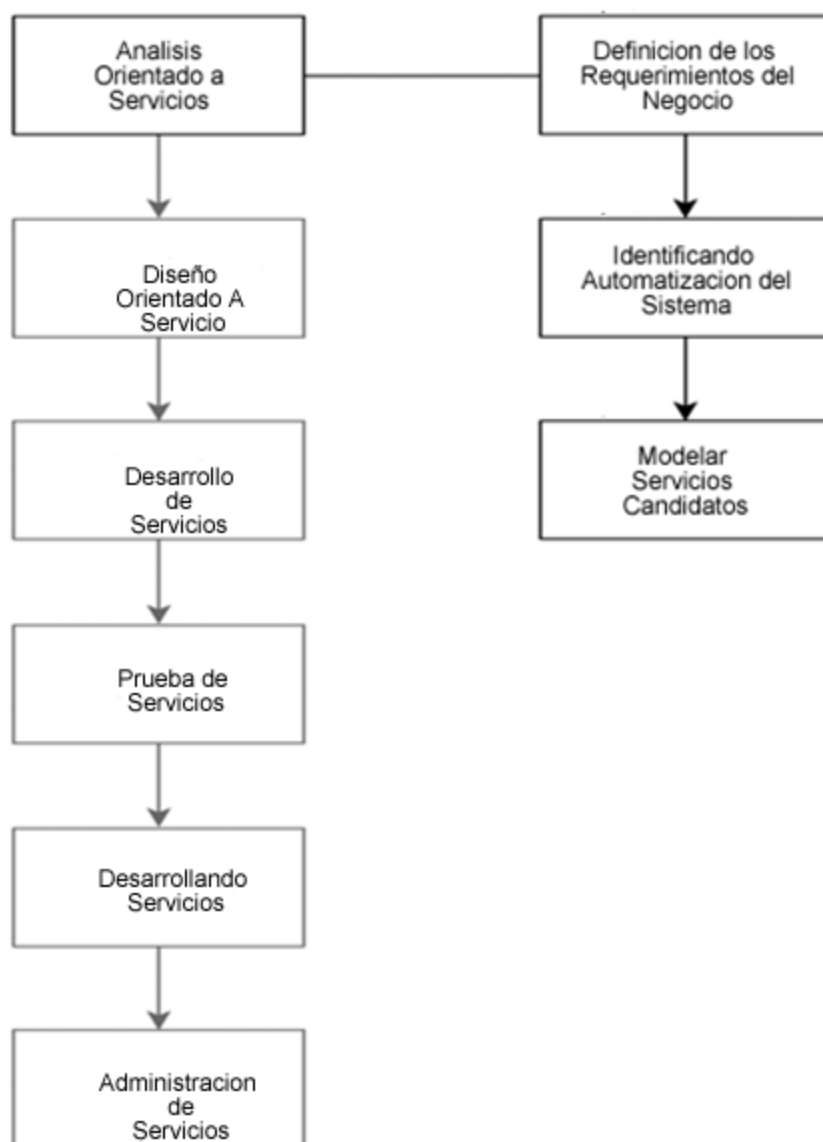


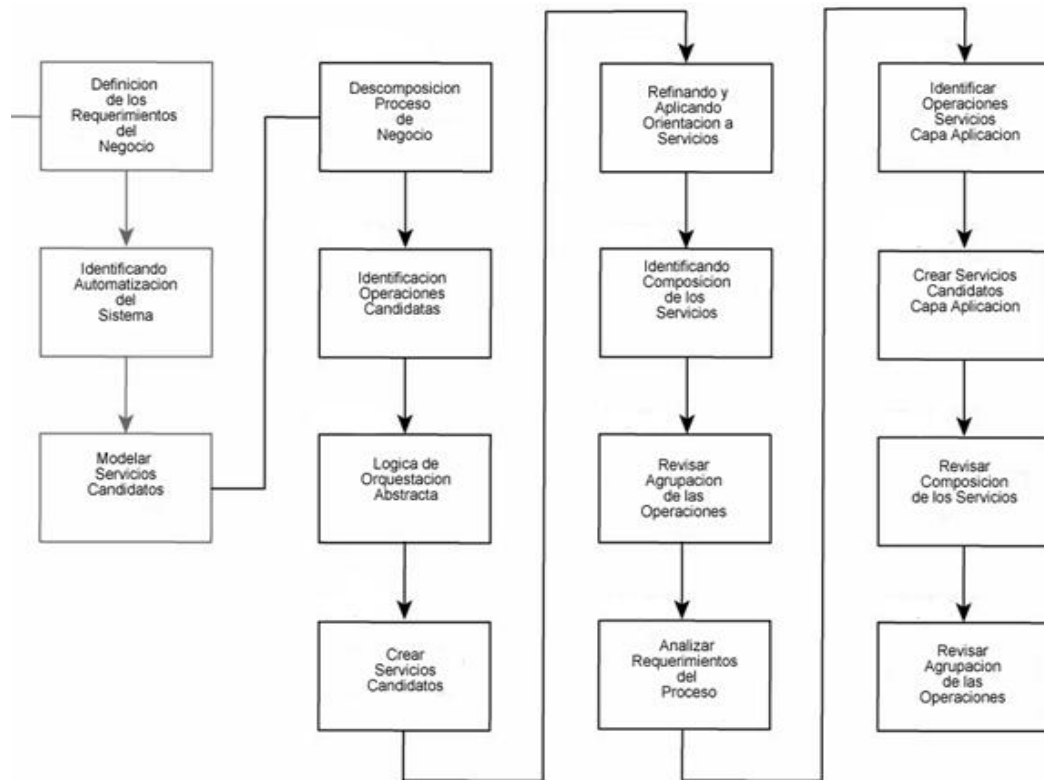
Eleccion Multiple con Sincronizacion

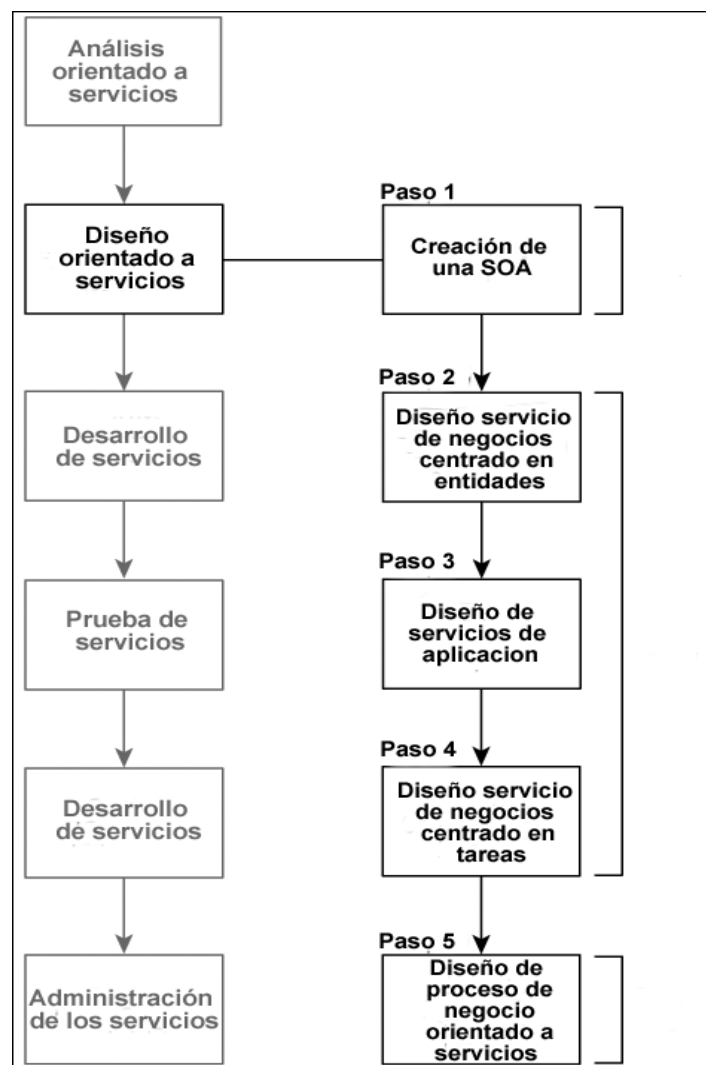


Multiples Instancias



Anexo II. Pasos del Análisis Orientado a Servicios

Anexo III. Pasos del Modelo Orientado a Servicios

Anexo IV. Diseño Orientado a Servicios

Anexo V. Código de descripción de servicios

A continuación el código de descripción del servicio centrado en tareas Arbitraje:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2008 rel. 2 (http://www.altova.com) by mazuki
(darksiderg) -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace">
  <wsdl:types>
    <xs:schema targetNamespace="http://new.webservice.namespace"
elementFormDefault="qualified"/>
  </wsdl:types>
  <wsdl:message name="Get_Permisos"/>
  <wsdl:message name="Set_Permisos"/>
  <wsdl:message name="Set_Arbitraje"/>
  <wsdl:message name="Get_Arbitraje"/>
  <wsdl:message name="Info_Autor"/>
  <wsdl:portType name="Arb_Soap">
    <wsdl:operation name="Permisos_Doc">
      <wsdl:input message="tns:Get_Permisos"/>
      <wsdl:output message="tns:Set_Permisos"/>
    </wsdl:operation>
    <wsdl:operation name="Selec_Arbitraje">
      <wsdl:input message="tns:Set_Arbitraje"/>
      <wsdl:output message="tns:Get_Arbitraje"/>
    </wsdl:operation>
    <wsdl:operation name="Info_Autor">
      <wsdl:input message="tns:Info_Autor"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="Arbitraje_Soap" type="tns:Arb_Soap">
    <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Permisos_Doc">
      <soap:operation soapAction="http://localhost/"
style="document"/>
    <wsdl:input>
```

```

        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Selec_Arbitraje">
    <soap:operation soapAction="http://localhost/"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Info_Autor">
    <soap:operation soapAction="http://localhost/"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Arbitraje_Soap">
    <wsdl:port name="Arbitraje" binding="tns:Arbitraje_Soap">
        <soap:address location="http://localhost/Servicio_Revista/">
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

A continuación el código de descripción del servicio centrado en tareas

Confeccion de la Revista:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2008 rel. 2 (http://www.altova.com) by mazuki
(darksiderg) -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace">
    <wsdl:types>

```

```

        <xs:schema targetNamespace="http://new.webservice.namespace"
elementFormDefault="qualified"/>
    </wsdl:types>
    <wsdl:message name="Datos_Autor"/>
    <wsdl:message name="Datos_Autor_Out"/>
    <wsdl:message name="Set_Notificaciones"/>
    <wsdl:message name="Envio_Notificaciones"/>
    <wsdl:portType name="CR_Soap">
        <wsdl:operation name="Get_Datos_Autor">
            <wsdl:input message="tns:Datos_Autor"/>
            <wsdl:output message="tns:Datos_Autor_Out"/>
        </wsdl:operation>
        <wsdl:operation name="Notificaciones">
            <wsdl:input message="tns:Set_Notificaciones"/>
            <wsdl:output message="tns:Envio_Notificaciones"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="Conf_Rev_Soap" type="tns:CR_Soap">
        <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="Get_Datos_Autor">
            <soap:operation soapAction="http://localhost/"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="Notificaciones">
            <soap:operation soapAction="http://localhost/"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="Confeccion_Revista_Sopa">
        <wsdl:port name="Conf_Revista" binding="tns:Conf_Rev_Soap">
            <soap:address location="http://localhost/Servicio_Revista/">
        </wsdl:port>
    </wsdl:service>

```

</wsdl:definitions>

A continuación el código de descripción del servicio de aplicación Selección de Arbitraje:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2008 rel. 2 (http://www.altova.com) by mazuki
(darksiderg) -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace">
  <wsdl:types>
    <xs:schema targetNamespace="http://new.webservice.namespace"
elementFormDefault="qualified">
      <xs:complexType name="Arbitraje">
        <xs:attribute name="Nombre_Abitro"
type="xs:string"/>
        <xs:attribute name="Temetica_Arbitros"
type="xs:string"/>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="Tematica_Arbitraje">
    <wsdl:part name="parameter" type="xs:string"/>
  </wsdl:message>
  <wsdl:message name="Conjunto_Arb">
    <wsdl:part name="parameter" type="tns:Arbitraje"/>
  </wsdl:message>
  <wsdl:message name="Set_Arbitros">
    <wsdl:part name="parameter" type="tns:Arbitraje"/>
  </wsdl:message>
  <wsdl:portType name="Arbitraje_Selec_Soap">
    <wsdl:operation name="Get_Arbitros">
      <wsdl:input message="tns:Tematica_Arbitraje"/>
      <wsdl:output message="tns:Conjunto_Arb"/>
    </wsdl:operation>
    <wsdl:operation name="Set_Arbitros">
      <wsdl:input message="tns:Set_Arbitros"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="Seleccionar_Arb" type="tns:Arbitraje_Selec_Soap">
```

```

        <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="Get_Arbitros">
            <soap:operation soapAction="http://localhost/"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="Set_Arbitros">
            <soap:operation soapAction="http://localhost"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="Selec_Arb_Soap">
        <wsdl:port name="Arbitraje" binding="tns:Seleccionar_Arb">
            <soap:address location="http://localhost/Servicio_Revista"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>

```

A continuación el código de descripción del servicio de aplicación Manejo de Documentos:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2008 rel. 2 (http://www.altova.com) by mazuki
(darksiderg) -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://localhost/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:ns="http://new.webservice.namespace" targetNamespace="http://localhost/">
    <wsdl:types>
        <xs:schema xmlns:ns1="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace"
elementFormDefault="qualified">
            <xs:complexType name="Save_Doc">

```

```

        <xs:attribute name="camino" type="xs:string"
use="required"/>
        <xs:attribute name="dest_camino" type="xs:string"
use="required"/>
    </xs:complexType>
</xs:schema>
</wsdl:types>
<wsdl:message name="Guardar_doc">
    <wsdl:part name="parameter" type="ns:Save_Doc"/>
</wsdl:message>
<wsdl:message name="ssageResponse">
    <wsdl:part name="parameter" type="xs:string"/>
</wsdl:message>
<wsdl:message name="Get_Permisos">
    <wsdl:part name="Nomb_Apell" type="xs:string"/>
    <wsdl:part name="Tipo_permiso" type="xs:string"/>
</wsdl:message>
<wsdl:message name="Set_Permisos">
    <wsdl:part name="Nomb_Apell" type="xs:string"/>
</wsdl:message>
<wsdl:portType name="Documentos_Soap">
    <wsdl:operation name="Guardar_Doc">
        <wsdl:input message="tns:Guardar_doc"/>
        <wsdl:output message="tns:Guardar_doc"/>
    </wsdl:operation>
    <wsdl:operation name="Permisos_Doc">
        <wsdl:input message="tns:Get_Permisos"/>
        <wsdl:output message="tns:Set_Permisos"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="Doc_Soap" type="tns:Documentos_Soap">
    <soap12:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Guardar_Doc">
        <soap12:operation soapAction="http://localhost/"
soapActionRequired="true" style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Permisos_Doc">
        <soap12:operation soapAction="http://localhost/"
soapActionRequired="true" style="document"/>

```

```

        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="Document_Soap">
    <wsdl:port name="Doc_Soap" binding="tns:Doc_Soap">
        <soap12:address
location="http://localhost/Servicio_Revista/">
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

A continuación el código de descripción del servicio de aplicación Notificaciones:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2008 rel. 2 (http://www.altova.com) by mazuki
(darksiderg) -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://new.webservice.namespace"
targetNamespace="http://new.webservice.namespace">
    <wsdl:types>
        <xs:schema targetNamespace="http://new.webservice.namespace"
elementFormDefault="qualified">
            <xs:complexType name="Notificar_Autor">
                <xs:attribute name="nombre" type="xs:string"
use="required"/>
                <xs:attribute name="apellidos" type="xs:string"
use="required"/>
                <xs:attribute name="observaciones" type="xs:string"
use="required"/>
            </xs:complexType>
            <xs:complexType name="Datos_Autor">
                <xs:attribute name="nombre" type="xs:string"/>
                <xs:attribute name="apellidos" type="xs:string"/>
                <xs:attribute name="dir" type="xs:string"/>
                <xs:attribute name="correo" type="xs:string"/>
            </xs:complexType>

```

```

        <xs:complexType name="Info_Autor">
            <xs:attribute name="Tematica_Artículo"
type="xs:string"/>
            <xs:attribute name="Nombre_Artículo"
type="xs:string"/>
            <xs:attribute name="Observaciones_Autor"
type="xs:string"/>
        </xs:complexType>
    </xs:schema>
</wsdl:types>
<wsdl:message name="Datos_In">
    <wsdl:part name="parameter" type="tns:Datos_Autor"/>
</wsdl:message>
<wsdl:message name="Notificacion">
    <wsdl:part name="parameter" type="tns:Notificar_Autor"/>
</wsdl:message>
<wsdl:message name="Info_Autor">
    <wsdl:part name="parameter" type="tns:Info_Autor"/>
</wsdl:message>
<wsdl:portType name="Notificaciones_Autor_Soap">
    <wsdl:operation name="Notificar_Autor">
        <wsdl:input message="tns:Datos_In"/>
        <wsdl:output message="tns:Notificacion"/>
    </wsdl:operation>
    <wsdl:operation name="Info_Autor">
        <wsdl:input message="tns:Info_Autor"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="Notificaciones_Soap"
type="tns:Notificaciones_Autor_Soap">
    <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Notificar_Autor">
        <soap:operation soapAction="http://localhost/"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Info_Autor">
        <soap:operation soapAction="http://localhost/"
style="document"/>
        <wsdl:input>

```

```
                <soap:body use="literal"/>
            </wsdl:input>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="Notif_A_Soap">
        <wsdl:port name="Notificaciones"
binding="tns:Notificaciones_Soap">
            <soap:address location="http://localhost/Servicio_Revista/">
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

