



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Posgrado en Ciencia e Ingeniería de la Computación

**Arquitectura Orientada a Servicios aplicada a Flujos de Trabajo en
Organizaciones Jerárquicas**

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN CIENCIA E INGENIERÍA EN COMPUTACIÓN

PRESENTA:

VÍCTOR MANUEL CORZA VARGAS

M. EN C. GUSTAVO ARTURO MÁRQUEZ FLORES

IIMAS, UNAM

Ciudad Universitaria CD. MX.

Febrero 2017



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice General

Introducción	5
Problemática	5
Hipótesis.....	5
Objetivos	6
Aportación.....	6
Trabajos relacionados	6
Breve descripción de cada capítulo	7
1. Marco Teórico y Estado del Arte.....	9
1.1 Arquitectura Orientada a Servicios (SOA) y Servicios Web	9
1.1.1 Arquitectura de referencia SOA (SOA RA)	10
1.1.2 Bloques de Construcción SOA	16
1.1.3 El costo de una SOA y el retorno de Inversión	16
1.2 Servicios Web.....	17
1.2.1 Propiedades de los servicios Web	17
1.2.2 Arquitectura de los servicios Web.....	18
1.2.3 Estándares relacionados con servicios Web.	19
1.3 La tecnología de flujos de trabajo.	25
1.3.1 Estándares relacionados con flujos de trabajo	26
1.4 Estado del Arte de la composición de servicios.	27
1.4.1 Composición automática de servicios	28
1.4.2 Planeación (planning)	28
1.4.3 Composición de Servicios Ayudados por Mediación (Mediation-Aided Service Composition).....	28
1.4.4 Verificación de flujos de trabajo basados en servicios.	28
1.4.5 Ejecución Descentralizada de Flujos de Trabajo	29
1.4.6 Sistemas científicos de flujos de trabajo.....	30
1.5 Estructura organizacional	30
1.5.1 Estructura Jerárquica	30
1.5.2 Estructura matricial	31
1.5.3 Implicaciones de la estructura organizacional en una SOA	31
2. El Modelo Abstracto SOA de Microsoft.....	34

2.1 Definiendo la pertinencia de SOA.	34
2.2 Justificación del modelo de referencia SOA	34
2.3. El modelo de referencia abstracto SOA de Microsoft	35
2.4 Capacidades Arquitectónicas Recurrentes	36
2.4.1 Mensajes y Servicios.....	36
2.4.2 Flujo de Trabajo y Procesos.....	37
2.4.3 Datos.....	37
2.4.4 Experiencia de usuario.....	39
2.5 ¿Pueden considerarse equivalentes los conceptos de recurrencia y transversalidad?	39
2.6 Consideraciones del Modelo Abstracto de SOA	40
2.7 Plataforma Tecnológica SOA de Microsoft.	43
3. Propuesta de SOA para las PyMEs	45
3.1 Modelado del negocio (UML)	46
3.1.1 Identificación de Procesos de Negocio	46
3.1.2 Identificación de los Roles en los Procesos de Negocio	47
3.1.3 Caso de uso general.....	47
3.1.4 Descripción de casos de uso	47
3.1.5 Diagramas de secuencia UML	47
3.1.6 Diagrama de proceso	48
3.1.7 Diccionario de Datos.....	48
3.1.8 Diagrama de clases	48
3.2 Implementación de los servicios web	49
3.2.1 Windows Communication Foundation (WCF).....	49
3.2.2 Librería nusoap.....	51
3.3 Modelado del flujo de trabajo	52
3.4 Implementación de la interface de usuario.....	53
3.5 Despliegue	54
3.6 Seguimiento	55
3.6.1 Monitor Gráfico de Progreso	55
4. Caso de Prueba: “Solicitud de horas adicionales para técnicos académicos” en la Dirección General del Colegio de Ciencias y Humanidades.....	60
4.1 Modelado del negocio (UML)	60
4.1.1 Identificación del proceso de negocio	60

4.1.2 Identificación de los Roles en los Procesos de negocio	61
4.1.3 Caso de Uso General.....	61
4.1.4 Descripción del caso de uso	62
4.1.5 Diagrama de Secuencia del flujo básico.....	63
4.1.6 Diagrama de Proceso	63
4.1.7 Diccionario de Datos	65
4.1.8 Diagrama de clases	69
4.1.9 Descripción de la propuesta de solución	70
4.2 Implementación de los servicios Web.....	70
4.2.1 Servicios obtenidos del sistema SPAC-E	71
4.2.2 Servicios obtenidos del sistema DCDPA.....	76
4.3 Modelado del Flujo de Trabajo	78
4.4 Implementación de la Interfaz de Usuario	79
4.5 Despliegue	80
4.5.1 Descripción del hardware de despliegue:.....	80
4.6 Seguimiento	81
4.6.1 Funcionalidad de la aplicación	81
4.6.2 Monitoreo de una instancia de flujo de trabajo a través del depurador de flujos de trabajo	81
Evaluación y Beneficios de la implementación del modelo	83
Conclusiones.....	83
Referencias	84

Introducción

En la actualidad los procesos de negocio en una organización requieren la interacción entre distintos departamentos, esto genera la necesidad de integrar sistemas que están desplegados a través de diferentes plataformas, que están desarrollados en distintos lenguajes y que utilizan diferentes tecnologías. Estos sistemas además operan de manera independiente para dar soporte sólo al área que les corresponde, el resultado es una funcionalidad aislada, múltiples instancias de los mismos datos, actividades manuales redundantes, respuestas ineficientes para los clientes y costos de operación más altos.

La Arquitectura Orientada a Servicios, desde este momento SOA, y la composición de servicios a través de flujos de trabajo se han convertido en los últimos años en términos muy utilizados en el ámbito de las organizaciones jerárquicas, dado que éstas requieren de una colaboración entre empleados y directivos en un entorno muy estructurado. No obstante la rigidez de estructura organizacional jerárquica, ésta sigue siendo utilizada ampliamente en el ámbito de las PyMEs y de las grandes empresas.

Por estos motivos, ha surgido la necesidad de pensar en nuevas arquitecturas para la implementación de este tipo de soluciones que permitan a las organizaciones adaptarse rápidamente a los cambios que se producen en su entorno, mediante el desarrollo de procesos de negocio multifuncionales, facilitando la comunicación entre aplicaciones de software de acuerdo a la lógica de negocio prescrita por los procesos.

Problemática

Actualmente un gran número de Pequeñas y Medianas Empresas (PyMEs) se encuentran operando con entornos heterogéneos compuestos por plataformas y sistemas dispares. En este contexto, la conexión de estas aplicaciones y su vinculación con los procesos de negocio es un enorme desafío para la mayoría de las organizaciones debido a que:

- a. Las aplicaciones y sistemas clave para las operaciones de la organización están aislados. Las aplicaciones y sistemas son activos de la empresa y deben interrelacionarse con otros sistemas, tanto de la propia organización como externos a la misma. Cuando los sistemas representan islas desconectadas entre sí, los procesos de negocios que se apoyan en estos sistemas acaban siendo ineficientes.
- b. Muchos procesos de negocio necesitan gestionar y coordinar transacciones de larga duración como pedidos y órdenes de servicio, en las que participan diversos departamentos y sistemas de información de la propia organización.

Hipótesis

Una organización jerárquica sin importar su dimensión puede considerar pertinente la implementación de una SOA, siempre y cuando haya identificado la necesidad de integrar sistemas heterogéneos para modelar sus procesos de negocio.

Cada SOA que se implementa tiene características específicas que le permiten adaptarse a la organización en cuestión, las capacidades a implementar deben ser seleccionadas en función de

las necesidades de negocio de la empresa. Estas capacidades además deben estar bien ubicadas en las distintas capas de abstracción de una SOA para poder definir claramente el ámbito de una implementación y con ello evitar costos innecesarios.

Enmarcar una implementación en un modelo abstracto también contribuye a simplificar el proceso de adición de nuevas funcionalidades, especialmente aquellas que se extienden en distintos niveles de abstracción.

Otros factores para la reducción del costo de implementación son la existencia de herramientas de bajo costo para la creación de aplicaciones orientadas a servicios, y la disponibilidad a priori de una infraestructura suficiente que soporte aplicaciones distribuidas.

Objetivos

1. Diseñar una SOA de bajo costo aplicable a una organización jerárquica.
2. Enmarcar cada una de las funcionalidades a implementar en un modelo abstracto SOA que permita que el modelo propuesto sea escalable.
3. Implementar una aplicación de monitoreo de flujos de trabajo que permita a usuarios no especializados interactuar con los procesos de negocio.

Aportación

Un modelo de Integración de Procesos de Negocio de bajo costo basado en el modelo abstracto SOA de Microsoft. Este modelo guiará al desarrollador en la implementación de una SOA para el ámbito de las PyMEs concebidas como organizaciones jerárquicas.

Una herramienta de seguimiento de las instancias de un flujo de trabajo para que usuarios no programadores como analistas de negocio puedan interactuar con los procesos de negocio.

Trabajos relacionados

Picón, Fontana y Martín [1] abordan la implementación de un sistema de Integración de Procesos de Negocio (BPI) considerando a SOA únicamente como una tecnología equivalente con el uso de Servicios Web. La implementación describe un ejemplo de envío de mensajes entre sistemas heterogéneos, no refleja un caso de uso práctico.

Reyes y Villasís [2] proponen una fusión de las técnicas “Business Process Management: Rapid Analysis&Design”(BPM:RAD) y “Polymita”, resultando una metodología concreta y práctica para la implementación de proyectos BPM, sin embargo no ofrece un marco de referencia SOA que fundamente las selección de las actividades a realizar.

Bazán [3] establece una visión global que identifica las etapas y su interacción para la integración de procesos de negocio y cubre el ciclo de vida de los procesos de negocio y el ciclo de vida del software de una manera unificada, sin embargo nuevamente carece de un marco abstracto SOA.

El trabajo relacionado existente hace uso erróneo del concepto SOA para referirse únicamente al uso de servicios Web, sin embargo ésta consideración sólo apunta hacia la implementación y omite el perfil de “arquitectura” que una SOA lleva implícito. SOA no requiere del uso de la tecnología de los servicios Web para ser implementada, prueba de ello es que antes de su existencia ya existían sistemas orientados a servicios como CORBA(Common Object Request Broker Architecture) o DCOM(Distributed Component Object Model). Los servicios Web se han popularizado debido a las facilidades que ésta tecnología ofrece para la implementación de una SOA.

Breve descripción de cada capítulo

Esta tesis está dividida en cuatro capítulos. En el primer capítulo se describe el estado del arte en enfoques, tecnologías y herramientas que dan soporte a SOA, a flujos de trabajo y a la integración de servicios. Finalmente se abordan los conceptos sobre estructura organizacional necesarios para la implementación de una SOA en una organización jerárquica.

En el segundo capítulo se describe el modelo abstracto SOA utilizado a lo largo de este trabajo como marco de referencia para establecer las actividades pertinentes para la implementación de una SOA para el ámbito de las Pymes concebidas como organizaciones jerárquicas.

En el tercer capítulo se propone un modelo en el que se aborda paso a paso el proceso de implementación de una SOA. Este proceso consta de seis etapas, cuatro de ellas enmarcadas en el modelo abstracto SOA de Microsoft.

Finalmente, en el cuarto capítulo se desarrolla un caso de uso en el dominio de una organización que pone a prueba y refuerza el modelo propuesto.

Capítulo 1

Marco Teórico y

Estado del Arte

1. Marco Teórico y Estado del Arte

En este capítulo se aborda el estado del arte en enfoques, tecnologías y herramientas que dan soporte a una SOA. Se explican las características y niveles abstractos del modelo conceptual SOA propuesto por el OpenGroup¹, un consorcio industrial que hace posible el logro de objetivos organizacionales a través de estándares de IT.

También se describe a detalle la arquitectura y los estándares que permiten la implementación de los dos bloques de construcción a utilizar para la implementación del caso de prueba: los servicios Web y los Flujos de Trabajo. Finalmente hablo sobre las tendencias actuales con respecto a la composición de servicios a través de flujos de trabajo.

1.1 Arquitectura Orientada a Servicios (SOA) y Servicios Web

El OpenGroup en “The SOA Source Book” [4] define a la Arquitectura Orientada a Servicios como:

“Arquitectura Orientada a Servicios (SOA) es un estilo arquitectónico que soporta orientación a servicios. Orientación a servicios es un forma de pensar en términos de servicios y desarrollo basado en servicios y la salida de esos servicios”

El término SOA fue utilizado por primera vez para describir un estilo de cómputo multinivel que ayudaba a las organizaciones a compartir lógica de programación y datos a través de múltiples aplicaciones y modelos de uso [5].

Thomas Erl [6], quien es reconocido como uno de los principales contribuyentes de SOA la describe como una arquitectura que es abierta, ágil, extensible, federada y adaptable. Una arquitectura que está compuesta de servicios autónomos, diversos en cuanto a la plataforma, interoperables y potencialmente reutilizables. Erl también resume los siguientes 8 principios de SOA:

Contrato Estandarizado de Servicio. Los servicios deberían exponer su interfaz y su nivel de servicio explícitamente, en una manera estándar. Esa es la única información que el usuario necesita.

Débil Acoplamiento de Servicios. Los servicios pueden colaborar con otros, pero sus dependencias deberían ser minimizadas de tal manera que en sus comunicaciones, sólo necesiten conocer el contrato de otros.

Abstracción del Servicio. Los servicios deberían ocultar al mundo exterior su funcionamiento interno, excepto el contrato. Por ejemplo, un servicio no debería exponer a los usuarios sus detalles técnicos, tales como el lenguaje de programación, el sistema operativo, y la base de datos utilizada.

Reutilización de servicios. Los servicios deberían ser diseñados con una granularidad que promueva su reutilización. Si un servicio es utilizado por un solo usuario, entonces no hay razón para exponerlo como un servicio.

¹ The Open Group, <http://www.opengroup.org/>

Autonomía del servicio. Un servicio debería tener control sobre su conducta y responder a la invocación de solicitudes. En un sistema de servicio, un control centralizado no es necesario.

Ausencia de Estado en los Servicios. Los servicios deberían minimizar la información de estado expuesta a otros usuarios y en terminales para favorecer el acoplamiento débil y escalabilidad de la arquitectura. Si un servicio no tiene estado (stateless), el lado del servidor no necesita mantener la información de sesión ni la de estado; es más importante que una solicitud pueda ser ruteada fácilmente para balancear la carga.

Capacidad para que el servicio sea descubierto. Los servicios deberían estar anotados con metadatos para que puedan ser descubiertos por los usuarios.

Capacidad para la composición de servicios. Los servicios deberían ser capaces de ser compuestos para cumplir requerimientos nuevos y más complejos. Este principio está relacionado de manera cercana con la reusabilidad. Esto es, los servicios son compuestos de tal manera que pueden ser reutilizados en diferentes aplicaciones.

1.1.1 Arquitectura de referencia SOA (SOA RA)

La arquitectura de referencia SOA es concebida por el OpenGroup como una arquitectura de 9 capas que representan nueve agrupaciones clave de consideraciones y responsabilidades que típicamente emergen en el proceso de diseño de una solución SOA o al definir un estándar arquitectónico empresarial:

Capas de Implementación e interface de servicios

Tres de las capas se encargan de la implementación y la interface de los servicios:

Capa de Sistemas Operacionales.

Todos los elementos de tiempo de ejecución de la arquitectura residen en esta capa. Efectivamente, esta capa puede conceptualmente ser pensada como el tiempo de ejecución de la solución. Si suspendemos momentáneamente las operaciones dentro de esta capa descubriremos la separación de aspectos más relacionados con el consumo de los servicios, los procesos que son coreografiados en un flujo y los servicios cuyas interfaces están expuestas para consumo.

Esta capa específicamente incluye:

- Toda la infraestructura de software y hardware necesario para soportar una SOA y sus componentes en tiempo de ejecución y de diseño (herramientas).
- El hospedaje operacional y en tiempo de ejecución de los componentes físicos subyacentes del sistema.
- Los recursos requeridos para soportar la funcionalidad de los servicios en una SOA incluyendo recursos incluidos con la aplicación o personalizados, nuevos servicios, servicios creados a través de composición, servicios de infraestructura, etcétera.

Capa de Componente de Servicio.

Esta capa contiene componentes de software, cada uno de los cuales provee la implementación de los servicios y sus operaciones, debido a esta característica recibe su nombre “componente de servicio”. Esta capa también contiene los componentes técnicos y funcionales que facilitan que un componente de servicio pueda componer uno o más servicios. Los componentes de servicios reflejan la definición del servicio que ellos representan ambos en términos de funcionalidad y calidad del servicio (Quality of Service QoS). Los componentes de servicios enlazan el contrato y la implementación del servicio en la capa de sistemas operacionales. Los componentes están hospedados en contenedores que soportan las especificaciones de servicio.

La capa de componente de servicio manifiesta el ajuste de IT con cada servicio (contrato/descripción, especificación) definido en la capa de Servicios; garantiza la alineación de la implementación IT(Information technologies)con la descripción del servicio.

En detalle, cada componente de servicio cumple con los siguientes objetivos:

- Realiza uno o más servicios.
- Provee un punto de aplicación a la realización del servicio.
- Habilita la flexibilidad IT reforzando el desacoplamiento en el sistema, ocultando detalles volátiles de la implementación de consumidores de servicio.

Existen múltiples categorías de capacidades que la capa de componente de servicio necesita para dar soporte a la SOA RA. Estas capacidades incluyen capacidades en tiempo de diseño y de ejecución. Estas categorías de capacidades son:

- Realización de servicios e implementación. La habilidad para crear un servicio.
- Publicación de servicio y exposición. La habilidad para publicar el servicio (contrato/descripción) cumpliendo estándares e interoperabilidad para otras capas de la SOA RA y repositorio de servicio de tiempo de diseño y registro de servicios en tiempo de ejecución en la capa de gobernabilidad.
- Despliegue del servicio. Habilidad para soportar el despliegue del servicio físico a la plataforma existente de solución la cual contiene el componente de solución de servicio asociado.
- Invocación del servicio. Habilidad para soportar una invocación de servicio que cumpla con estándares, que sea interoperable y que sea invocado en tiempo de ejecución.
- Enlace de servicios.
 - Habilidad para soportar la interoperabilidad de servicios.
 - Habilidad para implementar una parte del patrón agente.
 - Habilidad para convertir la descripción de servicio a llamadas de servicio soportadas por la plataforma.

Capa de Servicios.

La capa de servicios comprende todos los servicios definidos dentro de SOA. Esta capa puede ser entendida como un contenedor para capacidades de negocio, servicios, contratos de servicios y descripciones que serán usadas en tiempo de ejecución.

Los componentes de servicios o las aplicaciones empresariales existentes (sistemas legados, aplicaciones empaquetadas, etc.) son responsables por la implementación real. En tiempo de ejecución, esta implementación reside en un contenedor dentro de la capa de sistemas operacionales, que es responsable de la ejecución.

La capa de servicios es la capa de la SOA que describe capacidades funcionales de los servicios en la SOA. La capa de servicios introduce la noción de servicios como interfaces bien definidas para una capacidad. La noción de “programar interfaces en lugar de la implementación” sólo existía en los modelos de programación como Java y C++, pero nunca fue parte del estilo arquitectónico hasta la llegada de SOA y los servicios.

Esta capa fundamentalmente provee soporte para servicios, desde una perspectiva de la etapa de diseño. En particular, desde una perspectiva de la etapa de diseño esta capa incluye valores como descripciones de servicios, contratos y políticas. Define las capacidades del tiempo de ejecución para el despliegue de servicios. También provee los elementos de contrato de servicios que pueden ser creados en la etapa de diseño para soportar requerimientos subsecuentes durante la ejecución.

Capas que soportan el consumo de servicios

Tres capas soportan el consumo de servicios (Capa de Procesos de Negocio, Capa de Consumidor y Capa de Integración):

Capa de consumidor.

La capa de consumo es el punto donde los consumidores interactúan con la SOA. Habilita a una SOA para soportar un conjunto de funcionalidades independientes del cliente, el cual es consumido y renderizado a través de uno o más canales (plataformas de cliente y dispositivos). Por consiguiente este es el punto de entrada para consumidores humanos (humanos y otras aplicaciones/sistemas) y servicios provenientes de fuentes externas (ej. Escenarios Business to Business B2B).

De hecho, la capa de consumo es el punto de entrada para todos los consumidores externos a la SOA. Estos pueden ser otros sistemas, otras SOA's, consumidores de servicios en la nube, usuarios humanos, etcétera.

Este desacoplamiento entre el consumidor y el resto de una SOA provee a las organizaciones la habilidad de soportar agilidad, reutilización mejorada y permite mejorar la calidad y consistencia. Los canales pueden ser entendidos como plataformas por las cuales los consumidores SOA acceden a los servicios. Ejemplos de canales son interfaces web, sistemas IVR(Interactive Voice Response), ambos podrían aprovechar la misma funcionalidad dentro de SOA.

Otros medios de consumo de servicios pueden ser: una Interfaz Gráfica de Usuario (GUI por sus siglas en inglés) o bien una conexión a los servicios a través de una Interfaz de Programación de Aplicaciones (API). La capa de consumidor es responsable de invocar los servicios. Accede a los

servicios a través de la capa de integración o bien puede acceder a los servicios directamente si en la arquitectura no se desea implementar una capa de integración.

Capa de Proceso de Negocio.

La capa de proceso de negocio cubre la representación de procesos y composición, y provee bloques de construcción para agregar servicios débilmente acoplados como una secuencia de procesos alineados con objetivos de negocios. El flujo de datos y control de datos son utilizados para habilitar interacciones entre servicios y procesos de negocio. La interacción puede existir dentro de una empresa o a través de múltiples empresas.

Esta capa incluye el flujo de intercambio de información entre participantes (usuarios individuales y entidades de negocio), recursos, y procesos en una variedad de formas para lograr los objetivos de negocio. Gran parte de la información intercambiada puede también incluir mensajes no estructurados y no transaccionales. La lógica de negocio es utilizada para formar el flujo de servicio como una actividad paralela o actividades secuenciales basadas en reglas de negocio, políticas, y otros requerimientos de negocio

Los procesos de negocio representan la columna vertebral del flujo de un negocio. El lado dinámico de la arquitectura de negocios está representado a través de procesos de negocio. Estos procesos de negocio solían ser implementados a través de una combinación de aplicaciones estáticas o, en el mejor de los casos, un flujo de trabajo altamente acoplado. Con la orientación a servicios, un proceso puede ser generado por composiciones de servicios (orquestración o coreografía) y la habilidad de insertar “intervención humana” y soportar transacciones de duración prolongada.

Capa de Integración.

Es una pieza clave para una SOA ya que provee la capacidad de mediar la transformación, ruteo, y conversión de protocolos para las solicitudes de transportación de servicios desde el solicitante de servicios hasta el proveedor de servicios correcto. En consecuencia esta capa soporta las capacidades requeridas para habilitar a SOA como soporte de protocolo y conversión, ambiente de comunicación en el estilo mensaje/interacción, soporte para ambientes heterogéneos, adaptador; como herramienta de interacción de servicios, de ruteo, de habilitación de servicios, virtualización de servicios, servicios de mensaje, procesamiento de mensaje y transformación.

Esta es la capa en la SOA RA que soporta la integración con plataformas de solución provistas por otras capas en la SOA RA utilizando “adaptadores” el acceso de servicios por otras capas, y las capacidades asociadas con el transporte de servicios. Puede ser considerada la tubería que conecta a la SOA.

Capas que soportan las capacidades transversales

Las capacidades transversales (algunas veces llamadas no funcionales o suplementarias), están centradas en el soporte (Capa de Información, Capa de Calidad del Servicio, Capa de Integración y Capa de Gobernabilidad).

Capa de Información.

La capa de información es responsable por manifestar una representación unificada del aspecto de información de una organización, basándose en la información provista por los servicios IT, aplicaciones y sistemas que resuelven necesidades de negocio y procesos, todo esto alineado con el vocabulario de negocios (glosario de términos). Esta capa incluye la arquitectura de la información, inteligencia de negocios y analíticos, consideraciones de metadatos e asegura la inclusión de consideraciones clave pertenecientes a arquitectura de información que pueden también ser utilizadas como base para la creación de analíticos e inteligencia de negocios a través de almacenes especializados de datos (datamarts) y de almacenes de datos (data warehouses). También soporta la capacidad de capa de datos de información virtualizada. Esto permite que SOA soporte consistencia de datos, y consistencia en calidad de datos.

Hay un conjunto múltiple de capacidades que la capa de información necesita soportar en la SOA RA, estas categorías son:

- **Servicios de Información:** Esta categoría de capacidades se encarga de soportar los servicios de información. Los servicios de información proveen de una forma uniforme de representar, acceder, mantener, manejar, analizar e integrar datos y contenido a través de fuentes heterogéneas de información. Existen fundamentalmente dos aproximaciones para lograrlo. La primera aproximación se enfoca en construir una vista única de datos críticos para el negocio representados por clientes, productos, ubicación y otras consideradas según el contexto; esto es, una aproximación MDM (Master Data Management) que brinde una vista única de la empresa. La segunda aproximación se enfoca en integrar la información apropiada en una manera oportuna y consistente, analizando e intentando mejorar la calidad de los datos, y asegurando la consistencia e integridad de los datos críticos del negocio. Esta aproximación es conocida como Información como servicio (IaaS Information as a Service).
- **Integración de la información:** Esta categoría de capacidades se encarga de soportar la integración de información y habilita capacidades de servicios de información.
- **Basic Information Management:** Esta categoría de capacidades se encarga de los problemas de manejo de información básica como los metadatos y el manejo de información no estructurada.
- **Information Security and Protection:** Esta categoría de capacidades se encarga de soportar los problemas de seguridad y protección.
- **Business Analytics:** Esta categoría se encarga de soportar analíticos del negocio y el monitoreo de la actividad de negocios. Permite a las organizaciones aprovechar la información para entender mejor y optimizar el desempeño de los negocios. Soporta puntos de entrada de reporte para obtener analíticos profundos y visualización, planeación, métricas estratégicas alineadas, visibilidad basada en roles, acceso basado en búsqueda y acciones de alerta y detección instantáneos.

Capa de Calidad de Servicio.

Esta capa provee soluciones de manejo de calidad de servicio de varios aspectos, tales como la disponibilidad, confiabilidad y seguridad así como mecanismos para soportar, seguimiento, monitoreo, y control de soluciones de calidad de servicio.

La capa de calidad de servicio provee a los procesos de ciclo de vida de servicios y de la solución SOA las capacidades requeridas para asegurar que las políticas definidas, requerimientos no funcionales y regímenes de gobernabilidad estén dentro de rangos aceptables.

Esta capa soporta monitoreo y captura de servicios y métricas de solución en un sentido operacional y señalizando el no cumplimiento con requerimientos no funcionales relacionados a destaca cualidades y políticas asociadas con cada capa SOA. Las métricas de servicio son capturadas y conectadas con servicios individuales para permitir a los consumidores del servicio evaluar el desempeño del servicio creando niveles de confiabilidad incrementados.

Capa de Gobernabilidad.

Provee un punto central en el cual las políticas son establecidas en registros y repositorios. En general las funcionalidades de gobernabilidad y procesos son administradas y ejecutadas centralmente a través de esta capa. Esta es la capa subyacente para todas las demás capas funcionales y transversales de la Arquitectura de Referencia SOA [4].

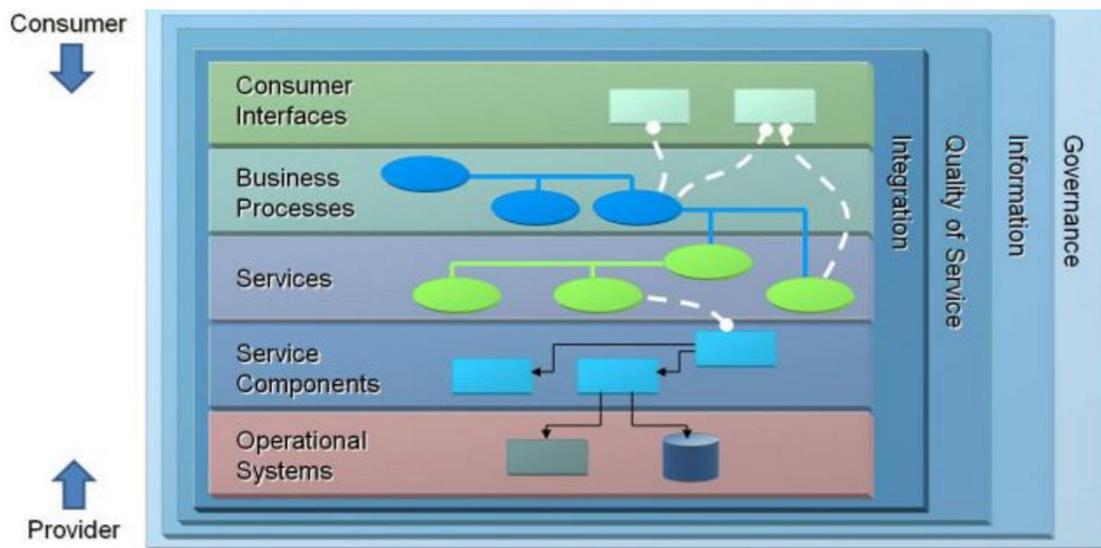


Imagen C1-1 Vista de la Solución Lógica de la Arquitectura de Referencia SOA²

La imagen C1-1 muestra una SOA como un conjunto de capas lógicas. Es importante notar que una capa no solamente depende de la capa que está debajo de ella. Debido a esto, SOA es considerada una arquitectura parcialmente en capas. Un consumidor puede acceder a la Capa de Proceso de

² Vista de la Solución Lógica de la Arquitectura de Referencia SOA. Imagen Recuperada de The OpenGroup "SOA Reference Technical Standard: Overview of the SOA RA Layers", http://www.opengroup.org/soa/source-book/soa_refarch/layers.htm

Negocio o a la Capa de Servicios directamente sin rebasar las restricciones del estilo arquitectónico. Por ejemplo, una solución SOA dada puede excluir una capa de Proceso de Negocio y permitir que la capa de Consumidor interactúe directamente con la capa de servicio. Tal solución no se beneficiaría del valor de las consideraciones del negocio; sin embargo, este valor podría lograrse en una etapa posterior agregando la capa de Proceso de Negocio. En este sentido la Arquitectura de Referencia SOA representa SOA con una arquitectura parcialmente en capas. El grado al cual una organización logra la Arquitectura de Referencia SOA diferirá acorde al nivel de madurez SOA que ésta exhiba y los requerimientos subyacentes de la organización.

1.1.2 Bloques de Construcción SOA

Cada bloque de construcción SOA puede jugar uno o dos roles existentes:

Proveedor de servicio

El proveedor de servicio crea un servicio web y publica su interfaz e información de acceso en el registro de servicios. Cada proveedor debe decidir cuáles servicios exponer, cómo mantener el equilibrio entre seguridad y disponibilidad, cómo cobrar el servicio, o (si no aplican cargos), cómo explotarlos para obtener otro tipo de beneficio. El proveedor también debe decidir en qué categoría debería ser listado el servicio para cierto agente de servicio específico y qué tipo de acuerdos entre socios comerciales son requeridos para usar el servicio.

El proveedor de servicio también registra qué servicios están disponibles dentro de él, y lista a todos los destinatarios potenciales del servicio.

Quién implementa el agente de servicio decide el ámbito del agente. También decide la cantidad de información expuesta en un servicio.

Agente de Servicio

También es conocido como el registro de servicio. Su principal funcionalidad es asegurar que la información referente al Servicio Web esté disponible para cualquier potencial solicitante. Quien quiera que implemente el agente, decide el ámbito del agente. Los agentes públicos están disponibles a través de internet, mientras que los agentes privados son accesibles únicamente a una audiencia limitada, por ejemplo, usuarios de una Intranet.

Solicitante del servicio / Consumidor

Localiza los registros en el agente de registro utilizando operaciones de búsqueda. Después, se vincula con el proveedor de servicio para invocar uno de sus servicios web y entonces poder utilizarlo.

1.1.3 El costo de una SOA y el retorno de Inversión

En el ámbito de las grandes empresas un proyecto SOA puede ser multimillonario. En la industria una de las aproximaciones más utilizadas para la estimación de costos para una implementación SOA se lleva a cabo a través de Linthicum [8] el cual establece directivas para la estimación de una SOA. Acorde a estas directivas el cálculo del costo de una SOA puede ser expresado como una suma de varios procedimientos de análisis de costos:

Costo de SOA = (Costo de complejidad de los datos
+ Costo de la complejidad de los servicios
+ Costo de la complejidad del proceso
+ Requerimientos Tecnológicos)

Además, Linthicum también provee algunas especificaciones detalladas. Por ejemplo, la tecnología de almacenamiento básico de los datos aparece con un porcentaje variable según el método utilizado (Relacional 30%, Orientada a Objetos 60% e ISAM (Indexed Sequential Access Method) 80%). El principal problema con Linthicum es que esta aproximación no es una métrica real.

Li y Keung[9] proponen un marco de trabajo que utiliza el método Divide y Vencerás (Divide-and-Conquer) en un intento para lidiar con el problema de la estimación del costo del desarrollo de software basado en SOA. Para ello clasifican a los servicios en los tipos primitivo y combinado.

Los servicios primitivos son manipulados como subproblemas que son lo suficientemente pequeños e independientes para ser resueltos. Los servicios combinados por su parte requieren de una división recursiva hasta que todos los servicios resultantes sean primitivos.

1.2 Servicios Web

No obstante que SOA fue propuesta por primera vez en 1996 por Roy W. Schulte y Yefim V. Natis [5], no fue sino hasta el año 2000 cuando el interés en SOA revivió con el surgimiento de la tecnología de servicios Web. El W3C(World Wide Web Consortium) define un servicio Web como un sistema de software diseñado para soportar interacción máquina – máquina sobre una Red[10].

Un servicio Web debería contar con una descripción de su interfaz y debería ser invocado en la manera especificada en dicha descripción utilizando protocolos relacionados con HTTP (Hypertext Transfer Protocol). Acorde con las recomendaciones de W3C, se debe utilizar WSDL(Web Service Definition Language) para describir su interfaz, y SOAP(Simple Object Access Protocol) sobre HTTP para invocarlo. Todas estas especificaciones (WSDL, SOAP, y otras relacionadas con la Web) están basadas en XML (Extensible Markup Language).[10]

1.2.1 Propiedades de los servicios Web

Las propiedades de los servicios Web se listan a continuación:

Los servicios son autocontenidos. Esto significa que no se requiere ningún software adicional para su funcionamiento: del lado del cliente es suficiente un lenguaje de programación con soporte para XML/HTML y del lado del servidor un servidor web y un servidor SOAP.

Débilmente acoplados: El cliente y el servidor se comunican a través de mensajes, un nivel de coordinación simple que permite que los sistemas basados en servicios Web sean fácilmente escalables.

Independientes del lenguaje e interoperables: cliente y servidor quizá puedan estar implementados en diferentes ambientes y en diferentes lenguajes.

Deben tener la capacidad de ser compuestos en otros servicios: Los servicios Web pueden ser combinados utilizando la tecnología de flujos de trabajo para desempeñar funciones de negocio de niveles más altos.

Es importante mencionar que el concepto de servicio no está ligado a ninguna tecnología en concreto, los servicios Web son simplemente una tecnología que permite que SOA pueda implementarse. Los principios para SOA descritos previamente proveen una guía para la implementación de servicios Web y los diferencian de las tecnologías competidoras tales como CORBA y DCOM [11]. Aunque SOA no necesariamente debe ser construida con servicios Web y la tecnología de servicios Web no es equivalente a SOA, hay una estrecha relación entre estos dos conceptos. [12]

1.2.2 Arquitectura de los servicios Web

El fundamento de la tecnología de servicios Web está representado por la interacción entre el cliente del servicio, el registro de servicios y el proveedor del servicio.

El registro de servicios actúa como un motor de búsqueda de servicios, cada proveedor de servicios registra en él su servicio brindando información sobre la función que realiza, la calidad de servicio, la información del proveedor, etcétera. El registro de servicios también categoriza los servicios basándose en criterios distintos para facilitar su localización. Los clientes de los servicios buscan el servicio que quieren y obtienen la referencia desde el registro de servicios.

Actualmente UDDI (Universal Description, Discovery, and Integration) [10] es el estándar designado para el registro de servicios, las interacciones completas entre estos tres participantes se describen a detalle a continuación:

Los proveedores de servicios envían un archivo WSDL a UDDI. El solicitante del servicio contacta a UDDI para averiguar quién es el proveedor de los datos que éste necesita, UDDI regresa al cliente el WSDL del servicio buscado, el cliente entonces contacta al proveedor del servicio utilizando el protocolo SOAP. El proveedor del servicio valida la solicitud y envía datos estructurados en un archivo XML en respuesta al cliente utilizando el protocolo SOAP. Este archivo XML será validado nuevamente por el solicitante del servicio utilizando un archivo XSD, todo este proceso está ilustrado en la figura C1-3.



1.2.3 Estándares relacionados con servicios Web.

La figura C1-4 describe la arquitectura en capas de los estándares relacionados con la tecnología de servicios Web:

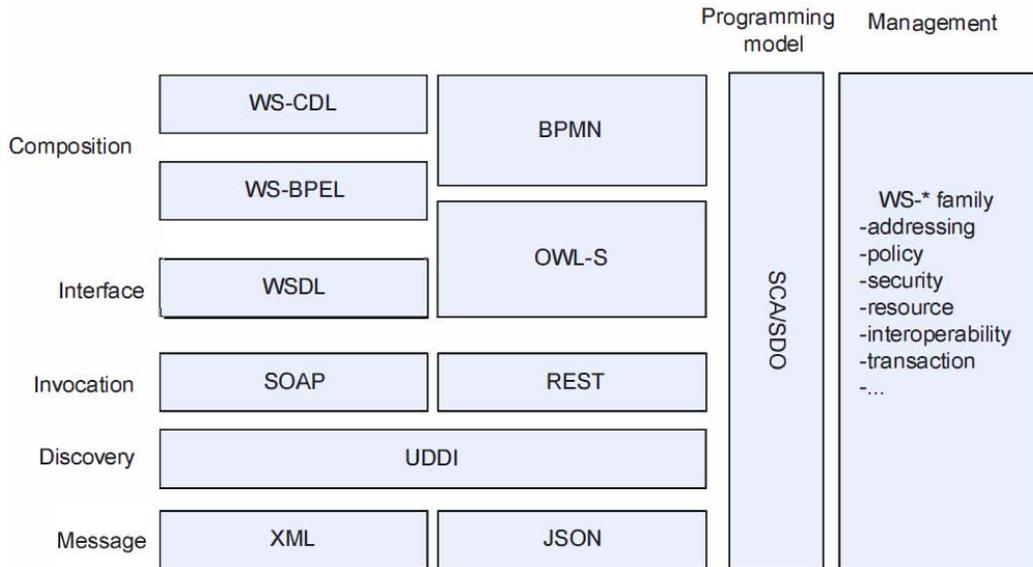


Imagen C1-4 Arquitectura de los estándares relacionados con Servicios Web.³

Capa de mensaje

La capa de mensaje se encarga de establecer los formatos de los mensajes intercambiados entre los servicios Web, y entre servicios y clientes. XML y JSON son los dos principales estándares utilizados en esta capa, ambos formatos son descritos a continuación:

XML

XML (Extensible Markup Language) [10] es comúnmente utilizado como el formato de intercambio de mensajes para las interfaces de los servicios Web. XML fue diseñado para transportar y almacenar datos. Utiliza etiquetas para organizar elementos, contenidos y atributos. La imagen C1-5 muestra la representación XML de una pieza de información que describe a un estudiante, la etiqueta <ntelefonico> también incluye el atributo “tipo” que especifica al tipo de número telefónico proporcionado, en este caso *domicilio*.

```

<estudiante>
  <cuenta>9982323</Cuenta>
  <nombre>Fernando</nombre>
  <apellido>Franco</apellido>
  <ntelefonico tipo="domicilio">55-26-45-39-87</ntelefonico>
</estudiante>

```

³ Tan, W. Zhou, M. *Arquitectura de los estándares relacionados con los servicios Web*. Imagen Recuperada de “Business and Scientific Workflows: A Web Service-Oriented Approach”, Primera Edición.

JSON

JSON (JavaScript Object Notation) [10] gradualmente se ha ido convirtiendo en un formato común, comparado con XML, JSON es un formato ligero de intercambio de datos. JSON está construido en dos estructuras, esto es, una colección de pares nombre/valor y una lista ordenada de valores. La imagen C1-6 muestra la representación JASON del mismo estudiante representado con XML en la figura C1-5. El objeto tiene dos campos de tipo cadena para el nombre y el apellido, respectivamente, un campo de número para el número de cuenta y un objeto anidado que almacena el número de teléfono.

```
{
  "cuenta"      : "9982323"
  "nombre"     : "Fernando"
  "apellido"   : "Franco"
  "ntelefonico":
  {
    "tipo"      : "domicilio"
    "numero"   : "55-26-45-39-87"
  }
}
```

Imagen C1-6 Ejemplo de archivo JSON

Capa de descubrimiento

Esta capa se encarga del registro y publicación de servicios para que los usuarios puedan encontrarlos, por lo tanto UDDI sirve a este propósito.

Capa de invocación

Esta capa define el protocolo de especificación cuando los servicios son invocados.

Actualmente, SOAP y RESTful API son los protocolos más utilizados. A continuación se describe el estándar que se utilizó en este trabajo.

The simple Object Acces Protocol (SOAP)

SOAP es un estándar para enviar mensajes y hacer llamadas a procedimientos remotos a través de internet. Es independiente del lenguaje de programación, sistema operativo y plataforma. Utiliza HTTP como protocolo de transporte y XML para la codificación de los datos.

SOAP define dos tipos de mensajes, "Request" y "Response", para permitir a los solicitantes de servicios "solicitar" un procedimiento remoto y para que los proveedores de servicios puedan "responder" a esas solicitudes.

Un mensaje SOAP consiste en dos partes, una cabecera y la carga XML.

La parte XML de una solicitud SOAP consiste en tres porciones principales:

Envelope (sobre). Define los espacios de nombres que son utilizados por el resto del mensaje SOAP.

Header (Encabezado). Es un elemento opcional para llevar información auxiliar para autenticación y transacciones.

Body (Contenido). Representa la carga principal del mensaje. Cuando SOAP es utilizado para desempeñar una llamada a un procedimiento remoto (Remote Procedure Call RPC), el contenido incluye un elemento simple que a su vez contiene el nombre del método, los argumentos y la dirección del servicio Web objetivo.

Una respuesta SOAP es entregada como un documento XML dentro de una respuesta HTML estándar.

A continuación se muestra un ejemplo de solicitud SOAP enviada sobre HTTP al servicio "CargadeCatalogo" que indica que la solicitud está siendo enviada a la URL <http://ejemplo.com/CargadeCatalogo>. La solicitud también especifica que la operación ObtenPreciodeMenudeo será ejecutada utilizando el símbolo RNG, esto significa que se obtendrá el precio de menudeo del artículo Revista NationalGeographic.

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:ObtenPreciodeMenudeo xmlns:m="http://ejemplo.com/CargadeCatalogo.xsd">
      <symbol>RNG</symbol>
    </m: ObtenPreciodeMenudeo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Diagram illustrating the SOAP request structure. The XML is annotated with red dashed boxes and brackets:

- A bracket on the right groups the namespace declarations (`xmlns:SOAP-ENV` and `SOAP-ENV:encodingStyle`) with the text "Espacios de nombres utilizados por el mensaje."
- A bracket on the right groups the `<m:ObtenPreciodeMenudeo>` element with the text "Carga principal del mensaje."

La respuesta SOAP contendrá el precio de menudeo de RNG de 10:

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m: ObtenPreciodeMenudeoRespuesta xmlns:m="
      http://ejemplo.com/CargadeCatalogo.xsd">
      <Price>10</Price>
    </m: ObtenPreciodeMenudeoRespuesta>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Diagram illustrating the SOAP response structure. The XML is annotated with red dashed boxes and brackets:

- A bracket on the right groups the namespace declarations (`xmlns:SOAP-ENV` and `SOAP-ENV:encodingStyle`) with the text "Espacios de nombres utilizados por la respuesta."
- A bracket on the right groups the `<m: ObtenPreciodeMenudeoRespuesta>` element with the text "Respuesta SOAP: Precio Menudeo del artículo RNG = 10"

Capa de Interfaz

La capa de interfaz se encarga de la especificación de la interfaz de los servicios. Ésta describe como puede ser llamado un servicio, qué estructura de datos espera, y qué estructura regresa. WSDL es el estándar en esta capa.

The web services description language (WSDL)

Para que una aplicación utilice un servicio Web, la interfaz programática del servicio Web debe ser descrita con precisión. WSDL es una gramática XML para especificar propiedades de un servicio Web tales como qué hace, dónde está localizado y cómo es invocado.

Un documento WSDL utiliza los siguientes siete elementos en la definición de servicios Web:

- *Type*. Un contenedor para definiciones de tipos de datos.
- *Message*. Una definición abstracta y con un tipo asignado de los datos que serán comunicados.
- *Operation*. Una descripción abstracta de una acción soportada por el servicio.
- *Port Type*. Un conjunto abstracto de operaciones soportadas por uno o más extremos.
- *Binding*. Un protocolo concreto y la especificación del formato de los datos para un tipo de puerto en particular.
- *Port*. Un extremo simple definido como la combinación de un binding (vínculo) y una dirección de red.
- *Service*. Una colección de extremos relacionados.

A continuación se muestra un ejemplo que describe un servicio Web que recibe la solicitud de precio de menudeo para un artículo que forma parte del catálogo de Ebay:

```
<types>
  <schema targetNamespace="http://ejemplo.com/CargadeCatalogo.xsd"
    xmlns="http://www.w3.org/1999/XMLSchema">
    <element name="ObtenPreciodeMenudeo">
      <complexType>
        <all>
          <element name="symbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="ObtenPreciodeMenudeoRespuesta">
      <complexType>
        <all>
          <element name="Price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>

<message name="ObtenPreciodeMenudeoInput">
  <part name="body" element="xsd1:ObtenPreciodeMenudeo"/>
</message>
<message name="ObtenPreciodeMenudeoOutput">
  <part name="body" element="xsd1:ObtenPreciodeMenudeoRespuesta"/>
</message>
```

Se declara el elemento symbol de tipo String para el elemento ObtenPreciodeMenudeo.

Se declara el elemento Price de tipo float para el elemento ObtenPreciodeMenudeoRespuesta.

Se establece la definición abstracta y tipeada de los datos que serán comunicados.

```

<portType name="ObtenPreciodeMenudeoPortType">
<operation name="ObtenPreciodeMenudeo">
<input message="tns:ObtenPreciodeMenudeoInput"/>
<output message="tns:ObtenPreciodeMenudeoOutput"/>
</operation>
</portType>
<binding name=" ObtenPreciodeMenudeoSoapBinding" type="tns: ObtenPreciodeMenudeoPortType">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/
soap/http"/>
<operation name="ObtenPreciodeMenudeo">
<soap:operation soapAction=" http://ejemplo.com/CargadeCatalogo.xsd"/>
<input>
<soap:body use="encoded" namespace=" http://ejemplo.com/CargadeCatalogo.xsd"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</input>
<output>
<soap:body use="encoded" namespace=" http://ejemplo.com/CargadeCatalogo.xsd"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</output>
</operation>
</binding>
<service name=" ObtenPreciodeMenudeoService">
<documentation>Mi servicio de precios</documentation>
<port name="ObtenPreciodeMenudeoPort" binding="tns:ObtenPreciodeMenudeoBinding">
<soap:address location="http://www.ebay.com/CargadeCatalogo"/>
</port>
</service>

```

Se declara un puerto (endpoint), dentro de él, se declara una operación y se definen los parámetros de entrada y salida para dicha operación

Se declara un enlace (binding) en formato SOAP.

Se define el servicio con la combinación de puertos relacionados.

Se define un puerto con la combinación de un enlace y una dirección de red.

Este documento define que el mensaje de solicitud es de tipo cadena, y el tipo de la respuesta es flotante, el nombre de la operación que debe ser invocada es "ObtenPreciodeMenudeo", HTTP es usado para llevar los mensajes los cuales están en formato SOAP y la dirección de red del servicio es http://www.ebay.com/CargadeCatalogo.

Capa de Composición.

La aproximación orientada a servicios de SOA requiere un lenguaje para realizar una descripción relativamente simple de cómo pueden componerse los servicios Web convirtiéndose en procesos de negocio y cómo éstos pueden ser ejecutados. Esto permite no sólo proporcionar definiciones de procesos abstractos, sino escribir especificaciones ejecutables de dichos procesos. WS-BPEL(Web Services Business Process Execution Language) es el estándar de facto de la industria para componer múltiples servicios web en procesos de negocio y para exponer estos procesos como un servicio compuesto. BPMN(Business Process Model and Notation) es una especificación popular de procesos de negocio, y tiene una relación muy cercana con la composición de servicios. WF (Workflow Foundation) es el marco de trabajo de Microsoft para la composición de servicios web en procesos de negocio.

Orquestación de Servicios

La combinación de varios servicios Web realmente implica la integración de las aplicaciones subyacentes y sus funcionalidades, los servicios web pueden combinarse de dos formas:

- Orquestación
- Coreografía

Cuando se hace uso de **orquestación**, un proceso central (que puede ser otro servicios Web) lleva el control de los servicios Web implicados en la realización de una tarea y coordina la ejecución de las diferentes operaciones sobre dichos servicios Web. Los servicios Web orquestados no “conocen” (y no necesitan conocer) que están implicados en un proceso de composición y que forman parte de un proceso de negocio de nivel más alto. Solamente el coordinador central de la orquestación es “consciente” de la meta a conseguir, por lo tanto la orquestación se centraliza mediante definiciones explícitas de las operaciones y del orden en el que se deben invocar los servicios Web [13].

En la coreografía no existe un coordinador central por lo tanto todos los participantes necesitan estar “informados” del proceso de negocio, esto convierte a la coreografía en un proceso mucho menos flexible.

Web Services Business Process Execution Language (WS-BPEL)

WS-BPEL es un lenguaje de orquestación de servicios. Es un lenguaje basado en XML que soporta tecnologías de servicios Web (incluyendo SOAP, WSDL, UDDI entre otras).

WS-BPEL puede utilizarse dentro de una empresa y entre empresas. Dentro de una empresa, el papel de BPEL es el de estandarizar la integración de las aplicaciones y extender la integración entre sistemas previamente aislados. Entre empresas, BPEL permite una integración más fácil y efectiva con partners del negocio.

Business Process Model and Notation (BPMN)

BPMN provee una notación gráfica intuitiva y uniforme para modelos de procesos de negocio. Puede ser usada como un lenguaje común entre múltiples interesados en el proceso (stakeholders) para compartir el conocimiento de los procesos de negocio de una organización. Los analistas de negocio pueden utilizarlo para expresar los objetivos y funcionalidades que deben ser alcanzadas; los expertos en TI pueden utilizarlo para implementar un flujo de trabajo que automatiza un proceso; los administradores pueden utilizarlo para monitorear la ejecución y examinar el desempeño de los negocios. El mapeo de BPMN a BPEL es también parte de esta especificación. Contiene también un modelo de colaboración para describir coreografías de procesos.

WF (Workflow Foundation)

Windows Workflow Foundation permite crear flujos de trabajo que utilizan servicios Web para darle soporte a procesos de negocio.

Capas transversales: Modelo de programación y Administración.

Las capas transversales (cross-layer) son: el modelo de programación y la Administración (Refiérase a figura C1-1 de la sección 1.2.3).

Modelo de Programación. Este modelo es independiente de cualquier lenguaje de programación específico, define como abstraer funciones como componentes y utilizarlos como bloques de construcción para el ensamble de soluciones SOA. Actualmente, la arquitectura de componente de Servicio (SCA) es un modelo de programación SOA muy popular; mientras que el Objeto de Datos de Servicio (SDO) es su modelo de datos.

Administración. La Administración de servicios por su parte, abarca muchas especificaciones no funcionales y también a las llamadas características transversales (cross cutting), tales como el direccionamiento, políticas, seguridad, recursos, interoperabilidad y transacción.

1.3 La tecnología de flujos de trabajo.

El Workflow Management Coalition (WfMC) define el término flujo de trabajo como la facilitación computarizada o automatización de un proceso de negocio completo o en parte.

El Modelo de Referencia de Flujo de Trabajo [14] publicado por WfMC en 1995 ha establecido las bases definiendo un vocabulario y arquitectura de un sistema de workflow. A pesar de la drástica evolución de la tecnología a lo largo de los años, este modelo de referencia todavía es aplicable a la mayoría de los sistemas actuales de flujos de trabajo.

El modelo de referencia establece un glosario común para describir procesos de negocios (también conocidos como flujos de trabajo) y varios artefactos asociados con ellos. Divide la función de un sistema de administración de flujos de trabajo en dos aspectos, tiempo de construcción y tiempo de ejecución.

En el tiempo de construcción, se establece un flujo de trabajo, generalmente con la ayuda de una herramienta de modelado, éste representa el proceso de negocio a ser automatizado.

La definición de un flujo de trabajo contiene un conjunto de actividades y la secuencia entre ellas. Una actividad pueden ser manual (que necesita la intervención humana) o bien automatizada (ejecutada por una aplicación de software).

En tiempo de ejecución, un motor de flujo de trabajo es el software que provee el ambiente de ejecución para los flujos de trabajo. El motor además brinda la posibilidad de comenzar múltiples instancias de un flujo de trabajo, por ejemplo: el flujo de trabajo para el registro de inscripción de un alumno puede tener múltiples instancias, cada una representa a un alumno en particular. Puede haber muchas instancias concurrentes en un motor de flujo de trabajo. En una instancia de flujo de trabajo, cada actividad también tiene su correspondiente instancia de actividad [12].

1.3.1 Estándares relacionados con flujos de trabajo

De la misma manera en que se describieron los estándares para servicios web en la sección 1.2.3, ahora se presentan los estándares relacionados con flujos de trabajo.

El más importante es el modelo de referencia WfMC, el cual ha establecido los fundamentos en la definición de la estructura de un sistema de flujo de trabajo. El modelo de referencia primero define un glosario general para ser utilizado en sistemas de flujo de trabajo. El modelo de referencia también incluye una arquitectura de referencia. En dicha arquitectura, el componente núcleo de un sistema de flujo de trabajo es uno o más motores de flujos de trabajo, cada motor tiene cinco interfaces interactuando con otros componentes de nivel más alto en el sistema. La imagen C1-7 ilustra el modelo de referencia de WfMC. A continuación se define qué es un motor de flujo de trabajo y cómo éste interactúa con otros componentes vía cinco interfaces.

Un motor de flujos de trabajo es el software para crear, ejecutar, y administrar instancias de flujos de trabajo, y facilitar sus interacciones con humanos y aplicaciones automáticas. En resumen, un sistema de flujo de trabajo consiste de uno o más motores de workflow que conforman los servicios de activación de flujos de trabajo.

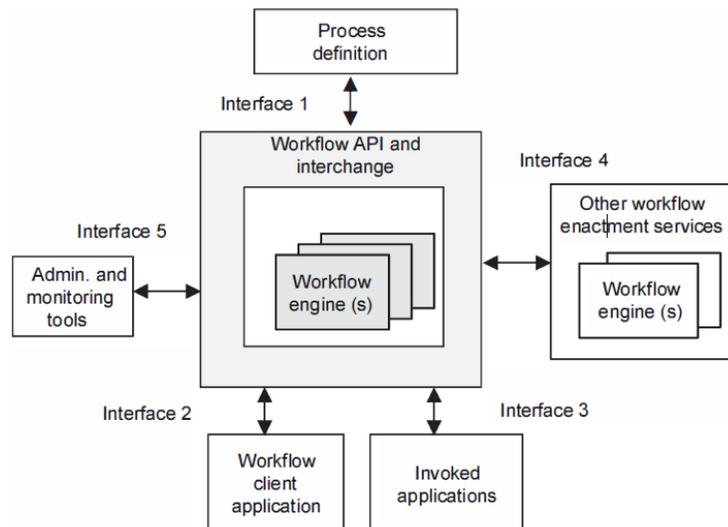


Imagen C1-7 El modelo de referencia de WfMC⁴

El servicio de activación interactúa con otros cinco componentes a través de cinco interfaces como se ilustra en la imagen C1-7.

Las cinco interfaces se describen a continuación:

Interface 1: Definición de Proceso

La interface 1 define el formato de intercambio para la definición de un proceso, y la API para obtenerlo. Esta es la separación lógica entre tiempo de construcción y tiempo de ejecución. Los usuarios pueden usar cualquier herramienta para construir la definición de un proceso que puede

⁴ Tan, W. Zhou, M. *Modelo de referencia de WfMC*. Imagen Recuperada de "Business and Scientific Workflows: A Web Service-Oriented Approach", Primera Edición.

ser recuperado por un servicio de promulgación a través de la interface 1 por medio de transferencia de archivo o API. WfMC ha establecido un meta-modelo y especificación de definición de proceso que es XML Process Definition Language (XPDL). XPDL es similar a BPEL, BPMN y XAML, aunque cada uno tiene sus propias características. El meta-modelo para un proceso incluye los siguientes elementos:

- Actividad, los eventos de inicio y fin, los participantes y las aplicaciones para realizarlos.
- Compuertas, tales como AND/OR estructuras de partición y unión.
- Calles, para distribuir un proceso acorde a diferentes participantes.
- Datos relevantes del flujo de trabajo que son utilizados en cada instancia de flujo de trabajo en tiempo de ejecución.

Interface 2: Aplicación cliente del flujo de trabajo

Es la estrategia a seguir con actividades manuales en un flujo de trabajo. La interface de aplicación cliente permite a los participantes recuperar las tareas junto con la información relacionada, terminar la tarea y proporcionar los resultados de la ejecución.

Interface 3: Aplicaciones invocadas

Es la estrategia a seguir con actividades automatizadas en un flujo de trabajo. Esta interface permite al motor invocar una aplicación cuando una instancia de flujo de trabajo alcanza una actividad automatizada. Hay varios protocolos para invocar una aplicación remota, la interface de un servicio Web se está convirtiendo en el protocolo predominante entre todos ellos.

Interface 4: Otros servicios de activación de flujos de trabajo

Los flujos de trabajo de diferentes sistemas pueden necesitar establecer comunicación entre ellos. Por ejemplo, un flujo de trabajo de procesamiento de una orden de compra en línea como Ebay, puede necesitar comunicarse con un flujo de trabajo de pago de un proveedor de pagos como PayPal. Debido a ello existe la necesidad de definir una interface para servicio de activación de flujos de trabajo múltiples para que puedan colaborar entre sí.

La interface 4 define como dos servicios de activación pueden ser configurados para establecer comunicación, transferir una definición de proceso, delegar la ejecución de un subproceso, y transferir datos entre ellos.

Interface 5: Herramientas de Administración y Monitoreo

La interface 5 define las interfaces de administración y monitoreo tales como la administración de usuarios, auditoría y control de recursos.

1.4 Estado del Arte de la composición de servicios.

Esta sección recorre el trabajo relacionado con la composición de servicios, categoriza dicho proceso en cinco áreas, estas son: composición automática de servicios, composición de servicios apoyada por mediación, verificación de flujos de trabajo basados en servicios, soporte para ejecución descentralizada y flujos de trabajo científicos.

1.4.1 Composición automática de servicios

La composición y reusabilidad están incluidos entre los ocho principios de SOA. Los servicios necesitan ser utilizados por distintos usuarios, de otra manera no necesitan ser expuestos como servicio, y los servicios deben tener la capacidad de componer otros servicios para poder ser reutilizados. Los métodos para componer servicios de manera automática o semiautomática juegan un papel importante en SOA, debido al gran número de servicios candidato y a la complejidad que es requerida para desempeñar tal composición.

Composición automática de servicios es una tarea compleja que ha inspirado una variedad de técnicas que intentan resolver el problema. Las dos técnicas de automatización de mayor importancia son: sistensis (synthesis) y planeación (planning). Esta última se describe con detalle.

1.4.2 Planeación (planning)

Es una rama de la Inteligencia Artificial, está basada en servicios Web Semánticos con descripciones de propiedades y capacidades de los servicios, y un mecanismo de razonamiento para seleccionar y agregar servicios.

La composición automática atrae la mayor parte de la atención en la investigación sobre SOA debido a su importancia. Sin embargo muchas de las aproximaciones son teóricas y generalmente requieren de un esfuerzo substancial para construir un modelo para cada candidato de servicio. Por ejemplo, el lenguaje DSOL (Declarative Service Orchestration Language) propuesto por Cugola y otros autores en 2012 el cual está asociado con un motor ad-hoc que utiliza la planeación para soportar composiciones de servicios auto-adaptativos en tiempo de ejecución [15].

1.4.3 Composición de Servicios Ayudados por Mediación (Mediation-Aided Service Composition)

En la vida real la composición de servicios Web enfrenta un fenómeno común, la compatibilidad parcial entre servicios, esto es, dos o más servicios proveen funcionalidad complementaria; sin embargo, sus patrones de interacción no ajustan exactamente y por ello no pueden ser directamente compuestos. La Mediación envuelve servicios heterogéneos para que puedan parecer homogéneos y por consecuencia sea más fácil integrarlos.

1.4.4 Verificación de flujos de trabajo basados en servicios.

Para Flujos de trabajo de gran escala es esencial hacer un análisis del flujo de trabajo, conocido como verificación, después de que el modelado del flujo de trabajo está terminado. La verificación puede ayudar a los diseñadores a analizar si las definiciones del flujo de trabajo son correctas, y entonces hacer modificaciones si es necesario.

Durante el proceso de verificación de flujos de trabajo puede analizarse un conjunto de características tales como la accesibilidad (reachability) y la existencia de puntos muertos (deadlocks) [16]. Accesibilidad significa verificar si el modelo de flujo de trabajo puede alcanzar cualquier estado específico como el resultado de una ejecución requerida de servicios. La búsqueda de puntos muertos (deadlock) se refiere a la situación en la cual una instancia de flujo de trabajo cae en un estancamiento y ninguna actividad puede ejecutarse. Un punto muerto puede ocurrir cuando un set de servicios atómicos son ejecutados concurrentemente y comparten un conjunto de recursos. Sin embargo, si no existe un control adecuado durante la ejecución,

puede aparecer un conflicto de acceso a un recurso y con ello impedir que los servicios atómicos puedan progresar debido al hecho de que los recursos críticos que necesitan no están disponibles.

La verificación se puede extender a los modelos de métricas de calidad de servicio, tales como tiempo de ejecución, costo, seguridad, nivel de dependencia y desempeño.

Esta verificación comúnmente se realiza a través de redes de Petri, las cuales como herramientas gráficas y matemáticas, proveen un ambiente uniforme de desarrollo para el modelado, análisis y diseño de sistemas de eventos discretos. Las redes de Petri han sido extensamente utilizadas en el análisis de flujos de trabajo. Ayudan a identificar actividades inalcanzables y acciones que puedan generar conflictos, permiten el análisis de correctés y validación, previenen puntos muertos y permiten la optimización del proceso.

Una red de Petri es un tipo particular de grafo bipartito dirigido con dos tipos de nodos llamados lugares y transiciones. Los nodos están conectados a través de arcos dirigidos. Las conexiones entre dos nodos del mismo tipo no están permitidas. Los lugares son representados por círculos y las transiciones por rectángulos.

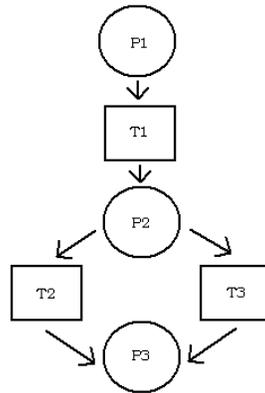


Imagen C1-8 Ejemplo Red de Petri

1.4.5 Ejecución Descentralizada de Flujos de Trabajo

Este enfoque está diseñado para organizaciones cuyos flujos de trabajo tienden a expandirse a través de las fronteras organizacionales. Como resultado, no es viable emplear un solo Sistema de Administración de Flujos de Trabajo (WFMS – Workflow Management System) centralizado para controlar la ejecución del flujo de trabajo debido a la limitada escalabilidad, disponibilidad y desempeño.

En este modelo, los flujos de trabajo son divididos en particiones llamadas flujos de trabajo autodescritos, y son manejados por un componente manejador de flujos ligero, llamado “workflow stub”, localizado en cada organización [17].

Investigadores del laboratorio hindú de IBM han establecido que las dependencias de datos y de control entre los componentes pueden ser analizados y el código puede ser particionado en componentes más pequeños para ejecutarse en localidades distribuidas [18]. En una composición descentralizada hay múltiples motores de flujo de trabajo, cada uno ejecuta una porción del

servicio compuesto original. Los motores se comunican directamente entre ellos (en lugar de utilizar un coordinador central) para transferir datos y control en una forma asíncrona cuando sea necesario.

1.4.6 Sistemas científicos de flujos de trabajo.

A diferencia de los flujos de trabajo de negocio, los flujos de trabajo científicos son utilizados para modelar la ejecución de experimentos científicos. Pueden ensamblar actividades de procesamientos de datos científicos y automatizar la ejecución de esas actividades. Los flujos de trabajo científicos se enfocan particularmente al flujo de datos y el paralelismo.

Dado que la ejecución secuencial de flujos de trabajo científicos tomaría mucho tiempo, se utilizan los sistemas manejadores de flujos de trabajo, dado que permiten la ejecución en paralelo de flujos de datos científicos y explotan los recursos distribuidos en diferentes infraestructuras tales como una malla (grid computing) y la nube (cloud computing).

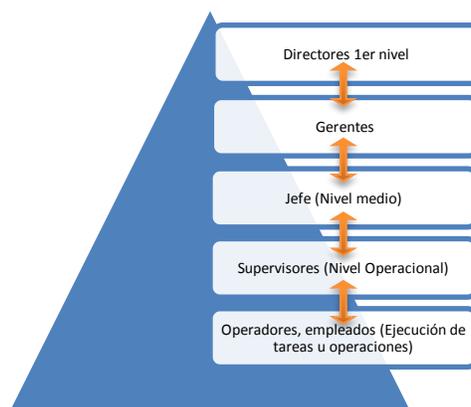
Ji Liu [19] describe una arquitectura de cinco capas para sistemas manejadores de flujos de trabajo que manejan grandes cantidades de datos, también detalla una taxonomía de técnicas de paralelización de flujos de trabajo y algoritmos de programación (Schedule) de flujos de trabajo científicos.

1.5 Estructura organizacional

La estructura organizacional aclara las funciones de los empleados, facilita la comunicación y establece una cadena de responsabilidad. La implementación de una estructura organizacional correcta es crítica para maximizar la productividad del personal. Existen dos tipos de estructuras organizacionales, esta son la estructura jerárquica (vertical) y la estructura matricial (horizontal).

1.5.1 Estructura Jerárquica

La imagen C1-9 muestra la estructura de una organización jerárquica (estructura organizativa vertical), en esta estructura cada entidad, excepto una, está subordinada a una entidad única. Los miembros de las estructuras jerárquicas principalmente se comunican con su superior inmediato y con sus subordinados inmediatos. El objetivo de este tipo de organización es reducir la sobrecarga de comunicación por medio de la limitación del flujo de información, sin embargo esta característica puede convertirse en una limitante si la comunicación entre niveles contiguos no es eficiente.



1.5.2 Estructura matricial

Es una estructura organizacional que usa tanto gerentes funcionales como gerentes de proyecto para administrar los mismos recursos, de forma que si existen N proyectos en curso en la organización, existen N gerentes de proyecto que coordinan N equipos compuestos por recursos provenientes de las áreas funcionales. El resto de los recursos permanece en su área funcional de la organización, bajo la coordinación del gerente funcional.

Al armar un equipo de proyecto existe un proceso de negociación entre el gerente del proyecto y los gerentes funcionales para asignar los recursos al proyecto. Una vez que estos recursos fueron asignados al proyecto, generalmente reportan al gerente del proyecto durante el período del proyecto, y dejan de reportar a su gerente funcional.

El grado de independencia de los gerentes de proyecto con respecto a los gerentes funcionales determina si estamos hablando de una organización matricial fuerte o débil. Fuerte: si los gerentes de proyecto tiene libertad total de decisión con respecto al uso del tiempo de los recursos del proyecto. Débil: si los gerentes funcionales tienen más peso en ese tipo de decisiones.

¿Cuáles son las ventajas de un esquema matricial?

- Más facilidad para armar un Equipo de Proyecto.
- Flexibilidad de elegir recursos más idóneos para el proyecto.
- Los proyectos se transforman en algo estándar en la organización y dejan de ser una excepción.
- Fomenta la promoción de Equipos Multidisciplinarios.
- Crea una trayectoria de carrera de Administración de Proyectos definida y reconocida dentro de la organización.
- Fomenta la proactividad en Administración de Proyectos cuando se implementa un proyecto con proveedores externos.

¿Cuáles son las desventajas de un esquema matricial?

- Los miembros de los Equipos de Proyecto tienen dos jefes durante el proyecto.
- Existe un proceso de negociación entre el Gerente de Proyecto y los Directores de Área para la asignación de recursos al proyecto.
- Debe existir ante cada proyecto un proceso de Construcción de Equipos, sin el cual el proyecto no puede comenzar.

1.5.3 Implicaciones de la estructura organizacional en una SOA

En una organización jerárquica la colaboración entre empleados y directivos en los procesos de negocio sucede en un entorno muy estructurado, debido a esta rigidez la comunicación entre los departamentos tiende a disminuir, favoreciendo con esto la existencia de sistemas aislados y la generación de datos duplicados y de mala calidad.

Otra de las implicaciones de la estructura organizacional jerárquica es la necesidad de un monitoreo constante por parte de los niveles directivos que les permita llevar a cabo una gestión efectiva del personal.

Una SOA aplicada a una organización jerárquica resuelve el problema de la existencia de sistemas aislados al exponer piezas de funcionalidad como servicios e integrarlas en procesos de negocio.

Esta integración se realiza a través de un flujo de trabajo que describe la cronología de actividades a realizar, los actores quienes las llevarán a cabo y la información que fluye a lo largo del proceso.

El aspecto de los actores que participan en un flujo de trabajo es crucial para entender la utilidad de una SOA en una organización vertical, dado que un actor representa un nivel en el organigrama y por ende un conjunto de actividades que sólo él puede realizar. En una organización matricial las decisiones dependen en el mejor de los casos de dos gerencias distintas, esto implicaría involucrar dos criterios distintos en la ejecución de un flujo de trabajo, lo que puede llevar a ambigüedades.

Lo expuesto en el párrafo anterior no implica que una SOA no pueda contribuir con la realización de procesos de negocio en una organización matricial, sólo que el enfoque debe ser distinto y debe considerar las posibles vertientes que el flujo puede tomar considerando la duplicidad de roles.

Por otro lado SOA también se encarga de proporcionar un medio de monitoreo a través del seguimiento de las diferentes instancias de un flujo de trabajo, permitiendo con ello a los directivos conocer el estado de sus procesos de negocio mientras éstos están llevándose a cabo.

En conclusión la aplicación de una SOA en una organización jerárquica facilita el flujo de información entre los distintos niveles respetando las jerarquías establecidas, garantiza la calidad, seguridad y disponibilidad de la información, favorece la realización de un monitoreo eficiente, y permite aprovechar de mejor manera las inversiones en TI al exponer en forma de servicios las funcionalidades de los sistemas existentes para orquestar procesos de negocio actuales o bien definir nuevos procesos de negocio.

Capítulo 2 El modelo Abstracto SOA de Microsoft

2. El Modelo Abstracto SOA de Microsoft

En este capítulo se describe el modelo abstracto SOA de Microsoft y se establecen equivalencias con el modelo de referencia propuesto por el OpenGroup. También se describe la plataforma tecnológica SOA de Microsoft y se seleccionan las herramientas para la propuesta de SOA que se muestra en el tercer capítulo.

2.1 Definiendo la pertinencia de SOA.

Las organizaciones tienen diferentes requerimientos y expectativas para SOA, esto se debe a que las necesidades y objetivos entre una organización y otra son muy diversos, por esta razón SOA no puede ser concebida como una metodología aplicable a todo tipo de organizaciones, si el contexto de la organización no es el adecuado la arquitectura de referencia SOA puede no proveer la mejor solución.

La pertinencia de SOA está definida por las necesidades de la organización, si ésta requiere integrar sistemas heterogéneos de software para definir alguno de sus procesos de negocio, entonces un proyecto SOA está justificado.

La figura C2-1 muestra el organigrama de una empresa en la que los sistemas de información de tres departamentos participan en un nuevo proceso de negocio.

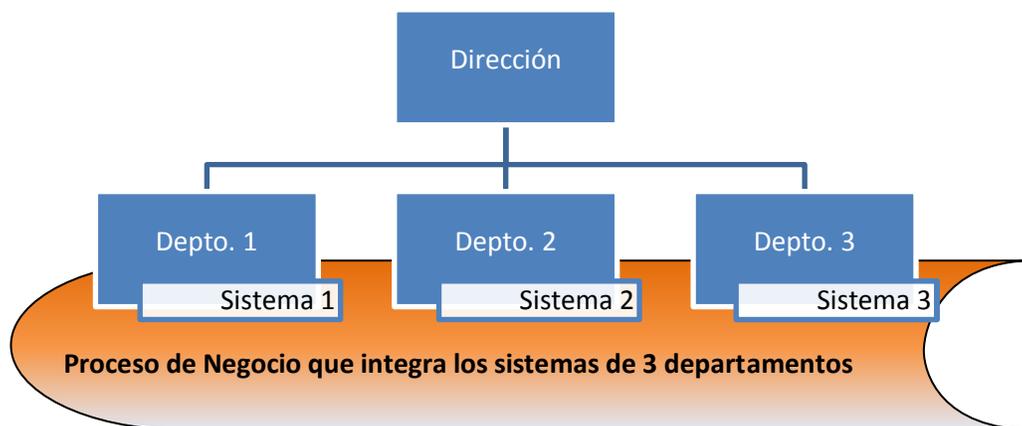


Imagen C2-1 Organigrama que integra múltiples sistemas en un Proceso de Negocio

2.2 Justificación del modelo de referencia SOA

Una implementación bien planeada y ejecutada de SOA puede contribuir con las organizaciones para que éstas alcancen una mayor capacidad de respuesta ante necesidades cambiantes (escalabilidad). Los proyectos SOA pueden ser parcialmente efectivos si la solución es propuesta sin considerar las necesidades estratégicas de la organización, esto sucede cuando quienes proponen la solución son desarrolladores que no están familiarizados con el contexto del negocio, a esto se le conoce como aproximación de abajo hacia arriba (bottom up), el resultado, una implementación poco efectiva sin relevancia para el negocio.

La contraparte es la aproximación top down, este enfoque implica conocer a fondo el contexto del negocio sin embargo, tiende a convertirse en un proyecto enorme, que para el momento en el que esté terminado ya no responde a las necesidades del negocio.

Microsoft implementa una aproximación conocida como middle out que combina la metodología bottom up y top down. En esta aproximación SOA es dirigido por una visión estratégica y por las necesidades del negocio y éstas son cubiertas a través de proyectos SOA iterativos e incrementales que están diseñados para entregar objetivos de negocio una necesidad a la vez [20].

2.3. El modelo de referencia abstracto SOA de Microsoft

En el capítulo 1 se describió la arquitectura de referencia de SOA propuesta por el OpenGroup, la cual consta de 5 capas funcionales, capa de Consumidor, capa de Procesos de Negocio, capa de Servicios, capa de Componentes de Servicios y la capa Operacional.

El modelo SOA de Microsoft no está estructurado en capas, esto se debe a que la filosofía SOA de Microsoft establece que un conjunto de capas bien definidas restringe el valor y la flexibilidad de los servicios, resultando en dependencias entre componentes no relacionados [Buscar una justificación al respecto]. El modelo abstracto de Microsoft está dividido en 3 niveles: Consumo, Composición y Exposición, cada una de estas divisiones representa una arquitectura independiente:

1. Arquitectura de Aplicación (Consumo)
2. Arquitectura de Integración de Servicios (Composición)
3. Arquitectura de implementación de Servicios (Exposición)

Estas tres arquitecturas abarcan una o más de las capas definidas en el modelo de referencia del OpenGroup, en la imagen C2-1 se describen estas equivalencias:

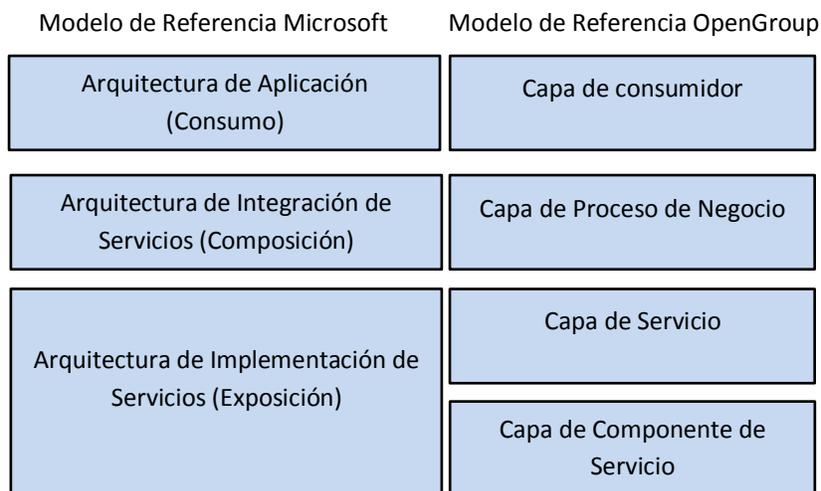


Imagen C2-1 Equivalencias entre el modelo abstracto SOA de Microsoft y el Modelo de referencia SOA del OpenGroup

Una breve definición de cada arquitectura se incluye a continuación:

Arquitectura de Implementación de Servicios (Exposición)

Esta arquitectura está centrada en cómo las inversiones en TI son expuestas como un conjunto de servicios basados en estándares y describe cómo estos servicios son desarrollados, implementados y administrados.

Arquitectura de Integración de Servicios (Composición)

Con un conjunto de servicios creados pueden hacerse combinaciones para generar servicios más complejos, aplicaciones o procesos de negocios con funcionalidades transversales. Esto es posible gracias a la independencia existente entre ellos. Esto permite que a pesar de que los procesos de negocios evolucionen, las reglas de negocio y las prácticas puedan ser ajustadas sin las restricciones de las aplicaciones existentes.

Una Arquitectura de Integración de Servicios describe un conjunto de capacidades para componer servicios y otros componentes y con ello formar construcciones más grandes tales como procesos de negocio.

Arquitectura de Aplicación (Consumo)

Cuando una nueva aplicación o proceso de negocio ha sido creado esa funcionalidad debe estar disponible para su consumo por los sistemas existentes, otros servicios o por usuarios finales.

La Arquitectura de Aplicación se enfoca en la entrega de nuevas aplicaciones resultado de la composición de servicios. Portales web, aplicaciones de escritorio, aplicaciones móviles, son algunos ejemplos de las salidas que esta arquitectura entrega.

2.4 Capacidades Arquitectónicas Recurrentes

Aunque cada una de estas arquitecturas está diseñada para ser independiente, comparten un conjunto de cinco capacidades, conocidas como Capacidades Arquitectónicas Recurrentes, las cuales se listan a continuación:

2.4.1 Mensajes y Servicios.

La capacidad arquitectónica de Mensajes y Servicios se enfoca en el concepto de orientación a servicios y en cómo diferentes servicios son utilizados para implementar SOA. El aspecto de Mensajes establece el patrón de comunicación a utilizar entre remitentes y destinatarios. Por su parte el aspecto de Servicios define las consideraciones importantes a tomar en cuenta durante la generación de los servicios, por ejemplo, determinar qué exponer y cómo, evitar el problema de la granularidad, definir la interacción entre diversos patrones, la exposición de orquestaciones como servicios, entre otras.

Granularidad de los servicios

La creación de servicios puede ser de grano simple (fine grained – un servicio que mapea a un solo proceso de negocio) o de grano grueso (coarse grained – múltiples servicios se unen para desempeñar un conjunto relacionado de funciones de negocio).

Equivalencia con la Arquitectura de Referencia SOA

Esta capacidad Arquitectónica puede compararse con la capa Sistemas Operacionales, la capa Componentes de Servicios y la capa de Servicios de la Arquitectura de referencia propuesta por el OpenGroup.

2.4.2 Flujo de Trabajo y Procesos.

La capacidad arquitectónica de Flujo de trabajo y Procesos se enfoca en el concepto de Flujo de trabajo y en cómo puede ser utilizado para orquestar y agregar servicios granulares en construcciones más grandes llamadas procesos.

Los flujos de trabajo efectivos no sólo comprenden el modelado de procesos, sino también su monitoreo y la obtención de información derivada de su ejecución, esto con el objetivo de responder de manera dinámica a la evolución de los procesos de negocio.

En esta capacidad se enmarca la creación de un Monitor de progreso de Flujos de trabajo, éste recoge información sobre la ejecución de una instancia de un flujo de trabajo para ser presentada a través de una interfaz gráfica que le permita al responsable del proceso de negocio monitorear su funcionamiento y extraer información.

Equivalencia con la Arquitectura de Referencia SOA

Esta capacidad Arquitectónica puede compararse con la capa Procesos de Negocio de la Arquitectura de referencia propuesta por el OpenGroup.

2.4.3 Datos.

El elemento unificador en el éxito de una arquitectura de integración es la habilidad para proveer un manejo óptimo de los datos. La necesidad de entregar una vista compartida a través de fuentes de datos dispares, comúnmente duplicadas es muy importante, dado que los negocios se esfuerzan en lograr un vista de 360 grados de la información organizacional. El manejo de datos maestros es un elemento crucial de una arquitectura de integración [20].

Manejo de Datos Maestros (MDM)

La mayoría de los sistemas de software tienen listas de datos que son compartidos y utilizados por varias aplicaciones que componen al sistema. Por ejemplo un ERP (Sistema de Planificación de Recursos Empresariales) tendrá un Registro Maestro de Clientes, otro de Artículos y otro de Cuenta. Esta información “maestra” representa uno de los bienes clave para cualquier organización [21].

Datos maestros son los sustantivos críticos de un negocio y pueden ser generalmente clasificados en cuatro agrupaciones: Personas, Cosas, Lugares y Conceptos. Cada una de esas agrupaciones se dividen en nuevas agrupaciones llamadas áreas de dominio o tipos de entidades. Por ejemplo, dentro de personas, existen clientes, empleados y agentes de venta. Dentro de cosas hay productos, partes, tiendas y bienes. Dentro de conceptos, hay contratos, garantías y licencias. Finalmente dentro de lugares hay ubicaciones de oficinas, y divisiones geográficas. Algunas de estas áreas de dominio pueden ser divididas en otros niveles.

MDM es la tecnología, herramientas y procesos requeridos para crear y mantener listas consistentes y precisas de datos maestros. MDM no sólo es un problema tecnológico, algunos de los retos MDM más difíciles son más políticos que técnicos, dado que en muchas ocasiones es necesario hacer cambios fundamentales en procesos de negocio para mantener datos maestros “limpios”.

Por otro lado, MDM comprende tanto la creación como el mantenimiento de los datos maestros. Invertir mucho tiempo, dinero y esfuerzo en crear un conjunto de datos maestros limpios y consistentes es un esfuerzo desperdiciado a menos que la solución incluya herramientas y procesos para mantener los datos en buenas condiciones al mismo tiempo que éstos son actualizados e incrementados.

La imagen C2-2 muestra el ejemplo de una solución de MDM en conjunción con una solución SOA, en ella el aspecto de mantenimiento de los datos maestros está resuelto a través de “data stewards” o administradores de los datos quienes son responsables de instanciar chequeos y balances de gobernabilidad de los datos en un nivel operacional para cumplir su rol en el flujo de trabajo: aprobando o rechazando cambios diarios en los datos, asegurando la precisión de los datos y brindando soporte en la generación de reportes.

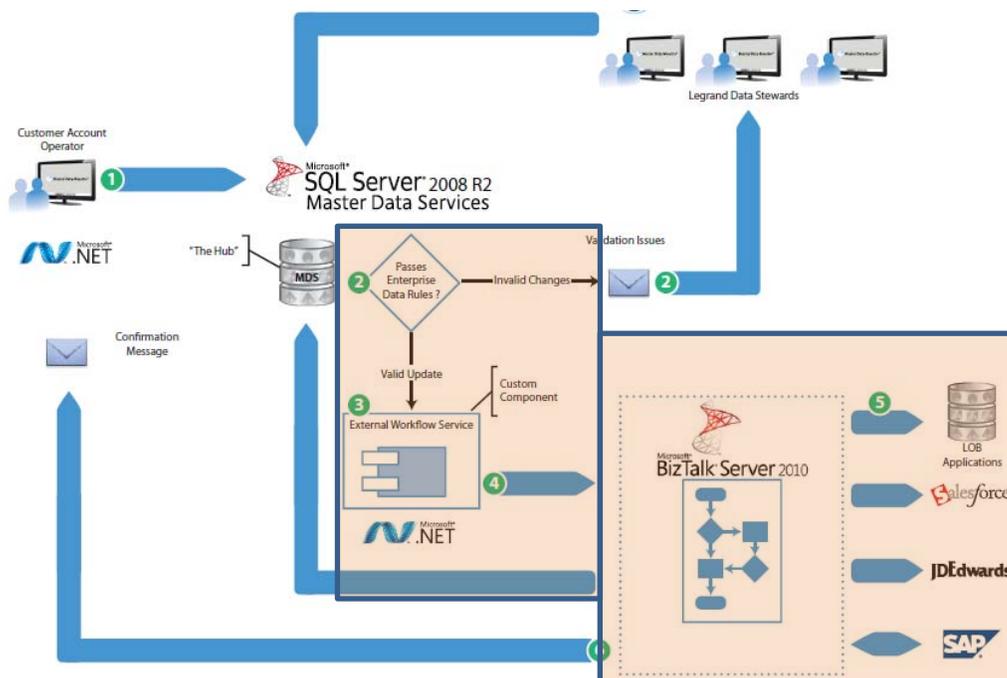


Imagen C2-2 Ejemplo de integración SOA y MDM.⁵

Las dos secciones en recuadros representan la funcionalidad SOA de la solución, en primer lugar un flujo de trabajo que verifica si los cambios en la base son válidos para posteriormente ser transformados en un servicio de flujo de trabajo externo que es enviado a un agente de

⁵ “Microsoft Success Stories, Legrand North America”. Imagen Recuperada de <http://www.blumshapiro.com/case/master-data- case-study-legrand-north-america-manages-enterprise-data>

integración (BizTalk Server⁶), este último provee un adaptador que permite a los usuarios escribir datos en los sistemas subscriptores sin tener que lidiar con los detalles técnicos de la forma en la que cada sistema procesa la información.

Equivalencia con la Arquitectura de Referencia SOA

Esta capacidad Arquitectónica puede compararse con la capa transversal Arquitectura de la Información de la Arquitectura de referencia propuesta por el OpenGroup.

2.4.4 Experiencia de usuario.

El consumo de servicios debe mapearse de manera natural con el flujo de trabajo de empleados, clientes y socios. Para lograr ese objetivo una experiencia de usuario integrada que se extiende a aplicaciones web y dispositivos móviles hacen posible que el consumo de servicios tenga la audiencia lo más amplia posible [20].

Equivalencia con la Arquitectura de Referencia SOA

Esta capacidad Arquitectónica puede compararse con la capa Interfaz del Consumidor de la Arquitectura de referencia propuesta por el OpenGroup.

2.5 ¿Pueden considerarse equivalentes los conceptos de recurrencia y transversalidad?

El modelo SOA del OpenGroup considera la existencia de cuatro capas transversales (cross cutting), capa de Integración, capa de Calidad del Servicio, capa de Arquitectura de Información y finalmente capa de Gobernabilidad; por su parte Microsoft implementa un concepto distinto al que denomina funcionalidades recurrentes, aunque ambos conceptos implícitamente describen funcionalidades presentes a lo largo de las capas del modelo no es posible establecer una correspondencia natural entre las funcionalidades recurrentes y las capas transversales del OpenGroup, debido a que ambos modelos proponen filosofías distintas, la transversalidad puede ser aplicada a una arquitectura en capas, la recurrencia de funcionalidades es aplicable a niveles concebidos como arquitecturas que son independientes entre sí.

A pesar de esta clara división, previamente en la descripción de cada funcionalidad se estableció intuitivamente un vínculo entre funcionalidades y capas transversales, tomando en cuenta las funcionalidades propuestas por ambos modelos.

Además el modelo abstracto SOA de Microsoft explícitamente incluye una funcionalidad que podría interpretarse como una capa transversal con funcionalidades análogas a las definidas en el modelo SOA del OpenGroup, ésta es la funcionalidad de Administración Integrada y Gobernabilidad, cuya correspondencia con el modelo del OpenGroup parece ser directa con la Capa de Gobernabilidad.

En conclusión, para poder establecer un modelo de correspondencia conciso es necesario conocer profundamente a ambos modelos, sin embargo ese no es el objetivo de este trabajo. Se deja este análisis hasta este punto para continuar con la descripción de la plataforma SOA de Microsoft.

⁶ Refiérase al capítulo 3 Secciones 3.2 y 3.3

2.6 Consideraciones del Modelo Abstracto de SOA

Como se mencionó en la sección 2.4, cada una de las capacidades arquitectónicas recurrentes se extiende a lo largo de las tres arquitecturas del modelo abstracto de Microsoft, éste a su vez provee un listado de consideraciones para cada una de las capacidades arquitectónicas recurrentes. En la tabla [número de tabla] se listan las consideraciones divididas por Arquitectura y por capacidad recurrente [10]:

Arquitectura de implementación de servicios (Exposición)	
Mensajes y Servicios	Determinar qué exponer y cómo
	Contratos de operación de servicios
	Contratos de mensajes y datos
	Configuración, conductas y control
	SLA's
	Gobernabilidad
	Administración de versiones
Flujo de trabajo y Proceso	Coordinación de servicios para procesos de ejecución prolongada
	Seguimiento de servicios con capacidades de acceder a eventos dentro del flujo de trabajo
	Servicios de compensación
Datos	Servicios de entidad
	Servicios de agregación de entidades
	Manejo de Datos Maestros
Experiencia de Usuario	Servicios Especializados para soportar interfaces de usuario.
Identidad y Acceso	Manejo de Identidad
	Servicios de Delegación y Personificación
	Subsistemas confiables
	Autenticación (Kerberos, SSL)
	Control de acceso basado en roles
	Crear/Revocar relaciones de confianza
	Los servicios necesitan hacer decisiones de autorización, tales como la aprobación de una solicitud antes de llevar a cabo una transacción de negocio.
	El servicio debe conocer la identidad del usuario final quien esta realizando la solicitud.

	<p>La necesidad de colocar en el flujo la identidad del usuario final es una propiedad inherente al modelo de delegación, no se utiliza para el modelo de subsistema confiable. Deben llevarse a cabo esfuerzos especiales para incluir esta característica.</p> <p>Para soportar la noción de confianza como está definido por el modelo, los servicios al menos deben ser capaces de:</p> <ol style="list-style-type: none"> 1. Autenticar / verificar la identidad de servicios de flujo entrante/ flujo saliente. 2. Decidir si el servicio es un subsistema confiable para funciones específicas (incluyendo la propagación de peticiones de identidad) 3. Proteger la integridad de los datos que son comunicados entre subsistemas confiables y servicios <p>Además de los datos de aplicación, también los datos de identidad tales como las peticiones de identidad del usuario final deben ser protegidas para que ningún (man-in-the-middle) pueda modificar la información de identidad que está en tránsito.</p>
Arquitectura de integración de servicios (Composición)	
Mensajes y Servicios	Patrones de interacción de servicios
	Exposición de orquestaciones como servicios
	Patrones de invocación de servicios asíncronos
Flujo de trabajo y Proceso	Transacciones
	Alta frecuencia de cambio
	Reglas de negocio
	Orquestación de servicio
	Patrones de interacción de servicio
	Proceso de externalización
	Proceso de ejecución prolongada
	Auditoría y analíticos
Datos	Seguimiento del estado de una instancia específica de flujo de trabajo
	Transformación de datos (ETL)
	Procesamiento y almacenamiento confiables de mensajes.
	Replicación
	Sincronización
	Repositorio de metadatos y administración
	Reconciliación de instancia

	Reconciliación de esquema
	Replicación de documento
	Sindicación / Agregación
Experiencia de Usuario	Aplicaciones compuestas (OBAs)
	Flujos de trabajo humanos (MOSS)
	Orquestaciones inician flujos de trabajo humanos via el adaptador de SharePoint
	Flujos de páginas
Identidad y Acceso	Delegación y suplantación
	Aprovisionamiento
	Sincronización del repositorio de identidad
	Flujos de trabajo de aprobación
Arquitectura de Aplicación (Consumo)	
Mensajes y Servicios	Consumo de servicios basados en formularios
	Partes Web
	Registro de servicios – registro/salida/búsqueda
	AJAX, REST
Flujo de trabajo y Proceso	Flujos de trabajo humanos (MOSS)
	Administración de eventos (CAB)
	Flujo de páginas
Datos	Entidades (OBA Business Data Catalog)
	Vista simple del problema del cliente
	JSON
Experiencia de Usuario	Aplicaciones compuestas (OBAs)
	Personalización, perfiles de usuario
	Portales
	Inteligencia de negocios
	Reportes
	Agregación de contenido
	Experiencia de usuario declarativa
Identidad y Acceso	Registro simple (sincronización de contraseña)
	Identificación de usuario
	Acceso basado en roles (RBAC)
	Servicios de directorio

	Manejo de contraseña
	Privacidad (firewalls, encriptación)
	compliance (cumplimiento de normas)

Tabla C2-T1 Lista de consideraciones para la implementación de una SOA⁷

Cabe mencionar que la tabla C2-T1 no es un listado completo, sólo es la propuesta de Microsoft.

En toda implementación de una SOA es necesario elegir qué consideraciones se tomarán en cuenta para el modelo, esta decisión debe estar basada en las necesidades de la organización, en la propuesta que se describe en el capítulo 3 se seleccionan un conjunto acotado de consideraciones, esto permite delimitar el modelo mínimo necesario para una SOA en el ámbito de las Pymes.

2.7 Plataforma Tecnológica SOA de Microsoft.

La plataforma de Microsoft ofrece un conjunto de herramientas que hacen posible la implementación de una SOA, la imagen C2-3 describe la Plataforma SOA de Microsoft dividida por capacidad arquitectónica recurrente:

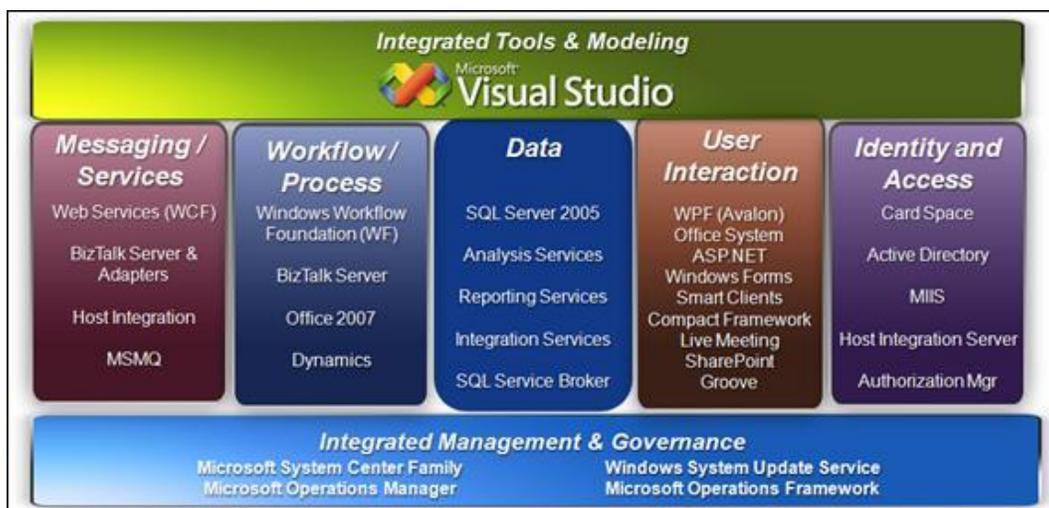


Imagen C2-3 Plataforma tecnológica SOA de Microsoft.⁸

⁷ Información obtenida de [20].

⁸ Microsoft Technological Platform Imagen Recuperada de [20]

Capítulo 3 Propuesta de SOA para las PyMEs

3. Propuesta de SOA para las PyMEs

En este capítulo se propone un modelo que es aplicable a la integración de procesos de negocio en sistemas dispares a través de una arquitectura SOA. El ámbito de este modelo son las organizaciones pequeñas y medianas.

Generalmente se concibe a una SOA como una inversión en TI considerable que sólo grandes organizaciones pueden pagar. Sin embargo, ningún modelo de referencia de SOA establece que esta arquitectura sólo sea aplicable a organizaciones de gran tamaño. De acuerdo con la hipótesis de este trabajo una SOA puede aplicarse a una organización de cualquier dimensión siempre y cuando existan procesos de negocio que requieran la integración de sistemas heterogéneos.

El modelo que se propone en este capítulo comprende 6 fases, éstas son:

1. Modelado del negocio
2. Implementación de los servicios web
3. Modelado del flujo de trabajo
4. Implementación de la interface de usuario
5. Despliegue
6. Seguimiento

Las fases vinculadas con el modelo SOA son las fases 2, 3, 4 y 6 comparables con capacidades arquitectónicas recurrentes del modelo abstracto SOA de Microsoft, la tabla C3-T1 describe esta equivalencia:

Fase	Capacidad Arquitectónica Recurrente
Implementación de los servicios web	Mensajes y Servicios
Modelado del flujo de trabajo	Flujo de Trabajo y Procesos
Implementación de la interface de usuario	Experiencia de Usuario
Seguimiento	Flujo de Trabajo y Procesos, Datos, Experiencia de Usuario.

Tabla C3-T1 Equivalencia entre las fases del modelo y las capacidades arquitectónicas recurrentes.

Estas fases pueden extenderse a más de una Capacidad Arquitectónica, éste es el caso de la fase 6 del modelo que es equivalente con las capacidades flujo de trabajo y procesos, datos y experiencia de Usuario.

Con respecto a las consideraciones del modelo abstracto de Microsoft, el objetivo de este trabajo es proponer un modelo general orientado a las PyMEs, por lo tanto la tabla C3-T2 describe una selección de consideraciones que responden a este ámbito:

Arquitectura de implementación de servicios (Exposición)	
Mensajes y Servicios	Determinar qué exponer y cómo
	Contratos de operación de servicios
	Contratos de mensajes y datos

Flujo de trabajo y Proceso	Coordinación de servicios para procesos de ejecución prolongada
	Seguimiento de servicios con capacidades de acceder a eventos dentro del flujo de trabajo
Arquitectura de integración de servicios (Composición)	
Flujo de trabajo y Proceso	Orquestación de servicio
	Proceso de ejecución prolongada
Datos	Seguimiento del estado de una instancia específica de flujo de trabajo
	Procesamiento y almacenamiento confiable de mensajes.
Arquitectura de Aplicación (Consumo)	
Experiencia de Usuario	Portales
	Experiencia de usuario declarativa

Tabla C3-T2 Selección de consideraciones para SOA en el ámbito de las PyMEs.

A continuación se describen cada una de las fases:

Fase 1.

3.1 Modelado del negocio (UML)

Para que una organización pueda alcanzar sus objetivos, generalmente debe estructurar sus actividades en un conjunto de procesos de negocio. Existen procesos de negocio que no superan las barreras departamentales y aquellos procesos más complejos que requieren la interacción de distintos departamentos involucrando entornos heterogéneos compuestos por plataformas y sistemas dispares, estos últimos son los procesos de negocio que pueden ser abordados por una SOA.

El primer paso en esta propuesta es identificar procesos de negocio que involucren distintos sistemas heterogéneos para su realización.

3.1.1 Identificación de Procesos de Negocio

Para completar este paso el punto de partida serán los objetivos estratégicos de la organización. Naturalmente la complicación en este respecto es la abstracción y complejidad existente en dichos objetivos, por esta razón es necesario descomponer cada objetivo en un conjunto de subobjetivos más concretos.

Estos subobjetivos pueden a su vez ser descompuestos en otros, de manera que se define una jerarquía de objetivos, aquellos que no pueden subdividirse más son los que se consideran procesos de negocio [22].

A partir de los procesos identificados se seleccionan aquellos procesos que requieren de la integración de varios sistemas dispares para su realización, para facilitar esta labor se propone dibujar un organigrama en el que se describa la jerarquía de objetivos para poder identificar aquellos procesos de negocio susceptibles para una SOA, durante todo este proceso debe participar activamente el personal vinculado con los procesos de negocio para que sean ellos quienes puedan brindar la información necesaria para decidir qué procesos de negocio requieren la integración de sistemas heterogéneos.

Finalmente se elige un proceso de negocio a implementar.

3.1.2 Identificación de los Roles en los Procesos de Negocio

Una vez elegido el proceso de negocio susceptible para SOA, es preciso delimitar a los actores involucrados en su realización. Cada uno de estos actores desempeña cierto papel cuando colabora con otros para llevar a cabo las actividades que conforman dicho proceso de negocio. En este punto es necesario distinguir dos tipos de actores, los que forman parte de la organización y aquellos externos a la misma.

3.1.3 Caso de uso general

Una vez elegido el proceso de negocio y los roles involucrados en él, puede construirse un diagrama de caso de uso general, en él cada proceso que conforma al proceso de negocio elegido aparecerá como un caso de uso.

Otra característica del caso de uso general es que sólo refleja los roles externos, esto con el objetivo de delimitar el entorno del proceso de negocio en cuestión.

3.1.4 Descripción de casos de uso

El siguiente paso es describir a detalle los casos de uso que comprende el caso de uso general. Para ello se utiliza una plantilla general de descripción, ésta contiene los campos Caso de uso, Objetivo, Descripción, prioridad, y sistemas a integrar.

Después de realizada la descripción de los casos de uso del Negocio, es momento de determinar los agentes internos que juegan un rol en el caso de uso del negocio, para ello se toma en cuenta el organigrama de la organización, y la descripción del caso de uso (tercer campo de la plantilla), ambos aspectos contribuyen para identificar el conjunto completo de roles involucrados, tanto externos como internos a la organización.

Al final, concluida la plantilla y delimitados los roles que forman parte del caso de uso seleccionado se dibujan un conjunto de diagramas que describen completamente el caso de uso general.

3.1.5 Diagramas de secuencia UML

El diagrama de secuencia describe el comportamiento de los roles que intervienen en la interacción. En cada proceso podemos distinguir entre el flujo básico de la interacción y los posibles flujos alternos.

3.1.6 Diagrama de proceso

Tomando como base el comportamiento descrito en el diagrama de secuencia se propone un diagrama de actividades delimitado por calles (swimlanes), cada calle representa a uno de los roles considerados en el diagrama de secuencia, dicha calle contiene las actividades realizadas por el rol correspondiente, esto permite representar de forma más detallada el flujo de trabajo que realiza cada caso de uso. Si alguna actividad es demasiado compleja, ésta puede ser descrita a través de otro diagrama de proceso. También es posible mostrar en diferentes diagramas el flujo normal y el flujo alternativo.

Dado que el caso de uso elegido integra la funcionalidad de diversos sistemas heterogéneos, el diagrama de proceso también incluye las referencias a los servicios requeridos para completar cada actividad.

El diagrama también muestra la información que necesita y produce cada actividad y el orden en el que se ejecutan las actividades. Los datos aparecen como objetos que fluyen entre las actividades y pueden tener un estado. Estos objetos son conocidos como objetos de información.

Objetos de Información

En un escenario de orquestación de servicios los objetos de información están divididos en dos categorías:

- **Información del negocio.** La información que necesita y produce cada actividad.
- **Información de la ejecución del flujo de trabajo.** La información que describe el estado de la ejecución del flujo de trabajo.

3.1.7 Diccionario de Datos

Las especificaciones de los de objetos de información que aparecen en los diagramas de procesos se reúnen en un diccionario de datos. Cada objeto de información se describe mediante un conjunto de atributos y restricciones de integridad que reflejan las reglas de negocio.

3.1.8 Diagrama de clases

Los objetos de información que fluyen entre las actividades de un caso de uso del negocio representan datos del dominio por lo que suponen una buena base para crear un modelo conceptual. Así cada objeto de información del diagrama de proceso se convertirá ahora en un concepto (y en la etapa de diseño dará lugar a una clase si el sistema debe dar soporte a dicho concepto). A partir de la especificación de cada objeto de información se obtendrá la definición del concepto asociado es decir, sus atributos relaciones con otras clases y restricciones.

Este modelo se describe mediante un diagrama de clases UML.

Nótese además que cuando un modelo conceptual evoluciona hacia un diagrama de clases, algunas responsabilidades se pueden obtener a partir de ciertas restricciones ya especificadas en el glosario.

Por otro lado, alguno de los roles identificados en el modelo del negocio, y por tanto especificado en el modelo de roles, podría ser incluido como una clase en el modelo conceptual [22].

Fase 2.

3.2 Implementación de los servicios web

Como se ilustra en la figura 3-1 esta fase es equivalente con la Capacidad Arquitectónica Mensajes y Servicios, por lo tanto la lista de consideraciones elegidas para llevar a cabo durante esta fase son:

Arquitectura de implementación de servicios (Exposición)
Determinar qué exponer y cómo
Contratos de operación de servicios
Contratos de mensajes y datos

Un servicio web representa una parte de la funcionalidad de un sistema que será expuesta para que pueda formar parte de un proceso de negocio. La primera consideración, “determinar qué exponer y cómo” establece qué actividades del diagrama de procesos requieren funcionalidades o datos de algún sistema externo. El “qué y cómo” van de la mano con la siguiente consideración “contratos de operación y de servicios”, ésta utiliza a WSDL para definir la interface pública del Servicio Web, esto es, la información necesaria para interactuar con él.

El WSDL especificará: nombre del método, datos de los parámetros de entrada y valores de retorno (nombre, tipo, descripción), formato del mensaje, las operaciones que pueden ser realizadas y qué mensajes involucran, entre otros.

Finalmente la consideración “contratos de mensajes y datos” es opcional, se refiere a implementar un registros de servicios Web utilizando el estándar UDDI [Capítulo 1], dónde el protocolo SOAP [Capítulo 1 – Capa de descubrimiento] funge como el protocolo para el envío de mensajes y llamadas a procedimiento remotos (RCP) a través de internet, esto con el objetivo de establecer un mecanismo de interacción entre el registro de servicios, el proveedor y el cliente.

Las tecnologías para la implementación de la fase 2 son:

- Windows Communication Foundation (WCF)
- NuSOAP de PHP

3.2.1 Windows Communication Foundation (WCF)

Es un marco de trabajo para construir aplicaciones orientadas a servicios. Utilizando WCF es posible enviar datos como mensajes asíncronos [mensajes que suceden a lo largo de la ejecución de una aplicación, normalmente para solicitar datos o utilizar la lógica de algún otro sistema] entre un servicio y otro. Los mensajes pueden ser tan simples como un carácter o palabra enviada como XML, o tan compleja como un flujo de datos binarios.

Un servicio puede estar disponible de dos maneras, hospedado por IIS (Internet Information Services) o bien, hospedado en una aplicación. [23]

WCF también permite exponer un flujo de trabajo implementado utilizando Windows Workflow Foundation (WF) como un servicio débilmente acoplado, esto permite que cualquier cliente

creado en cualquier plataforma pueda conectarse a cualquier servicio siempre y cuando los contratos esenciales se cumplan .

WCF logra la interoperabilidad para protocolos de aplicación a través del lenguaje de descripción de esquema XML 1.0 (XSD) y WSDL 1.1

La interoperabilidad para protocolos de infraestructura es provista por las especificaciones WS-* a través de canales WCF los cuales son configurados como elementos enlazados del marco de trabajo .NET, en la tabla C3-T3 se listan éstos elementos y los protocolos de infraestructura WS-* que implementan:

Elemento Enlazado (BindingElement)	Especificación / Documento
HttpTransportBindingElement	HTTP 1.1
	SOAP 1.1 HTTP Binding
	SOAP 1.2 HTTP Binding
TextMessageEncodingBindingElement MtomMessageEncodingBindingElement	XML
	SOAP 1.1
	SOAP 1.2 Core
	WS-Addressing 2004/08
	W3C Web Services Addressing 1.0 – Core
	W3C Web Services Addressing 1.0 - SOAP Binding
	W3C Web Services Addressing 1.0 - WSDL Binding*
	W3C Web Services Addressing 1.0 Metadata
	WSDL SOAP1.1 Binding
	WSDL SOAP1.2 Binding
MtomMessageEncodingBindingElement	XOP
	MTOM + SOAP1.2 Binding
	MTOM SOAP 1.1 Binding
	MTOM WS-PolicyAssertions
SecurityBindingElement	WSS: SOAP Message Security 1.0
	WSS: Username Token Profile 1.0
	WSS: X.509 Token Profile 1.0
	WSS: SAML 1.1 Token Profile 1.0
	WSS: SOAP Message Security 1.1
	WSS Username Token Profile 1.1
	WSS: X509 Token Profile 1.1
	WSS: Kerberos Token Profile 1.1
	WSS: SAML 1.1 Token Profile 1.1
	WS-Secure Conversation
	WS-Trust 1.4
	WS-SecurityPolicy 2005/07
	WS-ReliableMessaging 1.1
TransactionFlowBindingElement	WS-Coordination
	WS-AtomicTransaction
MetadataExporter, MetadataImporter and MetadataResolver	XML Schema Part 1: Structures Second Edition

	XML Schema Part 2: Data types Second Edition
	WSDL 1.1
	WS-Policy 1.2
	WS-Policy 1.5
	WS-PolicyAttachment 1.2
	WS-MetadaExchange 1.1
	WS-Transfer Get for metadata retrieval

Tabla C3-T3 Interoperabilidad de los protocolos de servicios Web⁹

3.2.2 Librería nusoap

La fase 2 no sólo utiliza tecnologías provistas por la plataforma tecnológica de Microsoft, el uso de la Librería NuSOAP de PHP(Hypertext Preprocessor) demuestra que en una arquitectura dónde existe un débil acoplamiento es posible integrar tecnologías ya que los estándares que permiten exponer funcionalidades “hablan el mismo idioma”, servicios WEB, esto quiere decir WSDL y SOAP.

Para la implementación de servicios web en la plataforma PHP se propone la librería nusoap, esta librería permite establecer la interfaz de un servicio web basándose en estándar WSDL 1.0, dicha interfaz es compatible con los enlaces disponibles en Microsoft Visual Studio a través de la opción agregar referencia a servicio web.

Comparación con las otras herramientas disponibles:

Biztalk server es considerado el mejor recurso existente para realizar la integración de aplicaciones dispares pues es fácil de monitorear, es robusto y es escalable, esto no significa que una solución a través de Biztalk no utilice los servicios creados a través de WCF.

En Biztalk la orquestación es hospedada dentro de un Proceso BizTalk y depende del tiempo de ejecución del servidor BizTalk.

Justificación del uso de WCF

El objetivo de este trabajo es proponer una SOA de bajo costo que habilite la integración entre sistemas dispares, WCF es totalmente gratuito, no tiene restricciones de uso, permite exponer las funcionalidades de los sistemas existentes en una organización y a partir de éstas construir procesos de negocio, además en conjunción con WF, es posible orquestar nuevos procesos de negocio utilizando los flujos de trabajo previamente definidos, a esto Microsoft lo denomina un modelo Fractal [10].

Es obvio que WCF nativamente no incluye todas las capacidades que BizTalk ofrece, sin embargo, es posible implementarlas con una solución ajustada a las necesidades de la organización.

⁹ Información obtenida de [24].

Fase 3.

3.3 Modelado del flujo de trabajo

Esta fase es equivalente con la funcionalidad recurrente flujo de trabajo y procesos, las consideraciones a realizar durante esta fase son:

Arquitectura de integración de servicios (Composición)	
Flujo de trabajo y Proceso	Orquestación de servicios
	Proceso de ejecución prolongada

Recordemos que WSDL y SOAP permiten exponer funcionalidades para que éstas sean consumidas por un proceso de negocio que requiere información de los diversos sistemas en su entorno. Por si mismos WSDL y SOAP no pueden ejecutar estas actividades en la secuencia especificada por el proceso de negocio.

Esta fase es el núcleo de la integración de procesos de negocio, esta consiste en pasar el diagrama de procesos a su representación como flujo de trabajo, por medio de un software BPM o bien alguna herramienta con una funcionalidad análoga. Cada actividad del proceso formará parte del flujo de trabajo, inclusive aquellos procesos que implican extraer datos de sistemas heterogéneos, y el objetivo será la automatización del proceso, a esto lo llamamos “orquestación de servicios”.

El BPM también se encarga de la segunda consideración “Proceso de ejecución prolongada” para ello almacena en una base de datos el estado del flujo de trabajo en ejecución. Otra de sus funcionalidades es la posibilidad de entregar información de seguimiento para monitorear la ejecución en tiempo real de flujos de trabajo esto con el objetivo de controlar el funcionamiento del flujo de trabajo, también permite obtener un mayor conocimiento del negocio.

La herramienta elegida para implementar la fase 3 se describe a continuación:

Workflow Foundation (WF)

Microsoft Workflow Foundation es un marco de trabajo que permite implementar flujos de trabajo de ejecución prolongada, es decir flujos de trabajo cuya ejecución pueda extenderse a días, semanas o meses, habilitando a los sistemas para obtener información asíncrona cuando alguna de las actividades requiera extraer datos o requiera utilizar la lógica de otros sistemas.

Permite que aplicaciones orientadas a servicios puedan implementar la composición de servicios a través de un flujo de trabajo, haciendo posible incluso que puedan realizarse cambios al vuelo en los procesos de negocio, este tipo de flexibilidad es un requerimiento común para flujos de trabajo que interactúan con personas.

Este marco de trabajo ofrece también herramientas para crear y modificar flujos de trabajo gráficamente. Dado que un flujo de trabajo consiste en un número definido de pasos, puede ser construido utilizando herramientas que ilustran esos pasos y las relaciones entre ellos.

Finalmente ofrece la habilidad de monitorear un flujo de trabajo en ejecución examinando su ejecución en tiempo real.

Comparación con las otras herramientas disponibles

La plataforma de Microsoft ofrece también a BizTalk Server como modelo de ejecución y orquestación de flujos de trabajo, la principal diferencia entre ambas tecnologías es que WF es utilizado para flujos de trabajo hospedados en aplicaciones y flujos de trabajo humanos, mientras que Biztalk es utilizado para sistemas de flujo de trabajo hospedados en un servidor.

Biztalk ofrece funcionalidades extendidas como el monitoreo de la Actividad de Negocios (BAM), B2B y la integración de aplicaciones empresariales (EAI). Además provee una extensiva línea de adaptadores y componentes de línea de producción que son un “deber tener” para la integración de aplicaciones empresariales heterogéneas (distinta plataforma). Otras características que Biztalk tiene pero que en WF deben ser implementadas son:

- Monitoreo: Nativamente integrado con el Monitoreo de Actividades de Negocio (BAM), aunque AppFabric provee algún monitoreo de salud.
- Transacción: Soporta una transacción atómica y una transacción de ejecución prolongada.
- Un conjunto extensivo de herramientas para administración, migración, y escalamiento comparado con AppFabric.

Justificación del uso de WF

Naturalmente BizTalk server incluye funcionalidades extendidas que de manera nativa están incluidas con el servidor, sin embargo no significa que no puedan ser implementadas con WF.

La funcionalidad clave de BizTalk es obviamente la Integración de Procesos de Negocio, la lista de conectores para sistemas heterogéneos es tan amplia que garantiza cubrir cualquier necesidad de Integración a pesar de la plataforma. WF por su parte cuenta con la solidez del marco de trabajo .NET de Microsoft, el cual permite registrar referencias a Servicios Web sin mayor configuración que brindar la URL del contrato de servicio WEB, si el contrato está comprendido en alguna de las tecnologías WS*, éste será reconocido sin dificultades.

Una ventaja de WF sobre BizTalk es que soporta la actualización dinámica de flujos de trabajo lo cual lo hace ideal para implementar procesos de negocio que puedan ser actualizados al vuelo por analistas de negocio que no cuenten con experiencia en programación.

Fase 4.

3.4 Implementación de la interface de usuario

Puede enmarcarse esta fase con la capacidad recurrente experiencia de usuario, la consideración correspondiente es:

Arquitectura de Aplicación (Consumo)	
Experiencia de Usuario	Portales

Esta fase se encarga de consumir la composición de servicios completada en la fase anterior a través de una interface para los usuarios que interactúan con el proceso de negocio, existen diversas opciones de aplicación de consumo, desde clientes ricos, aplicaciones móviles, portales web, entre otros. Para este modelo la aplicación de consumo estará basada en una aplicación web.

La oferta de herramientas que hacen posible la implementación de un portal web en poco tiempo es muy amplia, en esta propuesta y dado que estoy utilizando la plataforma tecnológica de Microsoft, la elección más lógica es ASP.NET, éste es descrito a continuación:

ASP.NET

Active Server Pages es una herramienta basada en el modelo cliente-servidor que permite la implementación de sistemas web. A partir de la liberación del marco de trabajo .NET y su integración con el Entorno de Desarrollo Integrado Visual Studio, ASP adoptó el nombre ASP.NET.

A diferencia de su predecesor ASP clásico, ASP.NET permite generar contenido web brindando una amplia gama de recursos para agilizar y facilitar el desarrollo de aplicaciones basadas en web.

Justificación del uso de ASP.NET

La arquitectura de aplicación (consumo) establece que una vez que un proceso de negocio ha sido creado, éste debe estar disponible para los usuarios finales. En el caso de este trabajo la entrega de una nueva aplicación resultado de la composición de servicios se llevará a cabo a través de una aplicación basada en Web.

Para lograr este objetivo se propone ASP.NET que es parte de la plataforma SOA de Microsoft, no la comparo con otras tecnologías ya que para generar aplicaciones web ASP.NET es la única disponible en esta plataforma.

ASP.NET está totalmente integrada al Entorno de desarrollo Visual Studio y por lo tanto ofrece características que facilitan la integración con los marcos de trabajo WCF y WF que también son provistos por la misma IDE. En otras palabras, la integración entre los tres es directa.

Fase 5

3.5 Despliegue

En esta fase se lleva a cabo el despliegue en la plataforma elegida.

Los procesos de negocio están modelados en el diseñador de flujos de trabajo de WF, el motor de ejecución de flujos de trabajo nuevamente es WF hospedado en una aplicación local.

Los servicios Web se deben desplegar en servidores web HTTP, es posible desplegar todos los servicios en un servidor único, sin embargo lo más recomendable es ejecutar los servicios Web en los mismos equipos donde se encuentran las aplicaciones de software de donde consumen información.

A pesar de estas consideraciones, no existen reglas definidas para el despliegue de una SOA, la decisión puede estar fundamentada en otros aspectos como seguridad, desempeño, disponibilidad e infraestructura.

El despliegue puede resumirse en un diagrama de despliegue que es útil para futuras referencias requeridas por el personal de IT o bien para la realización de mantenimiento. Éste describe cómo los componentes que forman parte de la solución están distribuidos en equipos físicos.

Fase 6.

3.6 Seguimiento

En esta fase se llevan a cabo actividades de seguimiento y control tales como el monitoreo en tiempo real del proceso, estado y el historial de ejecuciones

En este modelo se propone la creación de un monitor de progreso que permita brindar a los usuarios no-programadores una herramienta gráfica de seguimiento de los procesos de negocio en ejecución. A continuación una descripción más detallada de esta herramienta:

3.6.1 Monitor Gráfico de Progreso

Como se describe en la figura 3-1 la fase de seguimiento es equivalente con 3 de las capacidades recurrentes, a continuación la lista de consideraciones relacionadas esta fase:

Arquitectura de implementación de servicios (Exposición)	
Flujo de trabajo y Proceso	Seguimiento de servicios con capacidades de acceder a eventos dentro del flujo de trabajo
Arquitectura de integración de servicios (Composición)	
Datos	Seguimiento del estado de una instancia específica de flujo de trabajo
Arquitectura de Aplicación (Consumo)	
Experiencia de Usuario	Experiencia de usuario declarativa

Tabla C3-T4 Interoperabilidad de los protocolos de servicios Web

Una herramienta que permita monitorear gráficamente el progreso de las distintas instancias de un flujo de trabajo comprende consideraciones de las 3 arquitecturas propuestas en el modelo abstracto SOA de Microsoft, esta situación se ilustra en la tabla C3-T4, a continuación se aborda cada una de estas consideraciones:

Seguimiento de Servicios con capacidades de acceder a eventos dentro del flujo de trabajo

La primera consideración, “Seguimiento de servicios con capacidades de acceder a eventos dentro del flujo de trabajo” está enmarcada en la Arquitectura de Exposición del modelo abstracto de Microsoft, la capacidad recurrente correspondiente es Flujo de Trabajo y Procesos. Esta consideración implícitamente involucra la creación de una interfaz de usuario que brinde facilidades de drill up, drill down y características de navegación a través de un flujo de trabajo extenso. Esto es útil debido a que el seguimiento de un flujo de trabajo generalmente se realiza

por usuarios no-programadores, quienes deben ser capaces de navegar a lo largo del gráfico que representa el proceso de negocio sin necesidad de estar vinculados con la interface de desarrollo.

Para alcanzar el objetivo descrito en el párrafo anterior se utiliza el depurador de flujos de trabajo incluido en la versión 4.5 de WF, a continuación se describe brevemente esta herramienta:

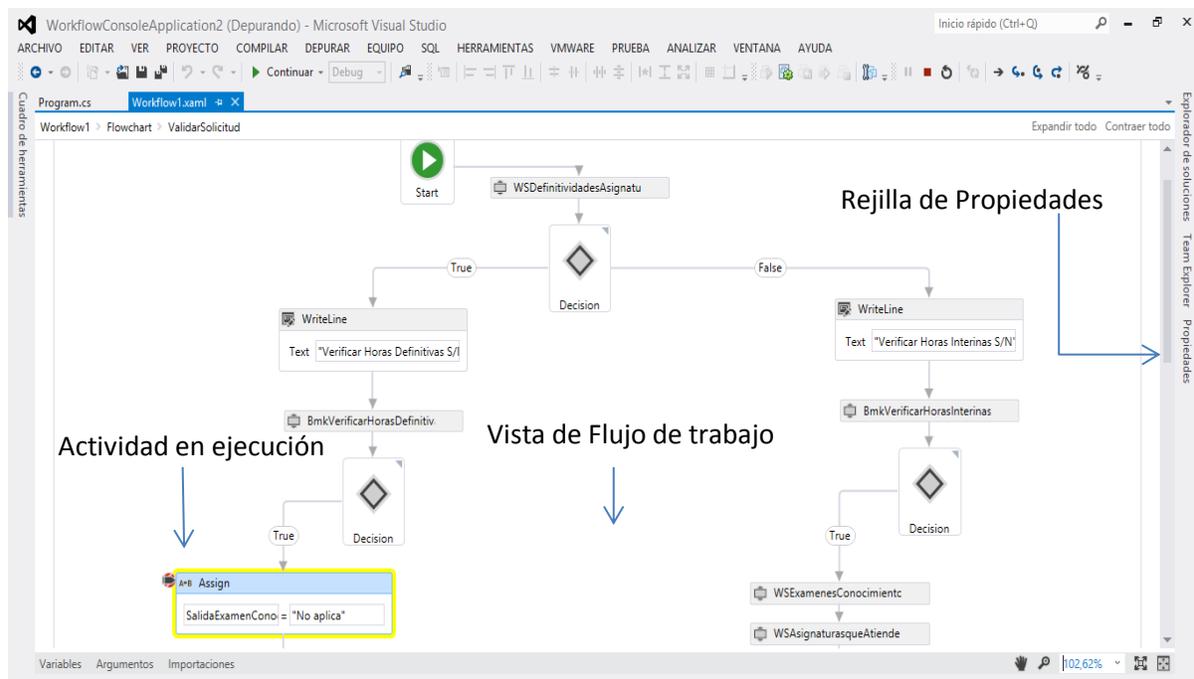
Depurador de flujos de trabajo

La versión 4.6.1 del marco de trabajo .NET ofrece distintas opciones para depurar flujos de trabajo en ejecución desde el entorno de desarrollo. Los flujos de trabajo pueden ser depurados en el diseñador, en el código XAML y en el código.

El servicio de depuración es público y puede ser utilizado como herramienta para monitoreo, simulación y depuración de flujos de trabajo, en esta propuesta esta característica permite realizar el monitoreo de la ejecución de una instancia de flujo de trabajo.

Componentes del diseñador de flujos de trabajo disponibles durante la depuración de una instancia

La figura C3-1 muestra los componentes del diseñador de flujos de trabajo presentes durante la depuración de una instancia:



Las capacidades que estos componentes proveen al escenario de depuración son:

Vista de Flujo de trabajo (WorkflowView): Este control representa el núcleo de cualquier escenario de re-hospedaje. Su función principal es ofrecer una navegación (simple) en diseños de flujos de trabajo, provee características como paneo (panning) y desplazamiento (scrolling) en diseños amplios sin importar su dirección (horizontal o vertical), múltiples funcionalidades de zoom.

Es ideal para diseñar escenarios dado que soporta la característica arrastrar y soltar (drag and drop), también puede ser utilizado para compartir diseños visualmente a través de su impresión y exportación en formato de imagen.

Finalmente las herramientas de representación son extensibles para que los desarrolladores puedan dibujar glyphs personalizados como íconos de estado.

Rejilla de Propiedades: La rejilla de propiedades es un control estándar del Windows Forms que permite mostrar los editores de propiedades y otras acciones en un escenario de re-hospedaje. Por ejemplo, este control puede lanzar los editores de reglas de flujos de trabajo.

Seguimiento del estado de una instancia específica de flujo de trabajo

La segunda consideración “Seguimiento del estado de una instancia específica de flujo de trabajo” está enmarcada en la Arquitectura de Composición del modelo abstracto de Microsoft, la capacidad arquitectónica recurrente correspondiente es datos. Esta consideración se refiere a la capacidad de guardar el estado de la ejecución de una instancia específica de flujo de trabajo en una base de datos para que el sistema pueda recuperar el estado de esa instancia posteriormente y pueda mostrar en la interfaz gráfica el estado actualizado.

Las herramientas que se proponen para completar este proceso son Microsoft SQL Server 2008 como manejador de base de datos y Workflow Instance Store, un espacio de nombres del Framework 4.6.1 de WF que permite persistir el estado de la información sobre las instancias de un flujo de trabajo almacenadas en una base de datos SQL Server 2008.

A continuación una descripción sobre ambas herramientas:

Microsoft SQL Server

SQL server es el manejador de bases de datos de Microsoft, al igual que muchos otros manejadores permite realizar consultas a través del lenguaje de consultas estándar SQL, al ser un producto de la familia de Microsoft incluye una conexión nativa con el ambiente de desarrollo integrado Visual Studio.

SQL Workflow Instance Store

Esta característica esta implementada como una clase llamada SQLWorkflowInstanceStore que deriva de la clase abstracta T:System.Runtime. Persistence.InstanceStore del marco de trabajo de persistencia de Microsoft .NET.

Esta clase constituye un proveedor de persistencia SQL, lo cual representa una implementación concreta de la Interfaz de Programación de la Aplicación (API) que una aplicación huésped utiliza para enviar comandos de persistencia al almacén de datos.

SQL Workflow Instance Store soporta flujos de trabajos auto hospedados utilizando la clase WorkflowApplication. También soporta servicios de flujos de trabajo y servicios hospedados en WAS(Windows Process Activation Service) utilizando WorkflowServiceHost, sin embargo esta

característica no la considero en la propuesta dado que el flujo de trabajo estará hospedado en una aplicación ASP.NET.

Experiencia de usuario declarativa

La consideración “Experiencia de usuario declarativa” está enmarcada en la Arquitectura Consumo, la funcionalidad recurrente a la que pertenece desde luego es Experiencia de usuario.

Esta clasificación apunta hacia el diseño de una aplicación destinada a usuarios finales dado que a través de ella “consumirán” los servicios y capacidades implementadas en las fases anteriores.

El objetivo de la consideración “Experiencia de usuario declarativa”, como su nombre lo indica, es brindar al usuario la capacidad de interactuar con la interfaz sin contar con conocimientos especializados de programación, ya que una experiencia declarativa apunta al qué se debe hacer, sin importar el cómo.

Recordemos que el diseñador permite que la configuración de flujos de trabajo de manera declarativa o bien a través de programación. La experiencia declarativa permite que el usuario no-programador pueda hacer cambios en el flujo de trabajo dinámicamente aprovechando la característica de drag and drop del diseñador de flujos de trabajo.

Capítulo 4. Caso de Prueba

4. Caso de Prueba: “Solicitud de horas adicionales para técnicos académicos” en la Dirección General del Colegio de Ciencias y Humanidades.

Con el objetivo de validar el modelo propuesto se aplicó una solución SOA en la Dirección General de Colegio de Ciencias y Humanidades, una instancia universitaria que regula el funcionamiento de los cinco planteles de la Escuela Nacional Colegio de Ciencias y Humanidades a través de sus nueve secretarías.

Fase 1

4.1 Modelado del negocio (UML)

4.1.1 Identificación del proceso de negocio

Entre los objetivos de la Dirección General está el de tramitar, dar seguimiento y ofrecer una respuesta por escrito a diversos asuntos del personal académico. El subobjetivo en el que centro esta propuesta es la solicitud de horas adicionales para Técnicos Académicos, de aquí en adelante éste será considerado el proceso de negocio a implementar.

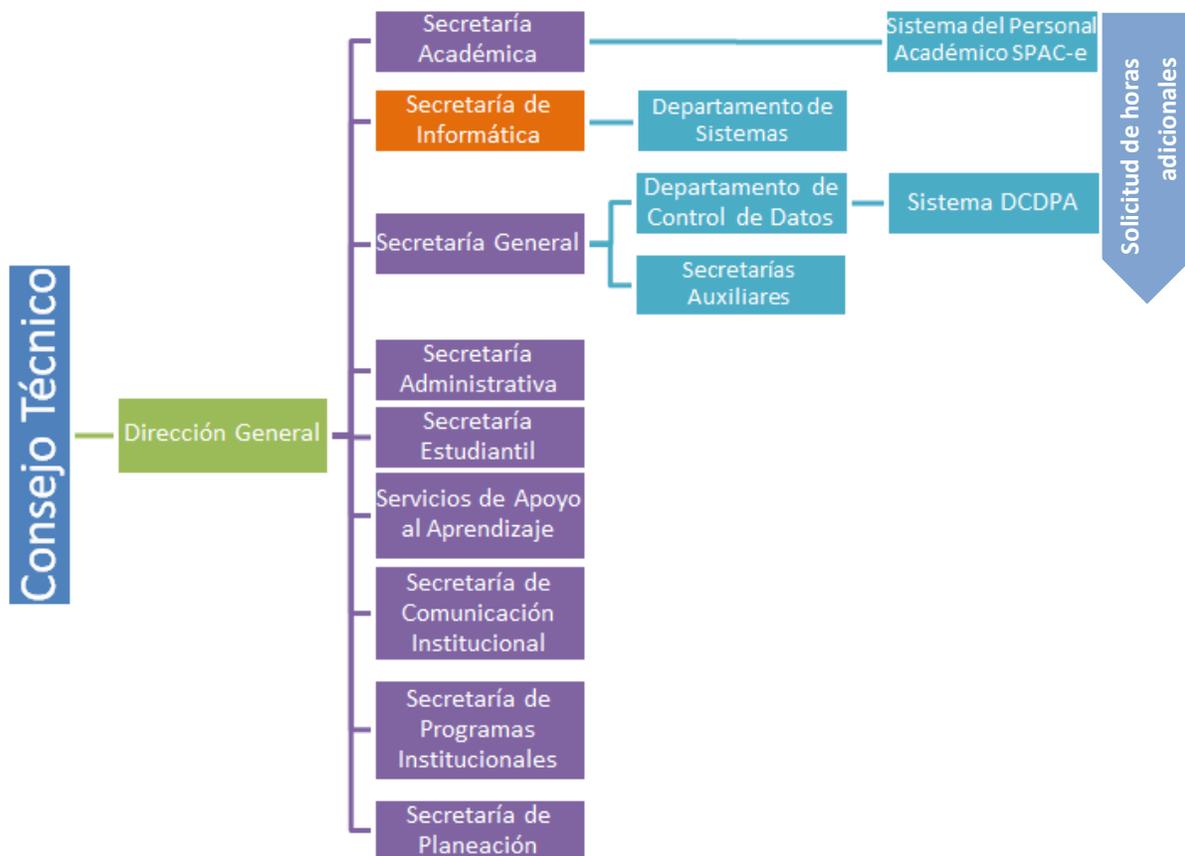


Figura C4-1 Organigrama de la Dirección General del CCH y ubicación jerárquica de los sistemas DCDPA y SPAC-E.

El proceso de negocio elegido involucra las actividades de procesamiento de solicitudes y administración de solicitudes. Más adelante cada una de estas actividades es representada como

un caso de uso, sin embargo sólo la primera de ellas es susceptible para ser implementada con una SOA dado que requiere información proveniente de dos sistemas heterogéneos que operan de forma independiente y que no tienen comunicación entre sí.

La figura C4-1 ilustra el organigrama de la Dirección General del CCH, también muestra la ubicación de los dos sistemas que estarán involucrados en el caso de uso procesamiento de solicitudes.

4.1.2 Identificación de los Roles en los Procesos de negocio

La tabla C4-T1 lista los roles involucrados en el proceso de negocio:

Rol	Interno/Externo
Plantel	Externo
Secretario Auxiliar	Interno
Consejo Técnico	Externo
Secretaría General	Interno

Tabla C4-T1 Roles involucrados en el proceso de negocio

4.1.3 Caso de Uso General

La figura C4-2 muestra el caso de uso general, éste incluye dos casos de uso descritos en el proceso de negocio "Solicitud de horas adicionales". Con el objetivo de describir el entorno del proceso de negocio sólo se muestran los roles externos.

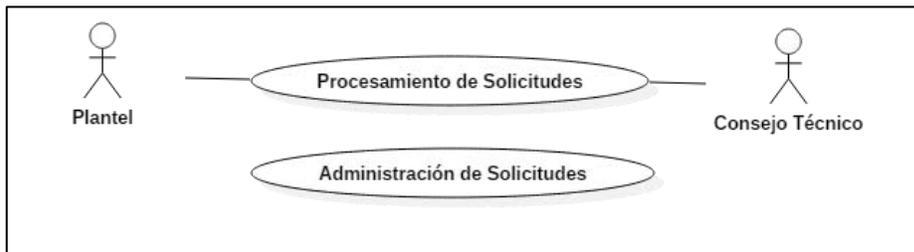


Figura C4-2 Caso de uso general

El caso de uso Procesamiento de Solicitudes puede ser dividido en 3 casos de uso más: Registrar Solicitudes, Validar Solicitudes y Aprobar Solicitudes, más adelante muestro el diagrama de secuencia que describe la interacción entre estas tres actividades y los actores involucrados en ellas.

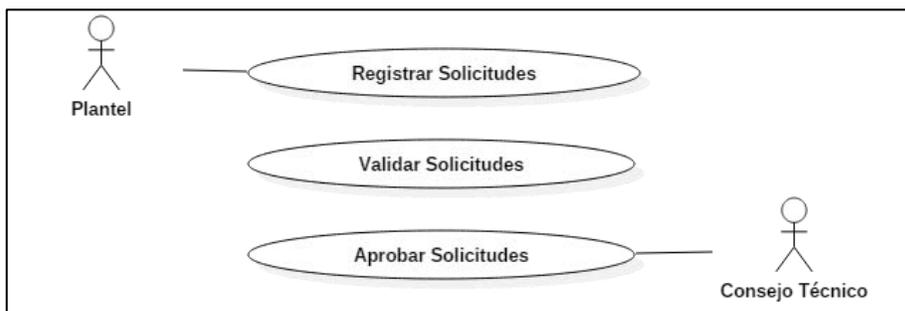


Figura C4-3 Caso de uso Procesamiento de Solicitudes

4.1.4 Descripción del caso de uso

Caso de Uso	Procesamiento de Solicitudes																
Objetivo	Procesar solicitudes de horas adicionales realizadas en los cinco planteles del CCH.																
Descripción	<p>Para realizar este proceso la dirección de los planteles envía una solicitud por escrito a la Secretaría General para que ésta a través del secretario Auxiliar correspondiente pueda ser validada. La validación puede tener dos vertientes:</p> <p>I. Las horas adicionales serán atendidas como profesor de asignatura interino, los requisitos a cubrir son:</p> <table border="1"> <thead> <tr> <th>Requisito</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>Examen de conocimientos de la materia acreditado.</td> <td>SPAC-E</td> </tr> <tr> <td>El número de horas asignadas debe ser menor o igual a 8.</td> <td>DCDPA</td> </tr> <tr> <td>El horario de clase no se traslapa con el horario de técnico académico.</td> <td>SPAC-E</td> </tr> </tbody> </table> <p>II. Las horas adicionales serán atendidas como Profesor de Asignatura Definitivo, los requisitos a cubrir son:</p> <table border="1"> <thead> <tr> <th>Requisito</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>El profesor es definitivo en la asignatura que atiende.</td> <td>DCDPA</td> </tr> <tr> <td>El número de horas asignadas debe ser menor o igual a 8.</td> <td>DCDPA</td> </tr> <tr> <td>El horario de clase no se traslapa con el horario de técnico académico.</td> <td>SPAC-E</td> </tr> </tbody> </table> <p>Cada una de las solicitudes validadas es presentada ante el Consejo Técnico del Colegio de Ciencias y Humanidades para que éste, como el organismo de mayor jerarquía “recomiende” la aprobación o no aprobación de los casos.</p>	Requisito	Sistema	Examen de conocimientos de la materia acreditado.	SPAC-E	El número de horas asignadas debe ser menor o igual a 8.	DCDPA	El horario de clase no se traslapa con el horario de técnico académico.	SPAC-E	Requisito	Sistema	El profesor es definitivo en la asignatura que atiende.	DCDPA	El número de horas asignadas debe ser menor o igual a 8.	DCDPA	El horario de clase no se traslapa con el horario de técnico académico.	SPAC-E
Requisito	Sistema																
Examen de conocimientos de la materia acreditado.	SPAC-E																
El número de horas asignadas debe ser menor o igual a 8.	DCDPA																
El horario de clase no se traslapa con el horario de técnico académico.	SPAC-E																
Requisito	Sistema																
El profesor es definitivo en la asignatura que atiende.	DCDPA																
El número de horas asignadas debe ser menor o igual a 8.	DCDPA																
El horario de clase no se traslapa con el horario de técnico académico.	SPAC-E																
Sistemas Integrar	<p>a La dirección general cuenta con una base de datos del personal académico que es resguardada y administrada por la Secretaría General a través del Departamento de Control de Datos del Personal Académico (DCDPA), éste departamento utiliza un sistema con el mismo nombre desarrollado en ASP clásico con un DBMS Microsoft SQL Server 2008 R2. El ámbito de los datos almacenados en el DCDPA son los registros históricos de los trámites de diversos asuntos del personal académico que son puestos a consideración ante el H. Consejo Técnico del Colegio de Ciencias y Humanidades.</p> <p>Por otro lado la Secretaría Académica cuenta con base de datos que resguarda y administra el departamento de Sistemas de la Dirección General. Se trata del Sistema de Personal Académico (SPAC-e) que está desarrollado en PHP con un DBMS MySQL. Los datos que el SPAC-E almacena son los registros históricos de los procesos que la Secretaría académica lleva a cabo en los cinco planteles.</p>																

	Ambos sistemas operan de manera independiente para dar soporte a su área correspondiente, no existe ningún tipo de comunicación entre ellos.
Prioridad	Normal

Tabla C4-T4 Plantilla de descripción del caso de uso

4.1.5 Diagrama de Secuencia del flujo básico

La figura C4-4 muestra el Diagrama de Secuencia UML del flujo básico del caso de uso procesamiento de solicitudes, en el cual una solicitud es registrada, validada por el secretario auxiliar correspondiente y finalmente aprobada por el Consejo Técnico.

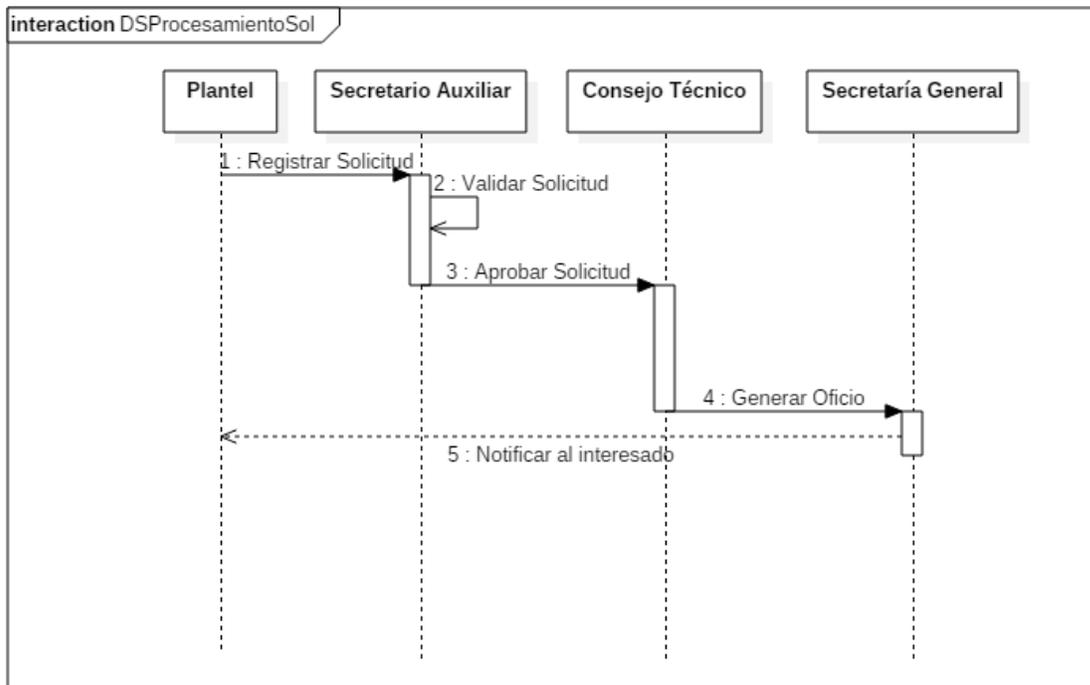


Figura C4-4 Diagrama de Secuencia del caso de uso básico.

4.1.6 Diagrama de Proceso

La figura C4-5 muestra el diagrama de proceso del caso de uso “Procesamiento de solicitudes” con su flujo básico y flujos alternativos, cada actor esta representado por una calle (swimlane). El diagrama también describe los objetos de información que necesita y produce cada actividad, estos objetos están representados por cuadros en línea punteada y una flecha hacia la actividad si el dato es una entrada y una flecha apuntando hacia afuera de la actividad si el dato es una salida. Estos objetos de información se integrarán más adelante en el diagrama de clases.

El diagrama también incluye los servicios web que cada actividad utiliza, los servicios web están identificados en rectángulos con línea continua y el nombre de cada servicio está precedido por el prefijo WS.

Diagrama de Proceso del caso de uso Procesamiento de Solicitudes

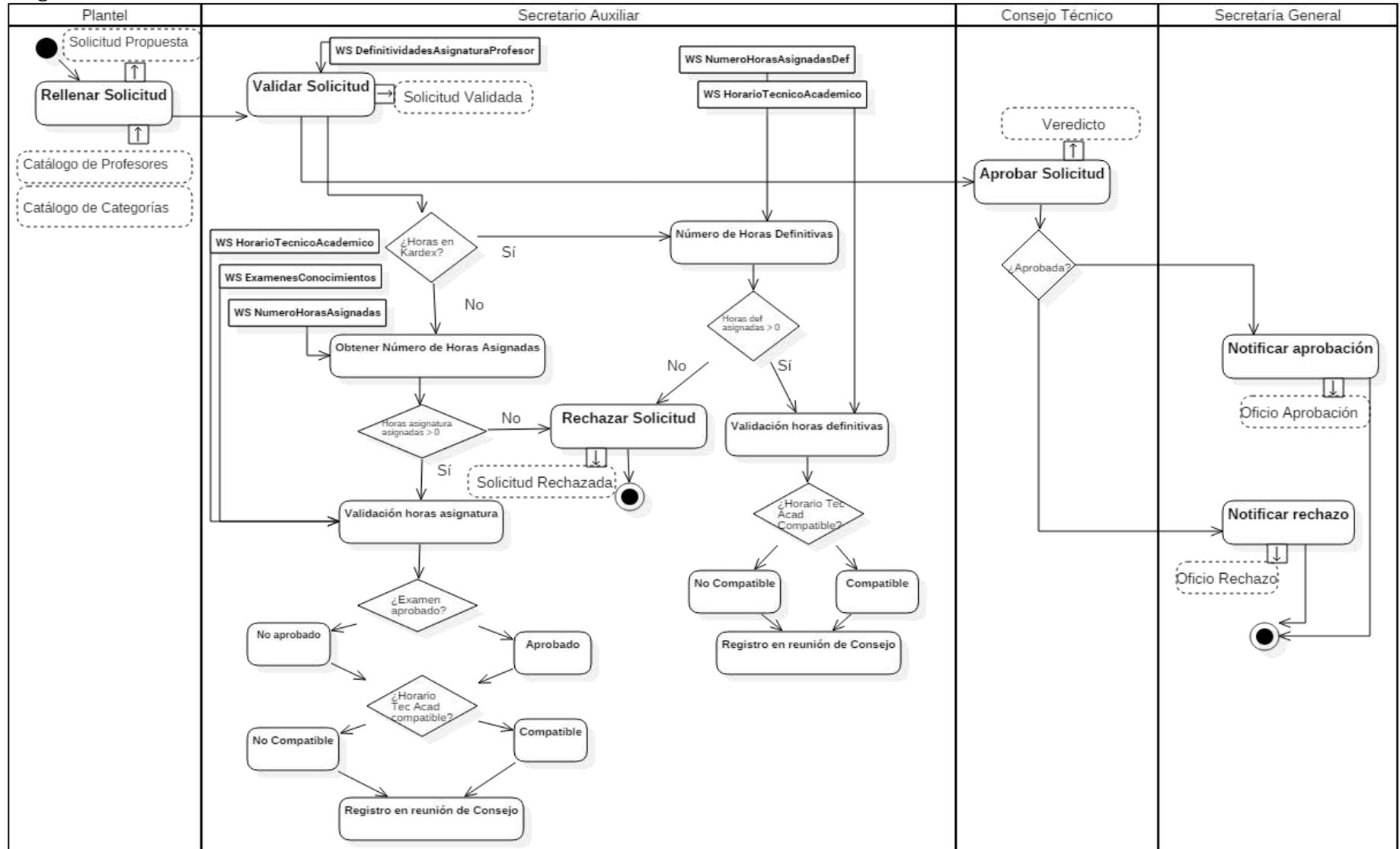


Figura C4-5 Diagrama de Proceso "Procesamiento de solicitudes"¹⁰

¹⁰ Notación basada en [17]

4.1.7 Diccionario de Datos

El diccionario de datos estará dividido en dos partes:

- Diccionario de datos del negocio.
- Diccionario de datos de la ejecución del flujo de trabajo.

Diccionario de datos del negocio:

Objeto de Información: Solicitud			
Nombre del atributo	Tipo	Descripción	Observaciones
idSesionCONSEJO	int	Clave de la sesión de ConsejoTécnico.	El consejo técnico se reúne periódicamente para evaluar las diversas solicitudes que durante cierto período de tiempo se acumulan.
id_asunto	int	Clave del tipo de solicitud.	En este caso sólo se están capturando solicitudes para impartir horas adicionales.
Periodo	int	Semestre en el que se realiza el trámite.	El semestre se captura en el forma aaaass, ej. 201710 haría referencia al semestre 2017-1.
id_plantel	int	Clave de plantel.	La clave del plantel corresponde a la clave usada en el catálogo de planteles de la base de datos del Sistema DCDPA. Restricción: El técnico académico debe tener horas asignadas frente a grupo en el plantel en que se realiza la solicitud.
RFC	String	RFC del Técnico Académico.	El RFC debe estar registrado en el catálogo de profesores.
idcategoría (idcatpropuesta)	int	Clave de categoría de contratación del profesor.	La clave de categoría corresponde a la clave usada en el catálogo de categorías..
id_area_acad	int	Clave del área académica.	La clave de área académica corresponde a la clave usada en el catálogo de áreas de la base de datos del Sistema DCDPA
id_area_tec_acad	int	Clave del área de técnico académico.	La clave de área de técnico académico corresponde a la clave usada en el catálogo de áreas de Técnico Académico de la base de datos Otros Asuntos del Personal Académico.
fsesionCONSEJO	datetime	Fecha de la reunión de consejo.	Fecha en de la sesión de Consejo en la que se evaluará la solicitud. Restricción: La fecha debe estar definida
h_Adicionales	float	Número de Horas Adicionales	Restricción: El máximo número de horas adicionales que un técnico académico puede atender es 8, si supera esta cantidad el Secretario Auxiliar debe rechazar la solicitud.
idCategoriaHorasAdicionales	int	Clave de categoría de las horas adicionales.	La clave de categoría corresponde a la clave usada en el catálogo de categorías. Restricción: Esta clave sólo puede ser Profesor Asignatura Interino o Profesor Asignatura Definitivo.

Objeto de Información: Profesores			
Nombre del atributo	Tipo	Descripción	Observaciones
Nombre	String	Nombre del profesor.	El nombre del profesor coincide con el registro de la base de datos del Sistema Integral del Personal (SIP).

RFC	String	RFC del profesor	El RFC coincide con el registro de la base de datos del Sistema Integral del Personal (SIP).
Categoria	int	Clave de categoría de contratación del profesor.	La clave de categoría corresponde a la clave usada en el catálogo de categorías. Este registro representa la última categoría registrada en el Sistema Integral del Personal (SIP).
CategoriaLetra	String	Categoría escrita con letra con abreviaturas.	La descripción con letra y con abreviaturas de la categoría <u>correspondiente</u> a la clave registrada.
Descripcion_larga	String	Categoría escrita con letra sin abreviaturas	La descripción con letra y sin abreviaturas de la categoría correspondiente a la clave registrada.
Area	int	Clave del área académica adscripción.	La clave de área académica corresponde a la clave usada en el catálogo de áreas de la base de datos del Sistema DCDPA
AreaLetra	String	Área académica con letra	La correspondiente descripción con letra de la clave de área académica registrada.
Plantel	int	Clave del plantel de adscripción	La clave del plantel corresponde a la clave usada en el catálogo de planteles de la base de datos del Sistema DCDPA.
PlantelLetra	String	Plantel con letra.	La correspondiente descripción con letra de la clave de plantel de adscripción registrada.
n_trabaj	String	Número de trabajador.	El número de trabajador coincide con el registro de la base de datos del Sistema Integral del Personal (SIP).
sexo	String	Género del trabajador	El género puede ser M(Masculino) y F(Femenino)

Objeto de Información: Categorías

Nombre del atributo	Tipo	Descripción	Observaciones
idCategoría	int	Clave de categoría	Clave del catálogo de categorías
Descripción_larga	String	Categoría escrita con letra sin abreviaturas	La descripción con letra y sin abreviaturas de la categoría correspondiente a la clave registrada.

Objeto de Información: Veredicto

Nombre del atributo	Tipo	Descripción	Observaciones
idSesionCONSEJO	int	Clave de la sesión de ConsejoTécnico.	El consejo técnico se reúne periódicamente para evaluar las diversas solicitudes que durante cierto período de tiempo se acumulan.
id_asunto	int	Clave del tipo de solicitud.	En este caso sólo se están capturando solicitudes para impartir horas adicionales.
Periodo	int	Semestre en el que se realiza el trámite.	El semestre se captura en el forma aaaass, ej. 201710 haría referencia al semestre 2017-1.
id_plantel	int	Clave de plantel.	La clave del plantel corresponde a la clave usada en el catálogo de planteles de la base de datos del Sistema DCDPA. Restricción: El técnico académico debe tener horas asignadas frente a grupo en el plantel en que se realiza la solicitud.
RFC	String	RFC del Técnico Académico.	El RFC debe estar registrado en el catálogo de profesores.
autorizar_consejo	bool	Autorización o rechazo de la	Autorización o Rechazo de la solicitud.

solicitud.

Objeto de Información: Oficio

Nombre del atributo	Tipo	Descripción	Observaciones
Periodo	int	Semestre en el que se realiza el trámite.	El semestre se captura en el forma aaaass, ej. 201710 haría referencia al semestre 2017-1.
ResponsableNombre	String	Nombre del responsable que firmará la solicitud.	El nombre del responsable corresponde al nombre del funcionario que firma la aprobación o rechazo de la solicitud. Restricción: Si el nombre de responsable no ha sido capturado para el período especificado, no permitirá la generación del oficio correspondiente.
ResponsableCargo	String	Cargo del responsable que firmará la solicitud.	El cargo del responsable corresponde al cargo del funcionario que firma la aprobación o rechazo de la solicitud. Restricción: Si el cargo del responsable no ha sido capturado para el período especificado, no permitirá la generación del oficio correspondiente.
ccp[1..4]	String	Con copia para[1..4]	Campo para almacenar a los funcionarios involucrados para que aparezcan en la sección "con copia para" del documento.
Siglas	String	Siglas que aparecen en el pie de página del oficio.	Estas siglas corresponden con las iniciales del secretario del Consejo Técnico.

Diccionario de datos de la ejecución del flujo de trabajo.

Objeto de Información: Vista de Instancia

Nombre del atributo	Tipo	Descripción
Instancelid	Identificador único de instancia	El ID de la instancia de un flujo de trabajo.
PendingTimer	DateTime	Indica que el flujo de trabajo esta bloqueado en una actividad <i>Delay</i> y que será reanudado después de que el temporizador expire.
CreationTime	DateTime	Indica cuándo fue creado el flujo de trabajo.
LastUpdatedTime	DateTime	Indica la última vez que el flujo de trabajo fue persistido en la base de datos.

ServiceDeploymentId	BigInt	Actua como clave foránea a la vista ServiceDeployments. Si la instancia de flujo de trabajo actual es una instancia de un servicio hospedado en web, entonces esta columna tiene un valor, de otra manera el valor es NULL.
SuspensionExceptionName	Nvarchar(450)	Indica el tipo de excepción que causó que el flujo de trabajo se suspendiera.
SuspensionReason	Nvarchar(max)	Indica porqué la instancia de flujo de trabajo fue suspendida. Si una excepción causo la suspensión, entonces esta columna contendrá el mensaje asociado con la excepción. Si la instancia fue manualmente suspendida entonces esta columna contendrá la razón especificada por el usuario para suspender la instancia.
ActiveBookmarks	Nvarchar(max)	Si la instancia de flujo de trabajo esta inactive, esta propiedad indica en qué marcadores (bookmarks) la instancia está bloqueada. Si la instancia no está inactiva, entonces esta columna es NULL.
CurrentMachine	Nvarchar(128)	Indica el nombre de la computadora que en ese momento tienen la instancia del flujo de trabajo en memoria.
LastMachine	Nvarchar(450)	Indica la última computadora que cargó la instancia del flujo de trabajo.
ExecutionStatus	Nvarchar(450)	Indica el estado actual de ejecución del flujo de trabajo. Los posibles estados son: Executing, Idle, Closed.
IsInitialized	Bit	Indica si la instancia de flujo de trabajo ha sido inicializada. Una instancia inicializada de flujo de trabajo es aquella que ha sido persistida al menos una vez
IsSuspended	Bit	Indica si la instancia de flujo de trabajo ha sido suspendida.
IsCompleted	Bit	Indica si la instancia de flujo de trabajo ha finalizado su ejecución
IdentityName	Nvarchar(max)	El nombre de la definición de flujo de trabajo.

4.1.8 Diagrama de clases

La figura C4-6 muestra el modelo conceptual inicial con los objetos de información representados como clases.

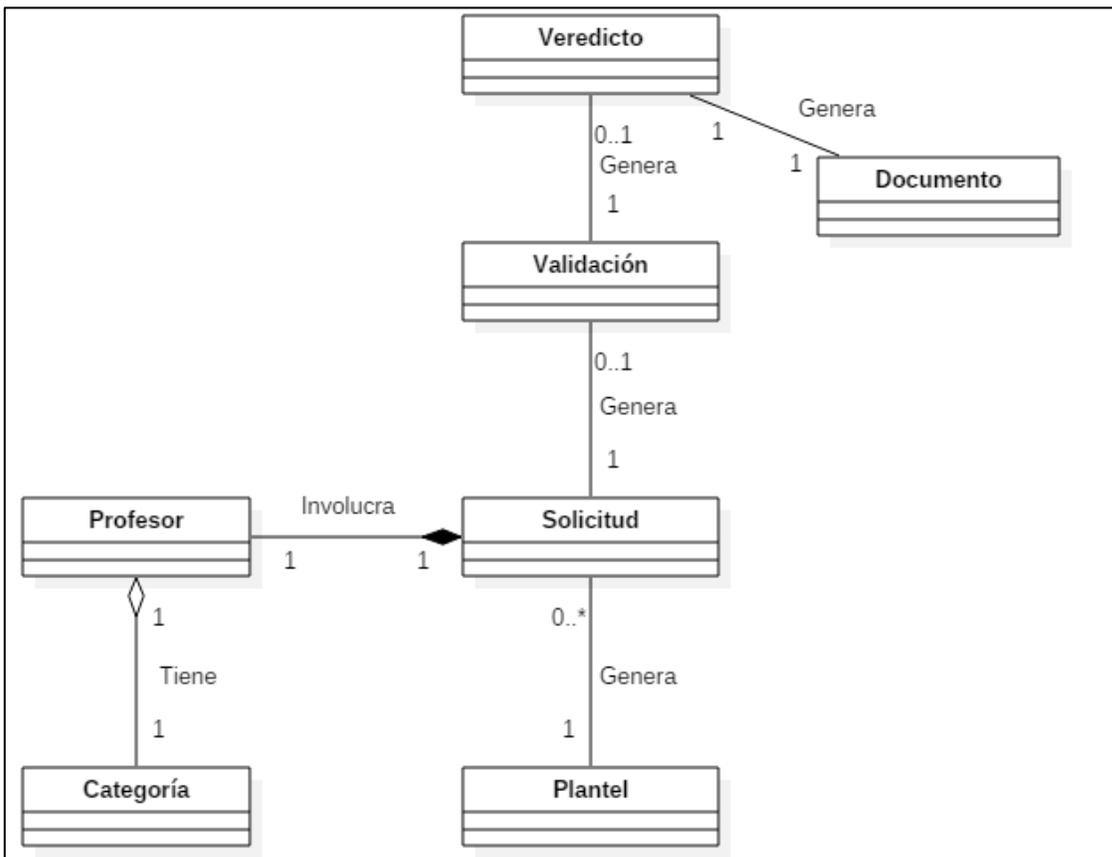


Figura C4-6 Modelo Conceptual Inicial

La figura C4-7 ilustra el refinamiento del modelo conceptual. En este diagrama se incluyen las clases generadas por la información recibida de los servicios Web y que el objeto de información *Validación* utiliza.

El tipo de solución a implementar implica la creación de flujo de trabajo semiautomático y asíncrono, semiautomático porque existen actividades que esperan alguna interacción con alguno de los actores, asíncrono porque esta verificación puede tomar días, semanas o meses antes de que el actor involucrado concluya la actividad y permita que el flujo de trabajo continúe. Estas actividades también son representadas en el diagrama como clases que guardan una relación de composición con las actividades los objetos de datos *verificación* y *veredicto*.

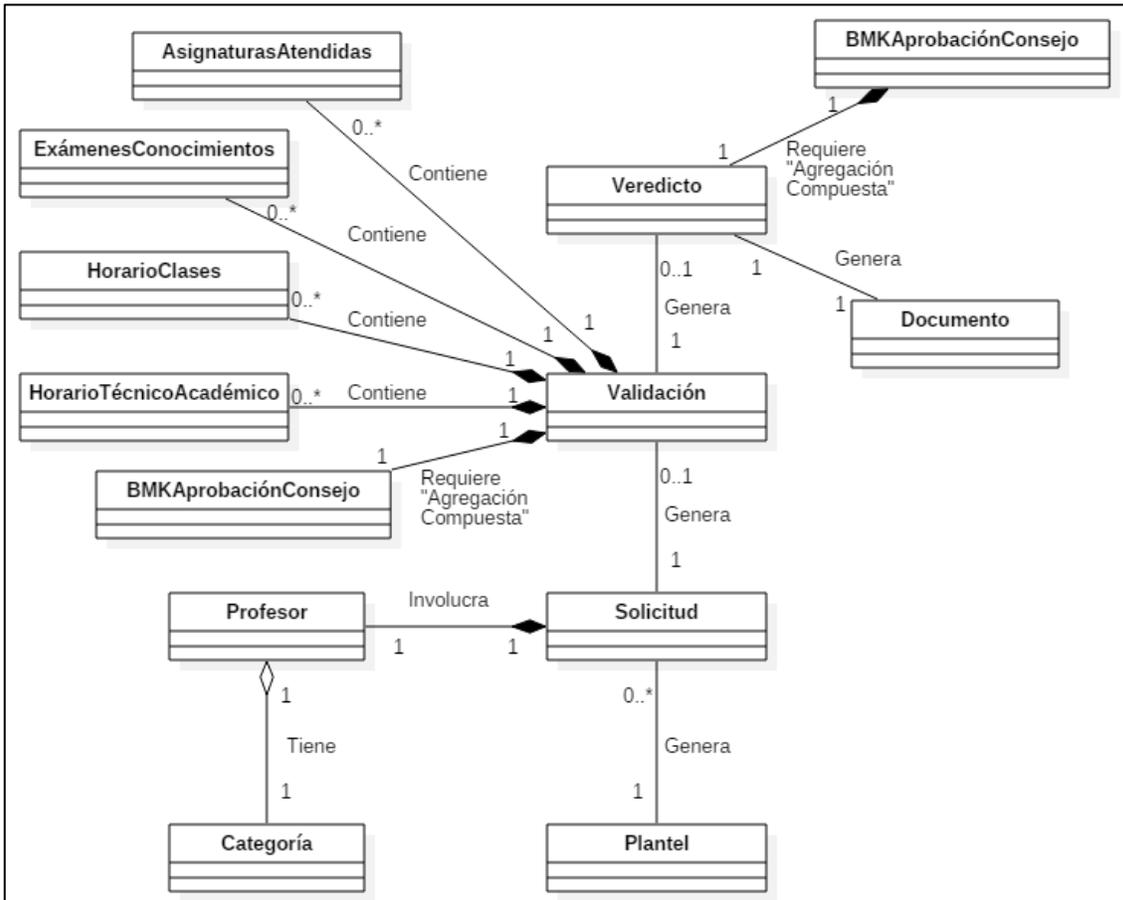


Figura C4-7 Diagrama de clases

4.1.9 Descripción de la propuesta de solución

Para la automatización de este proceso de negocio se propone una SOA que orqueste las actividades y servicios necesarios a través de un flujo de trabajo que considere las actividades de captura de solicitudes en los planteles, la validación de solicitudes por parte del secretario auxiliar correspondiente y un módulo de aprobación de solicitudes para el Consejo Técnico. La información necesaria para realizar la validación de las solicitudes proviene de los sistemas DCDPA y SPAC-E, exponiendo las funcionalidades requeridas a través de servicios Web.

La actividad aprobación de solicitudes marcará el final de cada instancia del flujo de trabajo con la aprobación o no aprobación de cada caso puesto a consideración.

Fase 2

4.2 Implementación de los servicios Web

Recordemos que la información necesaria para la validación de las solicitudes recibidas se obtiene de los sistemas DCDPA y SPAC-E. Esta información estará expuesta como servicio Web.

A continuación se describe cada uno de los servicios Web implementados, los servicios estarán divididos en dos grupos, servicios obtenidos del sistema DCDPA y servicios obtenidos del SPAC-E.

Los datos incluidos en la descripción son: dependiendo del grupo al que pertenezca el servicio, la tecnología utilizada para su implementación, la descripción general del servicio, parámetros de entrada, campos de los registros entregados, y finalmente la URL en dónde pueden consumirse.

4.2.1 Servicios obtenidos del sistema SPAC-E

Para la implementación de servicios Web en la plataforma SPAC-E se utilizó la librería nusoap.

El consumo de estos servicio se realizó a través de establecer una referencia web en Visual Studio a través del WSDL disponible en la URL del servicio Web correspondiente.

Servicio Web: HorarioTecnicoAcademico

Descripción del Servicio

Este método entrega los registros correspondientes al horario de trabajo de un técnico académico del CCH, en un semestre específico. El horario entregado corresponde a las 40 horas de la plaza de técnico académico, este servicio no incluye el horario de grupos adicionales de clases, que el técnico académico pueda tener asignados en alguno de los planteles del CCH.

Hasta la última edición de este documento (agosto de 2016), es posible obtener los horarios de técnicos académicos adscritos a los planteles del CCH, así como de los técnicos académicos adscritos a la Dirección General del CCH, con excepción del plantel Naucalpan. Conforme se vaya incorporando este plantel en el proceso de emisión de hojas de firmas de asistencia para técnicos académicos, esta información podrá estar disponible en el sistema también.

Si no se encontró ningún horario de trabajo registrado para el técnico académico dado, el método entrega un conjunto de registros vacío.

Este método es utilizado principalmente, dentro del proceso de Solicitud de horas adicionales para técnicos académicos.

Parámetros de entrada

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ntrabajador	String	Número de trabajador del técnico académico a buscar
period	String	Periodo o semestre de trabajo del técnico académico. Este dato deberá estar en el formato utilizado por el SPAC-E, el cual es "aaaa-s"

Campos de los registros entregados

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Observaciones</i>
ntrabajador	String	Número de trabajador del técnico académico a buscar	Si se encontró un horario de trabajo del técnico académico, este dato será el mismo para todos los registros.

planteltec	String	Clave de plantel del CCH en donde el técnico académico tiene un horario de trabajo registrado	La clave de plantel entregada es utilizada por el Sistema Integral de Personal de la UNAM (SIP). Esta clave está conformada por los tres primeros caracteres de la dependencia universitaria, así como por dos caracteres que indican la subdependencia de la UNAM, en este caso el plantel del CCH correspondiente.
dia	String	Día de la semana de trabajo en el horario del técnico académico	El día viene indicado en formato numérico, donde el 1 indica el día lunes, hasta el día número 5, que indica el día viernes.
hora	String	Hora de trabajo en el día de la semana del técnico académico	La hora viene indicada en formato numérico, indicando sólo la hora sin minutos ni segundos, en un rango de 0 a 23 horas. Cada hora corresponde a una hora completa de trabajo, respecto a las 40 horas que conforman las horas de trabajo a la semana del técnico académico.

URL del servicio Web

https://space.cch.unam.mx/spac_dgcch/dcdpa/ws_otros_asuntos.php

Interfaz del servicio Web

https://space.cch.unam.mx/spac_dgcch/dcdpa/ws_otros_asuntos.php?wsdl

Servicio Web: ExámenesConocimientos

Descripción del Servicio

Este método entrega los registros históricos completos de las evaluaciones de Exámenes de Conocimientos que ha tenido un profesor. Los registros se entregan por evaluación del profesor en una asignatura determinada, por número de promoción y en algunos casos, por la distinción entre una calificación ordinaria y una posterior calificación de revisión.

Los registros entregados contienen los exámenes de todas las materias del profesor. Dentro de la aplicación receptora de estos registros se deberán filtrar los mismos para una asignatura determinada, en función de las necesidades particulares que se requieran resolver.

Aunque los registros están almacenados en el SPAC-E de forma interna por materias, la entrega de éstos por el Web Service es por asignaturas, para facilitar el uso de los registros en determinados procesos de programación. Las asignaturas de exámenes que hacen referencia al Plan de Estudios del CCH original (1971) han sido transformadas a su asignatura equivalente, correspondiente al Plan de Estudios actual (1997).

Si no se encontró ninguna evaluación de Exámenes de Conocimientos del profesor, el método entrega un conjunto de registros vacío.

Este método es utilizado principalmente, dentro del proceso de Solicitud de horas adicionales para técnicos académicos.

Parámetros de entrada

Nombre	Tipo	Descripción
ntrabajador	String	Número de trabajador del profesor a buscar

Campos de los registros entregados

Nombre	Tipo	Descripción	Observaciones
ntrabajador	String	Número de trabajador del profesor a buscar	Si se encontraron registros de evaluaciones de Exámenes de Conocimientos del profesor, este dato será el mismo para todos los registros.
asignatura	String	Clave de la asignatura de la evaluación de examen del profesor	La clave de la asignatura corresponde a una asignatura curricular, perteneciente al Plan de Estudios del CCH, vigente desde 1997. La clave también puede corresponder a una asignatura de los Departamentos no curriculares de Educación Física y Opciones Técnicas, en donde el profesor ha obtenido su definitividad.
promocion	String	Número de promoción histórica en la cual el profesor realizó el examen	Este número de promoción se maneja de manera habitual en formato numérico. En algunos casos y como casos de excepción, se han manejado números de promociones inicialmente numéricas, acompañadas de alguna variante adicional dentro de la misma promoción. Ejemplos: 10 Bis, 31A,31B, 31C, etc.
tipocalificacion	String	Clave de tipo de calificación obtenida por el profesor en el examen	Existen dos tipos de calificaciones de examen: "O", que indica una calificación obtenida de forma ordinaria, y "R", que significa una calificación de revisión de examen solicitada por el profesor.
calificacionexamen	String	Calificación numérica y/o con letra obtenida del examen	En las promociones organizadas en los últimos años se han asentado calificaciones numéricas de 0 al 10, así como calificaciones de NP, que significa que el profesor no se presentó a realizar su examen. En algunas actas de exámenes de promociones históricas, quedaron asentadas algunas calificaciones indicadas con letra (MB, B, S), así como otras con letras, pero sin especificar de forma directa su equivalencia a una evaluación numérica (Aprobado, Apto, etc.)
fechaexamen	String	Fecha de aplicación del examen	La fecha está indicada en formato aaaa-mm-dd. Dentro de la aplicación en donde se reciban estos registros se deberá transformar este formato por otro personalizado, en función de las necesidades que se requieran resolver.
status	String	Estado o situación de aprobación del examen	Este campo puede tener dos posibles valores: Aprobado y No aprobado. Se agrega este campo de forma adicional, para resolver la ambigüedad de si en función de una calificación dada, el profesor aprobó o no el Examen de Conocimientos en la asignatura indicada. Para promociones históricas realizadas antes de la promoción 12, la calificación mínima aprobatoria para un examen era de 6. A partir de la promoción 12 en adelante, la calificación mínima aprobatoria es 8. Esta situación de aprobación del examen se indica también para calificaciones asentadas en forma de letras.

URL del servicio Web

https://space.cch.unam.mx/spac_dgcch/dcdpa/ws_otros_asuntos.php

Interfaz del servicio Web

https://space.cch.unam.mx/spac_dgcch/dcdpa/ws_otros_asuntos.php?wsdl

Servicio Web: HorarioGruposProfesor

Descripción del Servicio

Este método entrega los registros de horarios de grupos, que tiene asignados un profesor para dar clases, en todos los planteles, en un semestre determinado. Cada registro entregado representa la intersección de un día y una hora de clases, de un grupo asignado a un profesor, en un plantel determinado.

Si no se encontraron grupos asignados al profesor en el semestre dado, el método entrega un conjunto de registros vacío.

Este método es utilizado dentro de los procesos de Solicitud de horas adicionales para profesores de Carrera y para técnicos académicos.

Parámetros de entrada

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ntrabajador	String	Número de trabajador del profesor a buscar
period	String	Periodo o semestre escolar. Este dato deberá estar en el formato utilizado por el SPAC-E, el cual es "aaaa-s"

Campos de los registros entregados

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Observaciones</i>
period	String	Periodo o semestre escolar del grupo asignado	Si se encontraron grupos asignados al profesor en el semestre dado, este dato será el mismo para todos los registros.
plantel	String	Clave de plantel del grupo que tiene asignado el profesor	La clave de plantel entregada es utilizada por el Sistema Integral de Personal de la UNAM (SIP). Esta clave está conformada por los tres primeros caracteres de la dependencia universitaria, así como por dos caracteres que indican la subdependencia de la UNAM, del plantel del CCH.
asignatura	String	Clave de la asignatura del grupo asignado al profesor	La clave de la asignatura corresponde a una asignatura curricular, perteneciente al Plan de Estudios del CCH, vigente desde 1997. Esta clave puede corresponder también a una asignatura no curricular, del Departamento de Opciones Técnicas del CCH.

grupo	String	Número de grupo asignado al profesor	Para asignaturas curriculares pertenecientes al Plan de Estudios del CCH, este campo es la parte numérica del grupo. Para asignaturas no curriculares del Departamento de Opciones Técnicas, este campo es la clave alfanumérica de identificación del grupo.
seccion	String	Sección del grupo asignado al profesor	Si la asignatura del grupo contempla la división del mismo en dos secciones de grupo, entonces este campo contendrá la clave de sección correspondiente al grupo seccionado (sección A o B). Para grupos sin división de secciones o para grupos del Departamento de Opciones Técnicas, este dato aparecerá vacío o en blanco.
ntrabajador	String	Número de trabajador del profesor a buscar	Si se encontraron grupos asignados al profesor en el semestre dado, este dato será el mismo para todos los registros.
dia	String	Día de la semana de impartición de clases del grupo asignado	El día viene indicado en formato numérico, donde el 1 indica el día lunes, hasta el día número 6, que indica el día sábado de la semana de clases. Los grupos de asignaturas curriculares se imparten en los planteles de lunes a viernes, y algunos grupos no curriculares del Departamento de Opciones Técnicas pueden tener un horario que incluya al sábado como día de clases.
hora	String	Hora de impartición de clases en el día del grupo asignado	La hora viene indicada en formato numérico, indicando sólo la hora sin minutos ni segundos, en un rango de 0 a 23 horas. Cada hora corresponde a una hora completa de impartición de clases, respecto al total de horas que conforman el grupo asignado.

URL del servicio Web

https://space.cch.unam.mx/spac_dgcch/dcdpa/ws_otros_asuntos.php

Interfaz del servicio Web

https://space.cch.unam.mx/spac_dgcch/dcdpa/ws_otros_asuntos.php?wsdl

Servicio: RegistroKardexAsignatura

Descripción

Este método entrega el último registro detectado del profesor, en su historial de movimientos administrativos (kárdex) proporcionado por la Dirección General de Personal de la UNAM, con información relativa a la categoría como profesor de Asignatura del profesor, el número de horas en el contrato, así como la clave de la asignatura relacionada con la contratación.

Si no se encontró ningún registro del profesor, el método entrega un registro con todos los campos vacíos.

Este método es utilizado principalmente, dentro del proceso de Solicitud de horas adicionales para técnicos académicos y profesores de Carrera.

Parámetros de entrada

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ntrabajador	String	Número de trabajador del profesor a buscar

Campos de registro entregado

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Observaciones</i>
ntrabajador	String	Número de trabajador del profesor a buscar	En el código de programación, se debe validar que este campo contenga el mismo valor que el parámetro de entrada del método, para verificar que el Webservice sí entregó algún registro.
plantel	String	Clave de plantel en donde se detectó un contrato como profesor de Asignatura del profesor	La clave de plantel entregada es utilizada por el Sistema Integral de Personal de la UNAM (SIP). Esta clave está conformada por los tres primeros caracteres de la dependencia universitaria, así como por dos caracteres que indican la subdependencia de la UNAM, del plantel del CCH.
categoria	String	Clave de categoría de contratación del profesor	La clave de categoría entregada corresponde a la clave usada en el catálogo de categorías, de la base de datos del Sistema del DCDPA.
horas	String	Número de horas de último contrato de profesor	El número de horas corresponde al contrato del profesor como profesor de Asignatura. Este número de horas es adicional a las 40 horas de la plaza de Carrera del profesor.
asignatura	String	Clave de la asignatura de contrato del profesor	La clave de la asignatura corresponde a una asignatura curricular, perteneciente al Plan de Estudios del CCH, vigente desde 1997.

URL del servicio Web

https://space.cch.unam.mx/spac_dgcch/dcdpa/ws_otros_asuntos.php

Interfaz del servicio Web

https://space.cch.unam.mx/spac_dgcch/dcdpa/ws_otros_asuntos.php?wsdl

4.2.2 Servicios obtenidos del sistema DCDPA

Para la implementación de servicios Web en la plataforma DCDPA se utilizó WFC como método para exponer servicios en formato *svg*, los servicios fueron configurados en el IIS del servidor que hospeda al sistema DCDPA.

El consumo de estos servicios se realizó a través de una referencia web en Visual Studio a través del WSDL disponible en la URL del servicio Web correspondiente.

Servicio Web: DefinitividadesAsignaturaProfesor

Descripción

Este método entrega el último registro encontrado en el historial de movimientos administrativos del profesor (kárdex), proporcionados por la Dirección General de personal de la UNAM, en donde se indican las asignaturas del Plan de Estudios del CCH, en donde los profesores han realizado diversos trámites administrativos de contrataciones como profesor de Asignatura con horas definitivas, así como la última categoría registrada, para cada una de estas definitividades.

Si no se encontró ningún registro de definitividad del profesor, el método entrega un conjunto de registros vacío.

Este método es utilizado principalmente, dentro del proceso de Solicitud de horas adicionales para técnicos académicos.

Parámetros de entrada

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ntrabajador	String	Número de trabajador del profesor a buscar

Campos de los registros entregados

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Observaciones</i>
ntrabajador	String	Número de trabajador del profesor a buscar	Si se encontraron registros de definitividades del profesor, este dato será el mismo para todos los registros.
asignatura	String	Clave de la asignatura en donde el profesor es definitivo	La clave de la asignatura corresponde a una asignatura curricular, perteneciente al Plan de Estudios del CCH, vigente desde 1997. La clave también puede corresponder a una asignatura de los Departamentos no curriculares de Educación Física y Opciones Técnicas, en donde el profesor ha obtenido su definitividad. El dato de asignatura proporcionado se refiere a una asignatura impartida en un semestre dado, y no a un conjunto de asignaturas agrupadas en común, impartidas en diferentes semestres (materia).
categoria	String	Clave de categoría relacionada con el último trámite de definitividad del profesor	La clave de categoría entregada corresponde a la clave usada en el catálogo de categorías, de la base de datos del Sistema del DCDPA. En algunos casos la categoría indicada puede variar, respecto a la información almacenada en tabla materias_definitividad de la base de datos del DCDPA, en función del tiempo de otorgamiento de definitividad y/o promoción por el Consejo Técnico del CCH, así como a los tiempos de duración de los trámites administrativos del profesor, entre otras causas.

URL del servicio Web

http://dcdpa.cch.unam.mx/otrosasuntos/ws_otros_asuntos.svg

Interfaz del servicio Web

http://dcdpa.cch.unam.mx/otrosasuntos/ws_otros_asuntos.svg?wsdl

Fase 3

4.3 Modelado del Flujo de Trabajo

En esta fase el diagrama de proceso UML es representado como un flujo de trabajo a través del diseñador de flujos de trabajo de Visual Studio. La figura 3-6 describe las actividades para completar el proceso de negocio “procesamiento de solicitudes”, estas actividades corresponden con las actividades descritas en el diagrama de proceso de la primera fase de este modelo.

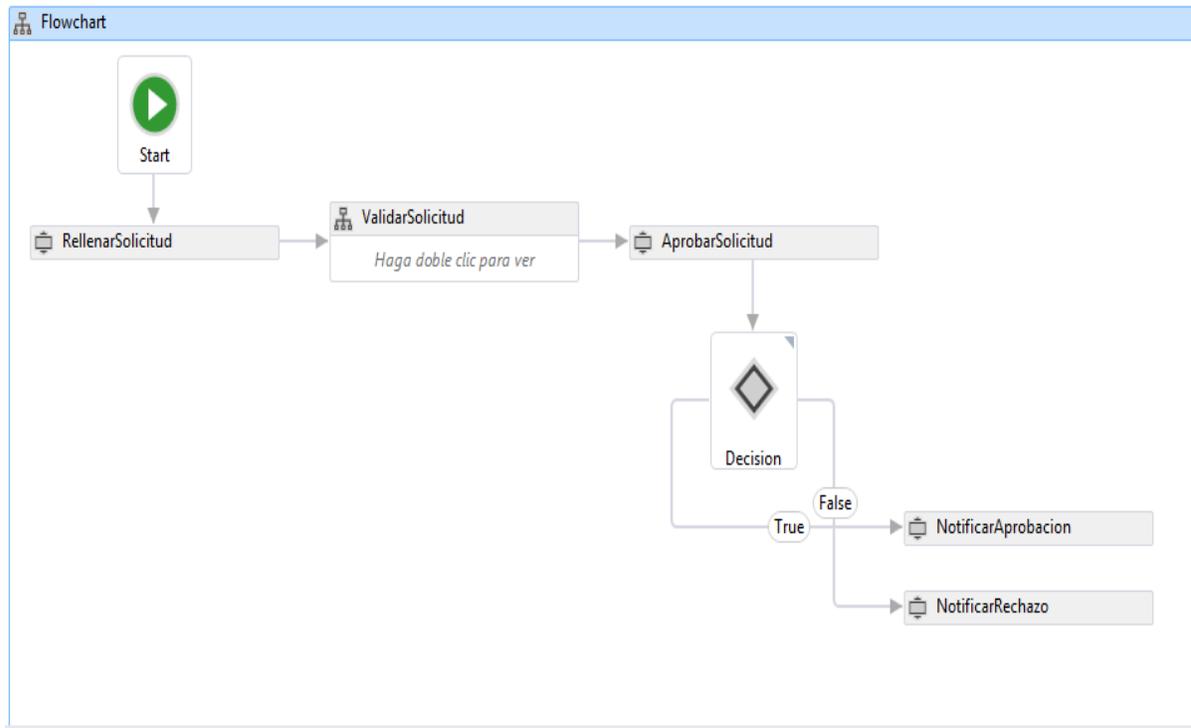


Figura C4-6 Flujo de trabajo del proceso “Procesamiento de Solicitudes”

Como se observa en la figura C4-6, en el diseñador de flujos de trabajo no es posible representar las calles que delimitan las actividades a realizar por cada rol.

La composición de servicios WCF y nusoap está representada en el flujo de trabajo a través de actividades de invocación de servicios, para diferenciar entre la invocación de servicios Web y las actividades de otra índole, antepongo el prefijo WS al nombre de la actividad que representa el consumo de un servicio.

Para actividades de una longitud considerable como la actividad *ValidarSolicitud*, se implementa una subactividad que describe a detalle la actividad de nivel superior correspondiente. Para acceder a dicha subactividad es necesario hacer doble clic sobre la actividad de nivel superior.

La figura C4-7 muestra el drilldown de la actividad *ValidarSolicitud*.

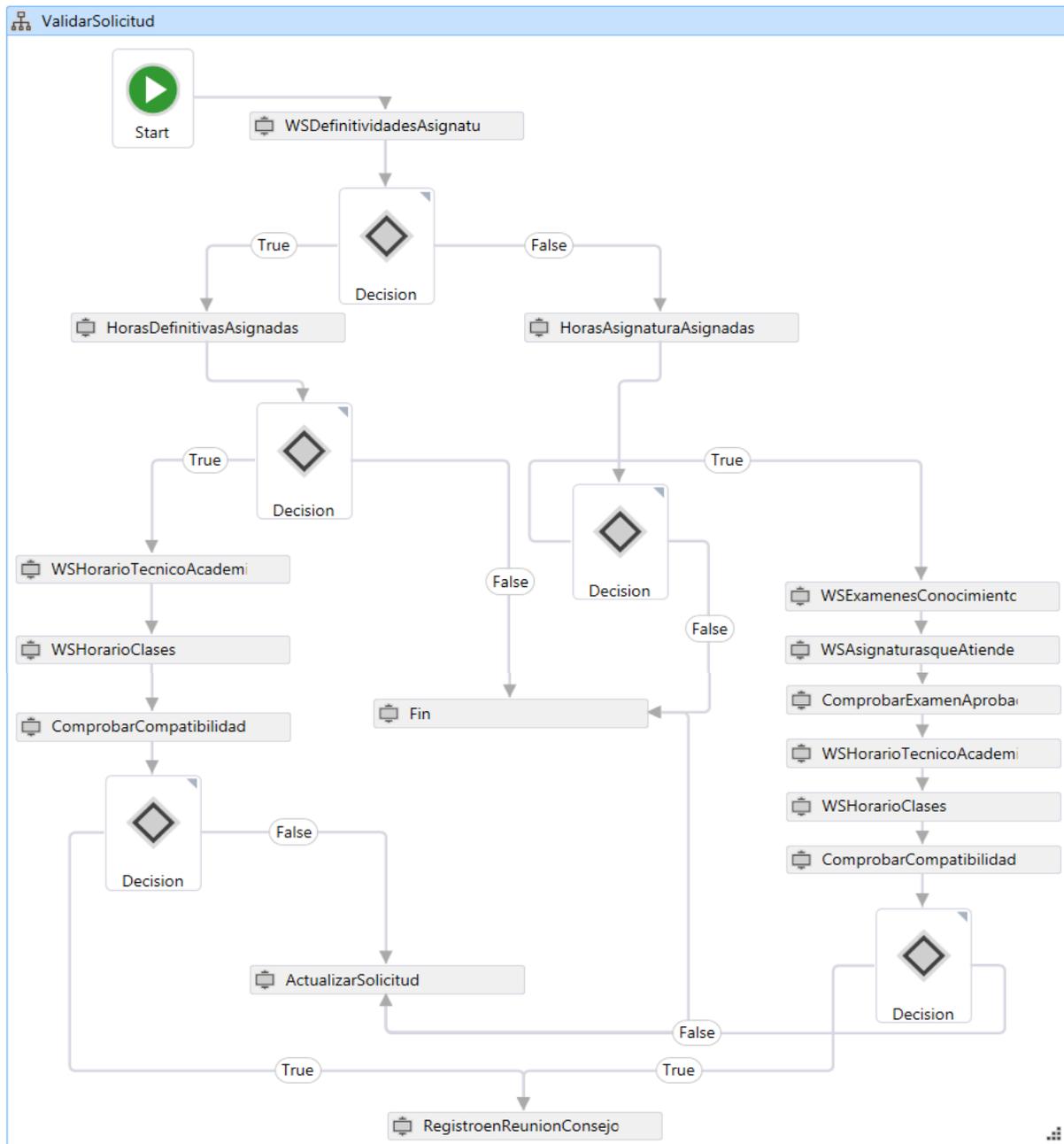


Figura C4-7 Subactividad VisualizarSolicitud

Fase 4

4.4 Implementación de la Interfaz de Usuario

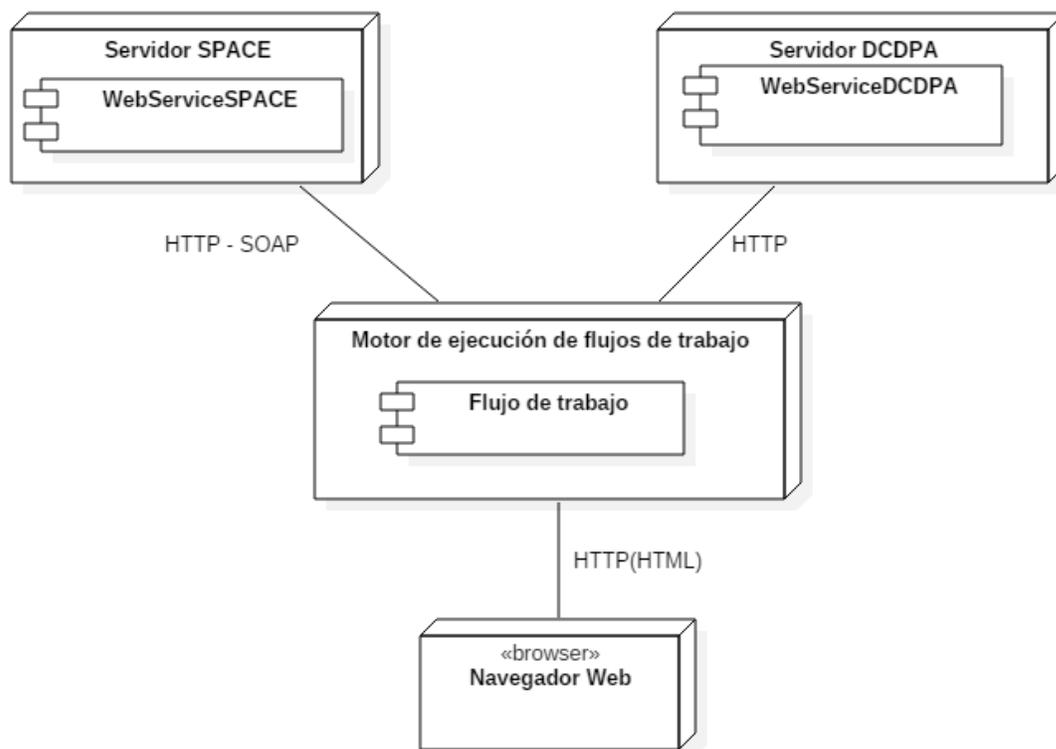
La interacción de los usuarios con el proceso de negocio se realiza a través de una aplicación ASP.NET, los formularios que conforman el proceso son:

- Rellenar Solicitud
- Validar Solicitud
- Aprobar Solicitud

La notificación de aprobación o rechazo de las solicitudes se realiza a través del envío del oficio correspondiente a los planteles.

Fase 5

4.5 Despliegue



C4-8 Diagrama de despliegue

4.5.1 Descripción del hardware de despliegue:

Servidor SPACE

Hardware	
Procesador	Intel Quad Core Xeon E5-2650 1.7GHz
Memoria RAM	32GB
Disco Duro	4 x 1TB SATA RAID Controller
Lan	Gigabit

Software	
Servidor Web	Servidor Apache HTTP
Sistema operativo	CentOS 7
Lenguaje de programación	PHP (Hypertext Preprocessor)

Servidor DCDPA

Hardware	
Procesador	Intel Dual Core Xeon 2.4 GHz
Memoria RAM	16GB
Disco Duro	4 x 500GB SATA RAID Controller
Lan	Gigabit

Software	
Servidor Web	Internet Information Server (IIS)
Sistema operativo	Windows Server 2008 R2
Lenguaje de programación	ASP (Active Server Pages)

Fase 6

4.6 Seguimiento

En este punto de la implementación se describe una herramienta que se encarga de coordinar los procesos de negocio producto de la composición de servicios WCF y nusoap a través de un flujo de trabajo que consta de una interfaz que permite a los usuarios no-programadores monitorear instancias en ejecución.

4.6.1 Funcionalidad de la aplicación

1. La habilidad de seleccionar y elegir de una lista de instancias en ejecución de un flujo de trabajo aquella cuyo estado debería ser mostrado.
Ejemplos de estado para cada actividad son:
2. La habilidad de mostrar, a través de una imagen representativa, el estado de una instancia específica de un flujo de trabajo.
3. Menús contextuales para controlar las funcionalidades de navegación (zooming, scrolling drillup y drilldown).

4.6.2 Monitoreo de una instancia de flujo de trabajo a través del depurador de flujos de trabajo

La figura C4-8 muestra una instancia del flujo de trabajo en ejecución, el recuadro de color amarillo define la actividad en ejecución, cada vez que el flujo encuentre un marcador este se detendrá a esperar la entrada del actor correspondiente.

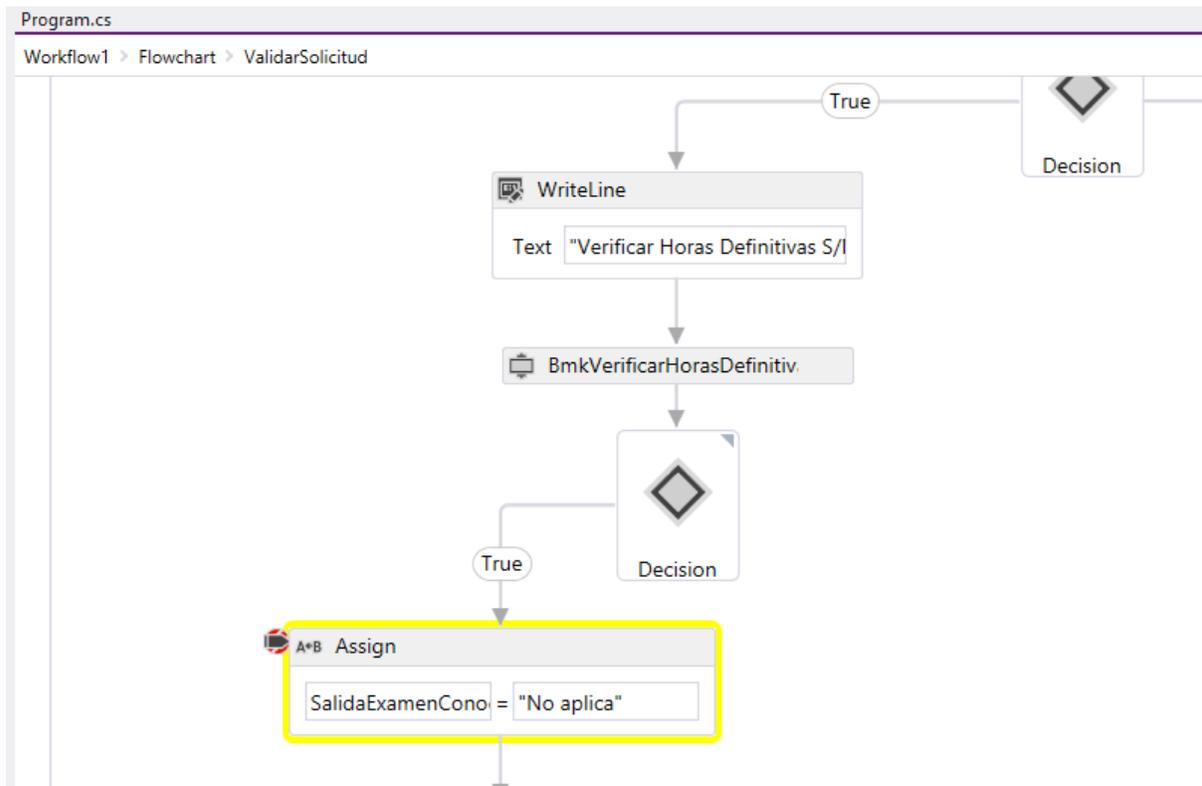


Figura C4-8 Instancia de flujo de trabajo en ejecución

Evaluación y Beneficios de la implementación del modelo

El modelo propuesto establece una serie de pasos ordenados para la implementación de una SOA y describe un modelo abstracto SOA como marco de referencia para las fases definidas.

Las herramientas elegidas así como el modelo abstracto en el que se basa la implementación pueden intercambiarse por una plataforma distinta, por ejemplo aquellas que utilizan BPEL como lenguaje para composición de servicios y un BPMS que sustituye al diseñador de flujos de trabajo de Visual Studio.

Para verificar la efectividad del modelo, el caso de prueba proyecta un conjunto de beneficios que pueden obtenerse con su aplicación:

- Facilita la definición de procesos de negocio flexibles y adaptables a cambios organizacionales.
- Es apto para la reingeniería de los procesos de la organización
- Favorece el aprovechamiento de los activos de IT definiendo un modelo de integración que facilite el acceso desacoplado a capacidades existentes.
- Favorece la comunicación entre procesos de negocio y su integración en un software.

La solución propuesta no requiere UDDI dado que los servicios Web son utilizados al interior de la organización y no se hacen públicos para otras organizaciones.

El manejo de errores no está considerado en este modelo, si el servidor que hospeda algún servicio Web no está disponible, el proceso de negocio genera una excepción.

Conclusiones

1. Una Implementación SOA en pequeña escala representa una parte importante en el proceso de aprendizaje e introducción de SOA en una PyME, evita la adquisición de infraestructura costosa en proyectos piloto y muestra utilidad en un tiempo considerablemente corto.
2. SOA sólo debe ser utilizada dónde sea la mejor manera para reunir los requisitos de negocio, y éstos deben ser los que guíen el proyecto antes incluso que la solución técnica (aproximación Middle Out).
3. Desarrollar una estrategia SOA que se extienda a lo largo de una organización no es una actividad trivial y no debería ser tratada como un esfuerzo de corta duración que baste con realizarse una sola vez. SOA debe ser un intento por alcanzar un nivel más alto de agilidad para la organización completa que favorezca la respuesta expedita a las necesidades de negocio.
4. Muchas organizaciones por sus dimensiones naturalmente no implementarán completamente cada una de las capacidades definidas en las capas de una Arquitectura de Referencia, sin embargo no reconocer los límites de una Arquitectura puede llevar a decisiones pobres concernientes a bloques de construcción importantes cuya ausencia pueda impactar la habilidad del modelo para alcanzar una mayor madurez.

Referencias

1. Picón, Fontana, Martín. *Integración de Procesos de Negocio Aplicando Servicios Web, Un Modelo para el BPI en el dominio de las PyMEs*. Universidad Nacional de la Patagonia Austral. Santa Cruz Argentina.
2. Reyes, José Villasís, et al. *Metodología para el análisis diseño e implementación de procesos con tecnología BPM*. Escuela Politécnica del Ejército, Ecuador.
3. Bazán Patricia (2009). *Un modelo de integrabilidad con SOA y BPM*. Facultad de Informática Universidad Nacional de la Plata.
4. The Open Group, Steve Nunn, et al. (2009) "*The SOA Source Book*", United Kingdom, recuperado de <http://www.opengroup.org/soa/source-book/soa/soa.htm>
5. Schulte, W. R. and Y. V. Natis (1996). "*Service Oriented*" Architectures, Part 1 Gartner.
6. Erl, T. (2005). *Service-oriented architecture: concepts, technology, and design*. Prentice Hall/Pearson PTR.
7. Erl, T. (2007). *SOA: principles of service design*. Upper Saddle River, NJ, USA, Prentice Hall Press.
8. D. Linthicum (2007). *How much will your SOA Cost?* SOAInstitute. Recuperado de: <http://www.soainstitute.org/articles/article/article/how-much-will-your-soa-cost.html>
9. Zheng Li, Jacky Keung. *Software Cost Estimation Framework for Service-Oriented Architecture Systems using Divide-and-Conquer Approach*. School of Computer Science and Engineering. Sydney, Australia.
10. W3C (2004), "*Web Service Glossary*" Recuperado de <https://www.w3.org/TR/ws-gloss/>
11. Fayad, M. and D. C. Schmidt (1997). *Object-oriented application frameworks*. Communications of the ACM 40 (10): 32–38.
12. Wei Tan, MengChu Zhou(2013). *Business and Scientific Workflows: A Web Service-Oriented Approach, First Edition*. John Wiley & Sons, Inc. The Institute of Electrical and Electronics Engineers, Inc.
13. Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, "Servicios Web y SOA" 2001.
14. Workflow Management Coalition, *The Workflow Reference Model*, WFMC-TC-1003, 1995. Recuperado de: <http://www.wfmc.org/standards/docs/tc003v11.pdf>
15. Cugola, G., Ghezzi, C. & Pinto, L.S. *Computing (2012) 94: 579*. doi:10.1007/s00607-012-0194-z
16. Angel Lagares, Florian Daniel, Boualem Benatallah(2016), *Web Service Composition: A Survey of Techniques and Tools*, ACM New York, NY, USA Vol 48 Issue 3, February 2016: 20 – 21
17. Atluri, V., Chun, S.A., Mukkamala, R. et al. (2017) *Distrib Parallel Databases*
18. Chafle G., Chandra S., et al. *Decentralized Orchestration of Composite Web Services*. IBM Research Laboratory, New Delhi India 2005.
19. Liu, J., Pacitti, E., Valduriez, P. et al. *J Grid Computing (2015) 13: 457*. doi:10.1007/s10723-015-9329-8
20. *MSDN Library. Chapter 1: Service Oriented Architecture (SOA)*. Recuperado de: <https://msdn.microsoft.com/en-us/library/bb833022.aspx>.

21. MSDN Library. *Chapter 4: Data*. Recuperado de: <https://msdn.microsoft.com/en-us/library/bb833025.aspx>
22. Ortín, M. Molina, Moros, et al. *El Modelo del Negocio como base del Modelo de Requisitos*, Universidad de Murcia, Murcia, España.
23. MSDN Library. *¿Qué es Windows Communication Foundation?* Recuperado de [https://msdn.microsoft.com/es-es/library/ms731082\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/ms731082(v=vs.110).aspx)
24. MSDN Library. *Guía de interoperabilidad de los protocolos de servicios web*. Recuperado de [https://msdn.microsoft.com/es-es/library/ms734776\(v=vs.110\).a spx](https://msdn.microsoft.com/es-es/library/ms734776(v=vs.110).aspx)