

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD # 5.

**MIDAC: MODELO PARA EL DESARROLLO DE APLICACIONES COMPUESTAS
BASADAS EN ARQUITECTURAS ORIENTADAS A SERVICIOS**

Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas.

ORESTES FEBLES DÍAZ

La Habana

2012

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD # 5.



**MIDAC: MODELO PARA EL DESARROLLO DE APLICACIONES COMPUESTAS
BASADAS EN ARQUITECTURAS ORIENTADAS A SERVICIOS**

Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas.

Autor:

Ing. Orestes Febles Díaz

Tutores:

Dra. Vivian Estrada Sentí

Dr. Juan Pedro Febles Rodríguez

La Habana

2012

AGRADECIMIENTOS

A mis tutores, que se han convertido en mis segundos padres, que me han brindado mucho más que su experiencia y me han hecho parte de un núcleo familiar repleto de virtudes.

A mi esposa, dueña de mi corazón y mis pensamientos. A su familia, que es también un poco mía, por ayudarme tanto en el transcurso de esta investigación.

A mi familia, principalmente a mis padres y a mi hermana, que son mi orgullo, mi apoyo, mi razón de superación y mi gran satisfacción.

A mis compañeros del CDAE, del PEFCI, directivos de la Facultad y la UCI que han contribuido de una forma u otra al desarrollo de este trabajo, y que hacen de esta universidad, un lugar incomparable.

A Margarita, por sus señalamientos precisos y por su manera dulce de explicarlos.

DEDICATORIA

A mi abuelo Pepe, hijo ilustre de Pueblo Nuevo, Doctor en buenas acciones...

SÍNTESIS

Las organizaciones buscan responder de una manera ágil a las crecientes necesidades del mercado. Entre las dificultades actuales se encuentra la falta de interoperabilidad en las soluciones para resolver las crecientes necesidades de integración entre sistemas. La alineación entre el negocio y las Tecnologías de la Información constituye un beneficio que se le atribuye a las arquitecturas orientadas a servicios. Las aplicaciones compuestas, al reutilizar componentes de software, ofrecen beneficios en busca de la agilidad organizacional, pero su desarrollo, su despliegue y su mantenimiento resulta complejo. Con el propósito de facilitar el proceso de composición de aplicaciones se propone un modelo que posee una infraestructura tecnológica para el desarrollo y el despliegue de aplicaciones compuestas. El modelo tiene como objetivo describir y representar los diferentes componentes de carácter tecnológico, de conocimiento y sus interrelaciones con un componente de selección inteligente de información que apoya la toma de decisiones en el proceso de composición de aplicaciones en entornos orientados a servicios. El modelo está dirigido a lograr una mejora continua como sistema integral aportando interoperabilidad y promoviendo el uso de estándares en el desarrollo de aplicaciones compuestas. La validez del modelo fue comprobada con la aplicación de métodos cuantitativos y cualitativos.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO I – MARCO TEÓRICO REFERENCIAL DE LAS APLICACIONES COMPUESTAS EN ENTORNOS ORIENTADOS A SERVICIOS.	11
1.1 EVOLUCIÓN DE LAS APLICACIONES INFORMÁTICAS	11
1.2 LA ORIENTACIÓN A SERVICIOS	13
1.3 APLICACIONES COMPUESTAS.	25
1.4 INFRAESTRUCTURA TECNOLÓGICA PARA DESARROLLAR Y DESPLEGAR APLICACIONES COMPUESTAS BASADAS EN SOA.....	28
1.5 SISTEMAS BASADOS EN EL CONOCIMIENTO.	42
CONCLUSIONES DEL CAPÍTULO.	45
CAPÍTULO II – MODELO PARA EL DESARROLLO DE APLICACIONES COMPUESTAS.....	46
2.1 MODELOS Y TRABAJOS ACTUALES RELACIONADOS CON LA COMPOSICIÓN DE APLICACIONES BASADAS EN SERVICIOS.....	46
2.2 DIAGNÓSTICO DEL DESARROLLO DE APLICACIONES COMPUESTAS EN CUBA.	48
2.3 LA ORIENTACIÓN A SERVICIOS Y LA COMPOSICIÓN DE APLICACIONES EN LA UCI.....	55
2.4 MODELO CONCEPTUAL PARA EL DESARROLLO DE APLICACIONES COMPUESTAS, MIDAC.....	59
2.5 PRINCIPIOS, CUALIDADES, PREMISAS Y COMPONENTES PARA EL DESARROLLO DEL MODELO PARA APLICACIONES COMPUESTAS BASADO EN SOA.	61
2.6 ESTRUCTURA DEL MODELO PARA EL DESARROLLO DE APLICACIONES COMPUESTAS, MIDAC.	63
2.7 INFRAESTRUCTURA TECNOLÓGICA PARA LA APLICACIÓN DEL MODELO.	70
2.8 INDICACIONES METODOLÓGICAS PARA LA IMPLANTACIÓN DEL MODELO PROPUESTO.	74
CONCLUSIONES DEL CAPÍTULO.	75
CAPÍTULO III – INSTRUMENTACIÓN Y EVALUACIÓN DEL MODELO	76
3.1 INSTRUMENTACIÓN DEL MODELO.	76
3.2 VALIDACIÓN DEL MODELO.....	81
CONCLUSIONES DEL CAPÍTULO	97
CONCLUSIONES GENERALES.....	99
RECOMENDACIONES.....	100
REFERENCIAS BIBLIOGRÁFICAS	101
ANEXOS.....	113

INTRODUCCIÓN

La ciencia y la tecnología son factores fundamentales en el desarrollo de las sociedades actuales. La aparición de Internet y las comunicaciones de alta velocidad son ejemplos palpables que han transformado de manera significativa la forma de manejar la información. Paralelo a su evolución, han surgido nuevos negocios y con ellos organizaciones que buscan responder de una manera ágil a las crecientes necesidades del mercado. Las organizaciones se enfrentan al reto de aumentar su agilidad de innovación y respuesta a los constantes cambios en los negocios y a los ambientes tecnológicos heterogéneos con la misión de reducir los gastos en sus Tecnologías de la Información (TI).

Las empresas esperan más de la informática de lo que esperaban antes por lo que la necesidad de evaluar las arquitecturas correctas para los sistemas a partir de los cambios que ocurren se convierte en una prioridad. Aunque el costo de hardware y software es cada vez menor, el costo de gestión y soporte es cada vez mayor. El 70% de los presupuestos de TI se asignan al mantenimiento de sistemas existentes dejando alrededor del 30% para nuevas soluciones. [Ferguson et al. 2007].

Una de las dificultades que existen en el desarrollo de aplicaciones informáticas es la falta de interoperabilidad entre sistemas ante las crecientes necesidades de integración. La alineación entre el negocio y las TI constituye uno de los beneficios más importantes que se le atribuyen a la Arquitectura Orientada a Servicios (*SOA* por sus siglas en inglés). *SOA* ha revolucionado principalmente la manera de adoptar la tecnología, ha surgido como paradigma capaz de soportar la agilidad de los procesos de negocio, aumentando la eficacia y eficiencia de las operaciones de las empresas en el mundo de hoy [Matsumara et al. 2009].

Según [Microsoft 2006]: La Arquitectura Orientada a Servicios es una filosofía de diseño que permite un mejor alineamiento de las TI con las necesidades de negocio, permitiendo a empleados, clientes y socios comerciales responder de forma más rápida y adaptarse adecuadamente a las presiones del mercado. SOA supone una estrategia general de organización de los elementos de TI, de forma que una colección de sistemas distribuidos y aplicaciones compuestas se pueda transformar en una red de recursos integrados, simplificada y sumamente flexible.

Entre los elementos claves asociados a SOA, que constituyen la base de sus beneficios, se incluye el desarrollo de servicios que son compartidos dentro de una infraestructura TI mediante el uso de estándares abiertos que permiten la interacción independiente de la tecnología subyacente. SOA es una evolución de las arquitecturas distribuidas y los métodos, principios de integración de aplicaciones más conocidos en el mundo computacional como EAI (Enterprise Application Integration). SOA enriquece la concepción y la puesta en práctica de los métodos que la antecedieron y se enfoca en la reusabilidad de sistemas legados, la composición de servicios, procesos de negocio y aplicaciones con el objetivo de lograr una interoperabilidad real [Blanvalet et al. 2006].

Paralelamente a las arquitecturas, las aplicaciones informáticas han experimentado una constante evolución. Las primeras aplicaciones eran sencillas, diseñadas para las necesidades más básicas de los usuarios que se ejecutaban comúnmente en una misma estación de trabajo. Posteriormente surgieron las aplicaciones distribuidas donde se separaban las diferentes capas de la aplicación.

La globalización, especialización y desarrollo de software ha requerido de un trabajo cada vez más colaborativo, por lo que se hace indispensable un cambio en cuanto a las aplicaciones que los clientes usan para tomar decisiones y emprender acciones. Estas aplicaciones deben sustituir

a las aplicaciones monolíticas y pobremente integradas entre sí. Al adoptar SOA se agregan beneficios referentes a la creación de servicios y aplicaciones que coexisten independientemente de la variedad tecnológica brindándole mayor reusabilidad al conjunto de funcionalidades existentes en la organización. El poder de combinación, composición y reutilización de los servicios estimula la construcción de aplicaciones compuestas que ofrecen constantemente soluciones flexibles a las necesidades de la empresa.

En el mundo de las telecomunicaciones existe una frase común para designar esa pequeña parte de tecnología que lleva los datos desde el último contacto de un enchufe hasta la caja de electricidad de la vivienda del cliente. Es la llamada "última milla" y generalmente es vista como uno de los mayores desafíos, pues este último paso en la cadena tecnológica, puede ser un considerable compromiso hacia el cliente. En la industria del software también existe una "última milla": se trata de poner la aplicación correcta en las manos del usuario final. Las aplicaciones compuestas son también encargadas de recorrer esta última milla, combinando generalmente una interfaz enriquecida con una tecnología de integración orientada a aplicaciones que usan como base arquitecturas SOA [Febles et al. 2012].

Las **Aplicaciones Compuestas** son colecciones de aplicaciones combinadas mediante el ensamblaje de componentes. Incluye las posibilidades de personalización y configuración para que los usuarios puedan modificar funcionalidades específicas en la aplicación. Tienen como objetivo principal brindar eficiencia a los procesos de negocio a través de las 3A: Agilidad, Adaptabilidad y Alineación [Banerjee 2006]. Aunque pueden construirse basadas en cualquier tipo de arquitectura, SOA es un entorno idóneo para construir este tipo de aplicaciones pues coinciden sus principios y los beneficios que se esperan obtener a partir de ellas.

La repercusión que tiene este nuevo enfoque de las aplicaciones compuestas es grande y está motivado en gran medida por las posibilidades que ofrece para hacer llegar los beneficios de SOA a los usuarios de forma ágil y flexible. Forrester Research [Dunlap 2006] ha sugerido que el 80% de los nuevos desarrollos de aplicaciones será en forma de aplicaciones compuestas, en lugar de ser aplicaciones tradicionales [Sopeña et al. 2008].

El desarrollo de aplicaciones compuestas es un proceso que presenta muchas de las bondades arquitectónicas que brinda SOA dirigidas a las capas más cercanas de los usuarios. El autor de esta investigación coincide con el planteamiento de que las aplicaciones compuestas son la última iteración de SOA que actualiza y potencia los sistemas, que mediante una adecuada gestión pueden brindar la posibilidad de crear nuevas soluciones a partir de elementos de software y fuentes de información existentes añadiendo valor a la empresa, aumentando la productividad de sus recursos humanos y dando un paso importante en busca de la agilidad empresarial al reutilizar componentes de software funcionales.

Las aplicaciones compuestas de los primeros años de la década del 2000 se basaban en una arquitectura relativamente simple. En el año 2005, los proveedores líderes de plataformas de middleware, como IBM, BEA Systems y Oracle, introdujeron motores especializados sobre sus plataformas de servidores de aplicaciones, los que brindaron una capacidad mucho mayor a las empresas que querían construir aplicaciones integrales. Ciertos servidores de portales, como IBM WebSphere Portal y BEA WebLogic Portal, se encargaron de las tareas relacionadas con las interfaces de usuarios aplicando las últimas tecnologías y marcos de trabajo referentes a la evolución de la Web. Más recientemente, los componentes pertenecientes al mundo SOA, como los buses de servicios empresariales (Enterprise Service Bus, ESB) y motores de lenguaje de ejecución de procesos de negocios (Business Process Execution Language, BPEL), posibilitaron

la prestación de servicios en línea y la rápida introducción de nuevas funcionalidades en una aplicación sin la necesidad de un cambio radical en su arquitectura [Bharadwaj 2010; ORACLE 2008; Panda and Maheshwari 2009]. Estos componentes ofrecieron un mayor dinamismo al desarrollo de aplicaciones compuestas [Laws and Combella 2011].

Las aplicaciones compuestas simbolizan una meta empresarial codiciada desde hace ya mucho tiempo con lo cual los usuarios pueden desarrollar capacidades de negocios en modo de componentes. Ofrecen la posibilidad de trasladar el debate de la reutilización desde el dominio técnico hacia el dominio de negocios, librando a las empresas de aplicaciones aisladas y permitiendo a sus desarrolladores definir comportamientos optimizados para sus negocios. Las tecnologías emergentes han cambiado la forma de ejecutar procesos de negocio y la composición se vuelve un aspecto cada vez más importante para la creación de lógica de negocios [Keyser 2007]. Aunque esta afirmación fue realizada hace 4 años, a criterio de este autor la misma mantiene su vigencia. Posiciones similares asumen otros autores las que se reflejan en este documento y son tenidas en cuenta como fundamentos teóricos de la presente investigación.

El soporte de software, la integración y el mantenimiento de los sistemas se eleva a un costo tal que limita a las organizaciones y reduce su capacidad de crear nuevas funcionalidades [Holley and Arsanjani 2010]. Las facilidades que brinda la composición estimulan la creación de aplicaciones mediante el empleo de funcionalidades existentes.

Si bien la construcción de una aplicación integral es una necesidad para las organizaciones en la actualidad, su desarrollo, despliegue y mantenimiento son tareas mucho más complejas [Duggan 2012]. La variedad de plataformas, herramientas y modelos arquitectónicos necesarios para la construcción de aplicaciones introducen una complejidad técnica que puede convertirse en algo abrumador, pues es necesario tener amplios conocimientos y habilidades para enfrentar este reto.

De no tomarse las decisiones correctas pudiera conducir a problemas en el rendimiento, el funcionamiento, la calidad de los servicios y la elevación de los costos de desarrollo y mantenimiento.

La Universidad de las Ciencias Informáticas (UCI) en sus primeros años enfrentó el reto de desarrollar sistemas informáticos complejos para empresas a nivel nacional e internacional, en las cuales no estaban ausentes los problemas referentes a la interoperabilidad, negocios altamente cambiantes y diversidad de tecnologías y proveedores de software. Cuba no cuenta con instituciones que tengan experiencia suficiente en este marco [León et al. 2010]. Por ello se crea el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) perteneciente a la Facultad 5 de la UCI que realiza investigaciones con el objetivo de definir las cuestiones necesarias para adoptar en las organizaciones cubanas el modelo arquitectónico basado en servicios y fomentar el desarrollo de aplicaciones compuestas.

Basado en el estudio realizado sobre las aplicaciones compuestas en iniciativas orientadas a servicios es posible destacar las siguientes dificultades y/o limitaciones:

- Las metodologías de desarrollo y marcos de trabajo de las empresas líderes vienen atadas a sus propias herramientas y tecnologías las cuales son vendidas a elevados precios.
- En las fuentes consultadas generalmente se trata la implantación de SOA sobre la base de componer aplicaciones entre servicios y por lo general no se refiere a cómo brindar los beneficios de la composición a desarrolladores y al usuario final, ni cómo apoyar la toma de decisiones [Duggan 2012; Juric and Krizevnik 2010; Mohebi 2012].
- La existencia de un número elevado de tecnologías y marcos de trabajo para el desarrollo de aplicaciones compuestas e interfaces enriquecidas dificulta la toma de decisiones en su proceso de selección.

- Existen repositorios para almacenar los recursos (servicios) que pueden ser reutilizados y que contienen información útil sobre los mismos, pero el proceso de selección muchas veces se hace complejo y ambiguo (la información no es clara y/o precisa).
- Las organizaciones por lo general no registran las mejores experiencias en el desarrollo de aplicaciones principalmente por la ausencia de herramientas que faciliten la gestión de conocimiento.
- Gran parte de estos problemas están asociados a que las organizaciones generalmente no poseen una infraestructura tecnológica adecuada para soportar la **interoperabilidad** entre aplicaciones, así como el intercambio de información de una manera **estandarizada** a través de servicios lo que dificulta el desarrollo de aplicaciones compuestas.

Toda esta problemática influye en que, con frecuencia, se hace complejo el desarrollo de aplicaciones compuestas y constituyen limitaciones que minimizan los beneficios esperados.

Lo referido anteriormente fundamenta la formulación del siguiente **problema científico**:

¿Cómo integrar herramientas, buenas prácticas y marcos de trabajo para el desarrollo de aplicaciones compuestas en iniciativas orientadas a servicios que contribuya favorablemente a la interoperabilidad y la estandarización del proceso?

Objeto de la investigación: Composición de aplicaciones basadas en Arquitectura Orientada a Servicios.

Como **objetivo general** de la investigación se plantea:

Elaborar un modelo basado en la integración de herramientas, buenas prácticas y marcos de trabajo para el desarrollo de aplicaciones compuestas en iniciativas orientadas a servicios que contribuya a la interoperabilidad y la estandarización del proceso.

Objetivos específicos:

1. Construir el marco teórico referencial de la investigación, relacionado con el desarrollo de aplicaciones compuestas y arquitecturas orientadas a servicios.
2. Desarrollar un modelo para la construcción de aplicaciones compuestas basado en SOA.
3. Seleccionar herramientas de software que integren la infraestructura que pueda sostener tecnológicamente al modelo.
4. Implementar una aplicación compuesta que apoye la toma de decisiones usando el paradigma de Razonamiento Basado en Casos.
5. Validar el modelo desarrollado.

Según lo expuesto, y derivada de la construcción del marco teórico referencial de esta investigación se plantea la siguiente **Hipótesis de investigación:**

La fundamentación y aplicación consecuente de un modelo basado en la integración de herramientas, buenas prácticas y marcos de trabajo para la construcción de aplicaciones compuestas contribuirá a la interoperabilidad y estandarización del proceso.

Entre los métodos de trabajo científico utilizados se destacan los siguientes:

Métodos generales: El método hipotético-deductivo para la elaboración de la hipótesis central de la investigación y para proponer nuevas líneas de trabajo a partir de los resultados parciales; el método sistémico para lograr que los elementos que forman parte del modelo sean un todo que funcione de manera armónica; el método histórico-lógico y el dialéctico para el estudio crítico de los trabajos anteriores, y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.

Métodos lógicos: El método analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta.

Métodos empíricos: El método de la entrevista para obtener los problemas presentes en las empresas estudiadas con relación a la composición de aplicaciones, criterios sobre indicadores de calidad, las competencias necesarias para el desarrollo de aplicaciones compuestas, entre otros aspectos. El método experimental para comprobar la utilidad de los resultados obtenidos a partir del modelo definido.

La Novedad Científica del trabajo se expresa en los siguientes aportes teóricos y prácticos:

Aporte teórico: Un modelo integrado de herramientas, buenas prácticas y marcos de trabajo para el desarrollo de aplicaciones compuestas, basadas en una arquitectura orientada a servicios.

Aportes prácticos:

- Una infraestructura que integra herramientas de software y soporta tecnológicamente al modelo.
- Una aplicación compuesta, de apoyo a la toma de decisiones, basada en el modelo desarrollado.
- Una base de casos que apoye a la toma de decisiones en la selección de herramientas y marcos de trabajo que permitan el desarrollo de aplicaciones compuestas.

Aporte docente:

Contribuye a enriquecer el sistema de conocimientos de la asignatura de Programación a través de nuevas maneras de desarrollar aplicaciones informáticas, uso novedoso de lenguajes de programación conocidos y su relación con estándares establecidos a nivel mundial. Esto puede

ser incorporado al programa de las asignaturas de programación en las actividades docente-investigativas de los estudiantes y en sus trabajos de diploma.

La tesis está estructurada en: introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y un cuerpo de anexos.

En el capítulo 1 se presenta los fundamentos teóricos de la investigación, estudio y actualidad de los distintos tipos de aplicaciones compuestas y su relación con SOA. La variedad de tecnologías y estándares presentes en la construcción de aplicaciones compuestas.

En el capítulo 2 se realiza un diagnóstico de la situación actual. Se define un modelo para el desarrollo de aplicaciones compuestas, se presenta la aplicación compuesta SI-Holmes de apoyo a la toma de decisiones y se fundamenta la incorporación de una base de casos en el funcionamiento del modelo.

En el capítulo 3 se validan los resultados obtenidos.

CAPÍTULO I

MARCO TEÓRICO REFERENCIAL DE LAS APLICACIONES COMPUESTAS EN ENTORNOS ORIENTADOS A SERVICIOS

CAPÍTULO I – MARCO TEÓRICO REFERENCIAL DE LAS APLICACIONES COMPUESTAS EN ENTORNOS ORIENTADOS A SERVICIOS.

En el capítulo se examina la evolución de los diferentes paradigmas relacionados con el desarrollo de aplicaciones informáticas. Se analizan modelos asociados con el paradigma orientado a servicios y su vinculación con el desarrollo de aplicaciones compuestas como una de sus bases fundamentales. Se examinan herramientas que forman parte de infraestructuras tecnológicas para el desarrollo y despliegue de aplicaciones compuestas y se fundamenta la opinión del autor siempre que sea necesario establecer delimitaciones conceptuales precisas para su aplicación en la presente investigación.

1.1 Evolución de las aplicaciones informáticas

A pesar de que los sistemas informáticos han logrado considerables mejoras en la productividad de las organizaciones, la arquitectura tradicional de las aplicaciones informáticas hace de los cambios y transformaciones, procesos costosos en cuanto a tiempo y recursos. Esto impide la existencia de una correspondencia entre las funcionalidades que ofrecen estos sistemas y las necesidades reales del negocio.

Los sistemas de información tradicionales tienden a proliferar como sistemas redundantes e inaccesibles entre sí (silos) que generan una gran cantidad de conexiones punto a punto (espaguetis) y provocan esquemas inconexos con marcadas ineficiencias en su funcionamiento [Davis 2009].

La adopción de una arquitectura adecuada que localice, conecte los servicios de las TI existentes y que las nuevas aplicaciones se aprovechen de esto sin necesidad de ser construidas desde cero, constituye una alternativa para eliminar las conexiones punto a punto [Febles and Santos 2010].

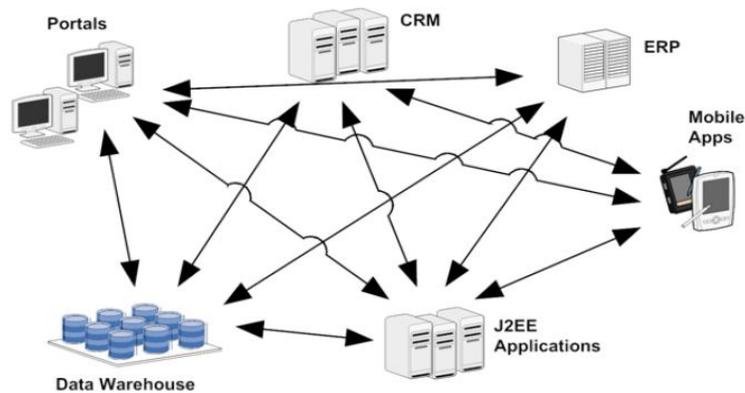


Figura 1.1 Ejemplo de conexiones punto a punto en los sistemas de información. Fuente : [Davis 2009]

El desarrollo de software se encuentra en una evolución constante que abarca lenguajes de programación, plataformas de desarrollo, tecnologías para el procesamiento de datos y transacciones, la web, Internet, entre otros elementos. Es creciente la demanda de software que responda a los rápidos cambios en los procesos de negocio y contribuya a ofrecer mayor valor a la información que llega a los clientes. En el centro de esta evolución se encuentra el desarrollo constante de las aplicaciones que se refleja en la Figura 1.2.

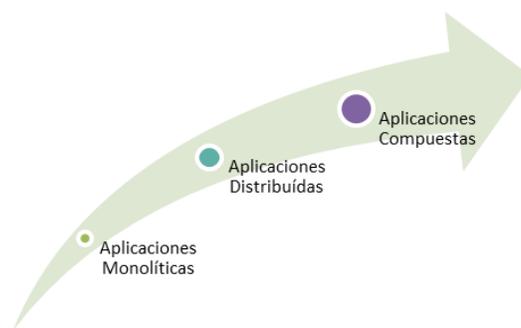


Figura 1.2 Tendencia en el desarrollo de aplicaciones Fuente: Elaboración propia.

El autor considera que la creciente necesidad de acoplar elementos de software para establecer un nivel confiable de intercambio de datos ha convertido a la integración en una meta estratégica dentro del desarrollo de aplicaciones.

1.2 La orientación a servicios

La orientación a servicios representa un estado de evolución en la historia del desarrollo de las aplicaciones informáticas que combina acertadamente elementos de diseño de paradigmas anteriores con nuevos elementos, estimulando la innovación tecnológica. En la Figura 1.3 se muestran las principales influencias de la orientación a servicios que fue concebida para brindar soluciones a los problemas heredados de repetidas generaciones de soluciones distribuidas tradicionales. Sus antecedentes se ven marcados por otros términos que pertenecen al dominio de las aplicaciones distribuidas [Erl 2005].



Figura 1.3 Influencias de la orientación a servicios Fuente: Elaboración propia

El paradigma de las aplicaciones distribuidas, que es un antecedente importante de la orientación a servicios, comenzó principalmente con DCE (Distributed Computing Environment), RPC (Remote Procedure Call) y los sistemas de mensajería como MQSeries. Más tarde surgieron elementos distribuidos como los ORBs (Object Request Brokers), CORBA (Common Object Request Broker Architecture), DCOM (Distributed Component Object Model), y RMI (Remote Method Invocation). Basados en los anteriores han sido desarrollados modelos como EJB (Enterprise Java Beans), COM+ (Component Object Model), .NET Enterprise Services, y CCM (CORBA Component Model) [Erl 2007].

La integración se convirtió en un punto de atención a finales de la década del 90. Numerosos sistemas fueron desarrollados con poca previsión de cómo se podría intercambiar datos fuera del contexto del propio sistema. Como resultado se crearon soluciones de integración que provocaron ineficiencias en cuanto a la interoperabilidad, la reusabilidad y el uso de estándares en el proceso de intercambio de información entre las aplicaciones [Erl 2007].

Las plataformas EAI (Enterprise Application Integration) introdujeron herramientas que permitieron una abstracción de las aplicaciones propietarias mediante el uso de adaptadores, mediadores y motores de orquestación. Estas plataformas de integración fueron más robustas y extensibles, pero también se convirtieron en soluciones extremadamente complejas y costosas que requerían compromiso a largo plazo con los propietarios de las plataformas de desarrollo.

La llegada de los marcos de trabajo libres para el desarrollo de servicios Web que se abstraían de la tecnología propietaria para su funcionamiento fue un nuevo horizonte en los proyectos de integración. Con ello disminuyeron las dependencias con los proveedores de las plataformas. Algunas innovaciones que alcanzaron popularidad en la era de las EAI fueron reconocidas e incorporadas a las metas globales asociadas a la orientación a servicios por la manera en que coincidían en principios como la abstracción, el bajo acoplamiento y la facilidad de composición [Al-Jaroodi et al. 2010].

Las iniciativas de integración implicaban la conexión de dos o más aplicaciones o programas que pudieran ser incompatibles, estar basados en tecnologías diferentes o no estar concebidos para la conexión fuera de sus fronteras. El incremento de las necesidades de acoplar elementos de software con el fin de establecer un nivel pertinente de intercambio de datos e información fue lo que convirtió la integración en un gran reto para la industria de las TI [Erl 2007].

Los procesos de integración han sido históricamente vistos como habilitadores de la interoperabilidad entre sistemas, sin embargo una de las metas fundamentales de la orientación a servicios es el establecimiento de una interoperabilidad nativa con el fin de reducir las necesidades de integración entre las aplicaciones informáticas existentes en las organizaciones.

El autor coincide con la definición de interoperabilidad propuesta por la IEEE como: “*la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada*” [IEEE 1990], por integrar los elementos esenciales de la interoperabilidad.

Es opinión del autor, formada no solo a través del estudio de los fundamentos teóricos de la literatura consultada, sino también basada en la experiencia personal y en el estudio de casos, que la **interoperabilidad** se puede lograr cuando se elaboran los servicios con una visión que se ajuste a los principios de la orientación a servicios, a los **estándares** y a las buenas prácticas de diseño.

1.2.1 Arquitectura Orientada a Servicios

La arquitectura orientada a servicios (*Service Oriented Architecture SOA*) representa una manera de pensar basada en servicios relacionados con el contexto de negocio y TI que busca reducir los costos de desarrollo y mantenimiento de las aplicaciones. Como promedio, las organizaciones con gran capacidad tecnológica gastan alrededor del 85% de su presupuesto destinado a los objetivos TI en mantener sus operaciones y sistemas existentes mientras que solo un 15% en innovaciones y la creación de funcionalidades para el negocio, desperdiciando oportunidades que le permitan la aplicación de nuevas tecnologías [Holley and Arsanjani 2010].

Según [Graham 2008] la idea principal de SOA es la de construir aplicaciones que soporten los procesos de negocio mediante la combinación de elementos de software poco acoplados entre sí,

llamados servicios los cuales son clasificados como: *“unidades de trabajo creadas por un proveedor para lograr un resultado deseado en los consumidores”*.

La arquitectura orientada a servicios estimula el bajo acoplamiento entre los servicios que componen una aplicación, al contrario de las aplicaciones monolíticas tradicionales que se caracterizan por una fuerte interdependencia de sus componentes.

El autor de la presente investigación, a partir del estudio realizado, considera que las razones principales para que las organizaciones decidan adoptar el paradigma orientado a servicios son: facilitar la integración de sistemas, aumentar la reusabilidad de las funcionalidades de las aplicaciones existentes, transformar y modernizar aplicaciones estratégicas, lograr respuestas ágiles a las necesidades de negocio y ganar en organización de los procesos de negocio.

La actualidad de la orientación a servicios se observa en su adopción por organizaciones de diversos sectores con resultados publicados como: [Aboulsamh 2009; Al-Jaroodi, Mohamed and Aziz 2010; Alahmari et al. 2010; Bertolino and Polini 2009; Coffey et al. 2010; Chen et al. 2008; Dahman et al. 2010; Delgado 2006; Dexter et al. 2012; Dias et al. 2010; Gronosky et al. 2010; Gu and Zhang 2010; Haines and Haseman 2009; Hirzalla 2010; Honghui and Xiaojun 2010; Jian et al. 2010; Kazem and Forte 2010; Kistasamy et al. 2010; Li 2010; Li and Xiong 2010; Obeidat and Zaatreh 2010; Papageorgiou et al. 2010; Postina et al. 2010; QH et al. 2010; Qi-wen and Qi 2010; Rajani et al. 2010; Sangroya et al. 2010; Silva-Lepe et al. 2010; Sousa and Neto 2012; Srivastava and Mazzoleni 2010; Tang et al. 2010; Voelz and Goeb 2010 ; Yan and Guo 2010; Yunliang et al. 2010; Zender et al. 2010; Zhang et al. 2011; Zhang and Zhang 2009; Zhou et al. 2010].

SOA estimula la creación de aplicaciones basadas en servicios, preparadas para el cambio, minimizando la rigidez y complejidad arquitectónica [Josuttis 2007]. También reduce los costos

de desarrollo y mantenimiento puesto que elimina la complejidad a la hora de modificar y probar las aplicaciones [Bieberstein et al. 2005]. Las aplicaciones responden al negocio porque componen servicios que pueden ser rápidamente modificados y re-desplegados en nuevos contextos de negocio, permitiendo una respuesta rápida a las necesidades de los clientes.

Los servicios representan activos del negocio que deben estar listos para ser consumidos, configurados, re-usados o ensamblados. En este entorno tienen su propio ciclo de vida, operan como una caja negra que puede ser accedida por diferentes fachadas (aplicaciones web, de escritorio, para móviles) y deben cumplir con los siguientes principios de diseño [Erl 2007]:

- No deben tener estado.
- Fáciles de descubrir.
- Bajamente acoplados.
- Gobernados por políticas.
- Independientes de su localización, lenguaje y protocolo.
- Fáciles de componer.

El autor se identifica con la definición de **servicio** dada en [Sopeña, Camino and Navarro 2008] :

“elementos de computación, que se pueden describir autónomos, independientes de la plataforma, que se pueden describir, publicar, descubrir, orquestar, y programar usando protocolos normalizados con el propósito de construir redes de aplicaciones colaborativas distribuidas dentro y a través de las fronteras de las organizaciones”, por el contenido técnico de la misma y su cercanía a los principios de la orientación a servicios.

Los servicios, como parte del esquema básico de la orientación a servicios, se instrumentan en función de tres roles: el consumidor solicitante del servicio y que coincide con el cliente, el proveedor, y el intermediario. Un cliente puede beneficiarse de diferentes servicios y ofertarlos

como uno nuevo, convirtiéndose en proveedor. El papel del intermediario es justamente ayudar a los clientes a encontrar el servicio más adecuado a sus necesidades. [Sopeña, Camino and Navarro 2008].

Autores como [Guelmes 2010] [Daigneau 2011; Duggan 2012; Rosen et al. 2008] conceden gran importancia a la etapa de diseño de servicios. En [Erl 2005] se define el diseño de servicios de la siguiente manera:

“El diseño orientado a servicios es el proceso mediante el cual diseños de servicios físicos concretos son derivados de los servicios lógicos candidatos y luego ensamblados en composiciones abstractas que implementan un proceso de negocio.”

Para el desarrollo de aplicaciones compuestas basadas en servicios, el autor considera que los mismos deben cumplir con un grupo de principios y buenas prácticas relacionadas con la orientación a servicios. Entre estos principios se destacan los siguientes:

- **Los servicios deben ser reutilizables:** la lógica encapsulada por el servicio debe ser suficientemente genérica como para ser empleada en numerosos escenarios y por varios consumidores.
- **Los servicios deben proporcionar un contrato estandarizado:** deben expresar su propósito y capacidades a través de un contrato de servicio o interfaces técnicas mediante las cuales puedan ser accedidos y estas últimas deben responder a un estándar de definición de contrato.
- **Los servicios deben tener bajo acoplamiento:** un contrato de servicio debe ser independiente de la implementación del servicio. Los servicios deben ser independientes unos de otros.

- **Los servicios deben tener alta cohesión:** las funcionalidades provistas por el servicio deberán estar fuertemente relacionadas.
- **Los servicios deben ser abstractos:** es necesario ocultar los detalles fundamentales sobre la tecnología y la lógica interna a los consumidores del servicio para que sea independiente de la tecnología subyacente.
- **Los servicios deben permitir la composición:** todo servicio debe ser construido de tal manera que pueda ser utilizado para construir otros servicios de mayor abstracción.
- **Los servicios deben de ser autónomos:** los servicios deben ejercer tanto control como puedan sobre su entorno de ejecución y recursos. De esta manera ganan en independencia y se potencia su reutilización.
- **Los servicios no deben tener estado:** un servicio no debe guardar ningún tipo de información relacionada con su estado y siempre debe estar listo para ser consumido en cualquier composición.
- **Los servicios deben poder ser descubiertos:** todo servicio debe poder ser descubierto para que pueda ser utilizado, esto contribuye a evitar la creación accidental de servicios que proporcionen las mismas funcionalidades. Los contratos de servicios deben contener metadatos mediante los cuales el servicio pueda ser descubierto y evaluado.

El autor de esta investigación, no solo coincide con estos principios sino que ha comprobado su validez durante su experiencia técnica y científica y los ha tenido en cuenta durante toda la investigación desarrollada.

Los servicios pueden ser desarrollados como funcionalidades nuevas, también pueden exponer funcionalidades de aplicaciones legadas con el objetivo de facilitar la integración entre las nuevas soluciones y las existentes. Los consumidores de servicios deben acceder a estos

mediante interfaces que definen un grupo de operaciones por servicio. Estas interfaces son contratos que deben estar separados de la implementación de los servicios, ser auto-descriptivos e independientes de la plataforma tecnológica para lograr un correcto intercambio de mensajes.

El autor de esta investigación considera que las dificultades que se presentan en la práctica para la selección de las tecnologías y la aplicación de los principios de la orientación a servicios, están dadas por la falta de conocimiento, el poco aprovechamiento de las experiencias acumuladas y por no disponer de herramientas adecuadas para gestionarlas.

Cuando en las entidades el desarrollo y consumo de servicios toma auge, existe el peligro de la desorganización en su gestión lo cual influye en un mal uso de las tecnologías y los principios de la orientación a servicios. El autor de esta investigación considera que resulta muy importante, para minimizar estos peligros, utilizar las estructuras correctas para la gestión de la meta información y el ciclo de vida de los servicios, en este caso, los Registros/Repositorios.

1.2.2 Registros/ Repositorios de servicios.

La reusabilidad de los servicios en diferentes contextos es una meta fundamental de SOA. Su práctica elimina la existencia de servicios con funcionalidades similares o repetidas. Los Registros/Repositorios son las estructuras que almacenan servicios y al mismo tiempo actúan como catálogos que simplifican y automatizan la búsqueda de servicios [Erl et al. 2010].

Estas herramientas suelen disponer de numerosas funcionalidades entre las que se encuentran:

- Gestión de ciclo de vida de los servicios.
- Monitoreo del funcionamiento de los servicios [Dirksen 2012].
- Facilidades de alerta de cambios a los suscriptores y desarrolladores de los servicios.
- Gestión de la información de los contratos de servicio y de las políticas.

Los proveedores publican servicios en los Registros/Repositorios para que los consumidores puedan descubrirlos, conocerlos por nombre, por funcionalidad o por propiedades específicas, estimulando la reusabilidad y jugando un papel fundamental en la gobernabilidad de un entorno SOA.

La forma más común de exponer los servicios se basa en el estándar WSDL (Web Services Description Language), documento XML que funciona como contrato de servicio. El uso de un repositorio permite disponer de los servicios organizados y fácilmente accesibles para ser reutilizados. Frecuentemente los registros usan el estándar UDDI (Universal Description Discovery and Integration) para facilitar su publicación y descubrimiento.

El autor de la investigación considera que el uso incorrecto de estándares como estos ha afectado los resultados del desarrollo de aplicaciones compuestas basadas en servicios debido a:

- El diseño de contratos de servicios sin tener en cuenta las buenas prácticas ni los principios de la orientación a servicios disminuye la interoperabilidad.
- La ausencia de Registros/Repositorios que soporten estándares para el descubrimiento de servicios afecta la relación entre proveedores y consumidores de servicios.

En los inicios de la era de la orientación a servicios, los directorios UDDI bastaban para almacenar y administrar los servicios existentes en las organizaciones. Pero el nivel de complejidad de los servicios y su crecimiento exponencial obligaron la evolución de las estructuras que se encargaban de estas tareas.

Actualmente los Registros/Repositorios (R/R) gestionan un conjunto de políticas que se relacionan a los servicios durante su diseño y su ejecución. Además gestionan el ciclo de vida y los contratos de servicios, que son importantes para administrar la relación entre proveedores y consumidores, a través de elementos como los SLA (Service Level Agreement). En la

bibliografía especializada se sostiene que el R/R debe incluir una descripción completa de los servicios, de los datos y parámetros necesarios para acceder a ellos, de modo que puedan ser fácilmente reutilizados.

El descubrimiento de servicios requiere no solo de su localización exitosa, sino también de la existencia de meta información que permita un buen entendimiento de los recursos, para conocer si la funcionalidad que se busca existe o se necesita desarrollar.

Si no es posible descubrir un servicio apropiado, es necesario decidir entre usar uno menos efectivo de los disponibles o desarrollar uno que pueda satisfacer las necesidades iniciales, lo que se convierte en una decisión clave. La calidad de la meta información definida y la manera en que se encuentra disponible tienen gran influencia en esta decisión.

Coincidiendo con [Erl 2007], el autor considera que aunque la reusabilidad no sea un requisito absoluto, es importante reconocer que la orientación a servicios hace un énfasis sin precedentes en el reuso de componentes, notándolo en su sinergia con el desarrollo de aplicaciones compuestas. El establecimiento de Registros/Repositorios de servicios reusables y autónomos facilita la combinación de los mismos como componentes de la lógica de nuevas aplicaciones. Estas aplicaciones no son construidas desde cero, sino que son el producto de la composición del material existente implementado.

Una vez que un inventario de servicios ha sido establecido, la composición de servicios se convierte en la forma más viable de reutilización de funcionalidades y aplicaciones. Para ello resulta decisivo facilitar el descubrimiento de los servicios que se coloquen a disposición de los consumidores. Cada uno de estos servicios debe tener asociada determinada QoS (QoS Quality of Service) [Voelz and Goeb 2010] que es la combinación de elementos de rendimiento y elementos de seguridad relacionados con cada servicio.

Las propiedades de las aplicaciones basadas en servicios dependen directamente de las características de estos servicios, por lo que una selección apropiada dentro de un Registro/Repositorio es de gran importancia. Se han elaborado múltiples métodos para el proceso de selección de servicios que se enfocan en los atributos no funcionales como principal factor de influencia en dicha selección.

El método holístico para la selección de servicios Web en el diseño de aplicaciones compuestas de [Bonders and Grabis 2011] formula la selección de servicios como un problema de optimización. Estos autores sostienen que la importancia del proceso de selección para la gestión de aplicaciones compuestas ha sido un tema poco valorado en la literatura hasta la actualidad.

Existen otras investigaciones y modelos como [Cabrera et al. 2010], [Tomczak et al. 2012], [Mohebi 2012], que se enfocan en la selección de servicios utilizando métodos que toman la QoS como medida para evaluar a los servicios. Estas mediciones incluyen seguridad, confiabilidad, costo de ejecución, entre otras características no funcionales de los servicios.

El autor coincide con [Angus and Lan 2009] y su concepto expandido de QoS para la selección de servicios web que define como “el grado en que el sistema, componente o proceso conozca las necesidades o expectativas de los usuarios”. En dicha investigación se considera que la evaluación de las características no funcionales de los servicios es decisiva en el proceso de desarrollo de aplicaciones compuestas. Desafortunadamente la evaluación de las características funcionales es un proceso mucho más complejo que requiere del juicio, la estimación del grado de satisfacción y la interpretación humana, lo cual siempre incorpora cierto grado de subjetividad.

1.2.3 Composición de Servicios

La orientación a servicios contiene una visión muy ambiciosa y atractiva por lo que muchas organizaciones están interesadas en adoptar sus principios para perfeccionar el funcionamiento de sus sistemas de información. Una de las metas más codiciadas por las organizaciones es el incremento de la interoperabilidad entre sus sistemas basada en los estándares establecidos a nivel mundial.

Las aplicaciones tradicionales han sido diseñadas, en su mayoría, para una larga duración, con un alto acoplamiento que dificulta los cambios y actualizaciones. Las aplicaciones desarrolladas bajo una arquitectura bien definida, teniendo en cuenta la posibilidad de cambios futuros, ofrecen facilidades para su modificación, aunque estas modificaciones en su mayoría equivalen a dificultades, consumo de tiempo y usualmente a errores inesperados.

La composición fue clave en la evolución de TI pues se convirtió en elemento común en las arquitecturas de sistemas genéricos y en el diseño de software. La composición es un cambio de paradigma del desarrollo de aplicaciones monolíticas hacia aplicaciones orientadas al usuario que respaldan procesos de negocio específicos.

La composición de servicios permite la modificación de procesos de negocio rápidamente de una manera flexible. Para la composición puede ser usado un lenguaje creado específicamente para ese fin o emplear los lenguajes de programación conocidos (Java, C#, entre otros). Esta última opción puede ofrecer resultados poco flexibles pues no ofrecen una clara separación entre el flujo del proceso y la lógica de negocio.

La composición de servicios es típicamente relacionada a la automatización de procesos de negocio o de tareas. Muchas de las configuraciones para la composición pueden ser consideradas como el equivalente a aplicaciones tradicionales, en este caso, aplicaciones compuestas

[Blanvalet, Bolie, Cardella, Carey, Chandran, Coene, Geminiuc, Matjaž B. Jurič, Nguyen, Poduval, Pravin, Thomas and Todd 2006].

Existen varios beneficios que se derivan de la composición de servicios. El primero de ellos es la combinación de datos o funcionalidades existentes en las organizaciones, reusándolas para construir nuevas aplicaciones o nuevas funcionalidades [Thuraisingham 2011]. Su desarrollo es comúnmente relacionado con la orientación a servicios porque esta mitiga los errores y fallos de las generaciones de soluciones distribuidas tradicionales. Representa un estado evolutivo que combina elementos de diseño bien logrados anteriormente con elementos novedosos que impulsan una innovación desde el punto de vista conceptual y tecnológico. Entre sus principios fundamentales de diseño se encuentran:

- Aumento de oportunidades para el uso de la lógica de solución para múltiples propósitos.
- Aumento de la disponibilidad, escalabilidad, de un comportamiento pronosticable o predecible.
- Reducción de las dependencias entre las unidades lógicas de la solución.
- Aumento de la consistencia en la representación de los datos y funcionalidades.
- Incremento de la percepción y/o conocimiento de las soluciones disponibles.
- Reducción del conocimiento de los detalles de las soluciones subyacentes y de los detalles de su implementación.
- Incremento de posibilidades de combinación de las unidades de lógica de solución en diferentes escenarios.

1.3 Aplicaciones Compuestas.

Una de las piedras angulares de SOA es su capacidad para componer servicios, al convertirse la composición en un proceso cada vez más común, el concepto tradicional de “**aplicación**“, de sistema o solución comienza a disiparse. Las aplicaciones en la orientación a servicios no consisten en componentes auto controlados de código que son responsables de automatizar un grupo específico de tareas, sino que equivalen a una composición de servicios. Estos servicios probablemente participan indistintamente en otras composiciones que respondan a las necesidades de otras aplicaciones bajo configuraciones diferentes. Las aplicaciones en este tipo de ambiente pierden su individualidad, una gran parte de su lógica no le pertenece exclusivamente y son más conocidas como **aplicaciones compuestas** [Erl 2007].

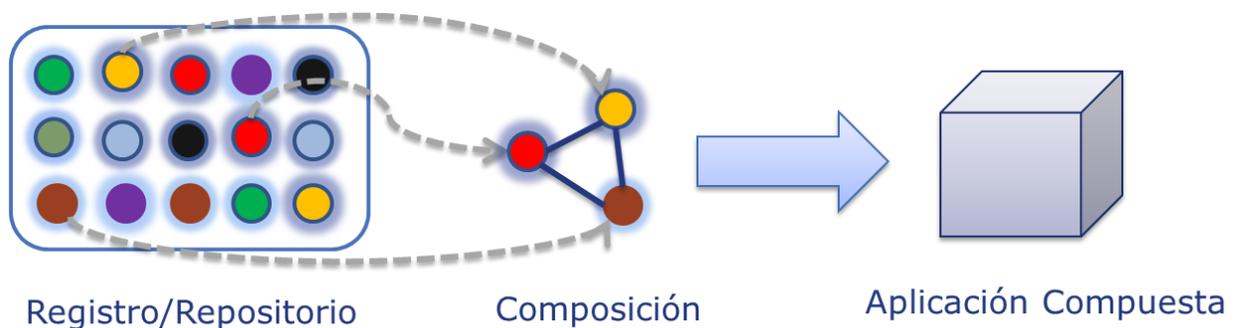


Figura 1.4 Aplicaciones compuestas por servicios Fuente: Elaboración propia

Las estructuras de aplicaciones compuestas tienen la capacidad de cambiar la forma en la que los usuarios finales construyen, entregan y experimentan las aplicaciones. Los más comunes escenarios de integración de sistemas usando aplicaciones compuestas son:

- Extensión de funcionalidades de aplicaciones empaquetadas.
- Enlace de nuevas soluciones de negocio a los módulos existentes.
- Agregar nuevos módulos operacionales a las aplicaciones existentes.

Con el establecimiento de un repositorio de servicios con un alto porcentaje de recursos reusables, se acentúa la centralización de la lógica de las soluciones que existían anteriormente en los silos de aplicaciones. Como resultado se satisfacen más necesidades de negocio, no con el desarrollo de nuevas funcionalidades si no con la composición de servicios existentes en nuevas aplicaciones.

1.3.1 Diseño y estructura de las aplicaciones compuestas

Para diseñar una aplicación compuesta se debe:

- Seleccionar uno o más contenedores desde cada capa y un conjunto de componentes que se puedan implementar dentro de estos contenedores.
- Elegir componentes. Definir el repositorio, los recursos que se deben crear a partir de las necesidades del negocio.
- Definir la aplicación compuesta. Establecer la forma en la que se ensamblarán los servicios para brindar un proceso funcional. La plataforma debe permitir que estas conexiones sean de acoplamiento bajo.

Estas aplicaciones atraviesan en su desarrollo distintas fases de ciclo de vida que representan los diferentes estados en la composición de servicios.

- Análisis y Diseño: Fase en la cual los miembros que formarán parte de la composición son seleccionados y es diseñada la configuración para la composición de servicios.
- Implementación: Fase donde la composición es operacional y activa.
- Gobierno: Fase donde fundamentalmente se considera la evolución a largo plazo de la composición de servicios, incluyendo la recomposición de sus miembros de acuerdo a las necesidades.

Entre las capas más comunes de las aplicaciones compuestas basadas en SOA se pueden destacar:

- **Capa de interfaz de usuario y lógica de cliente:** Las vistas se pueden ensamblar mediante partes de interfaz de usuario de estructura flexible que cooperan a través de una estructura subyacente. Esta composición se puede llevar a cabo a través de portales empresariales y sus tecnologías de apoyo para construir estructuras compuestas.
- **Capa de composición de servicios:** Se pueden desarrollar varios servicios junto con reglas, metadatos y flujo del proceso para proporcionar servicios de alto valor añadido.

La composición de servicio, por lo general, se realiza mediante motores de integración.

Para que la composición de aplicaciones resulte eficaz es necesario proveer un ambiente adecuado donde se creen servicios reutilizables e interoperables que pueden ser combinados rápidamente. Para una correcta composición es conveniente disponer de una asesoría respecto a la tecnología a utilizar (herramientas, lenguajes, marcos de trabajo, especificaciones, estándares, buenas prácticas), así como las funcionalidades que deben ser cubiertas y las habilidades que deben tener los desarrolladores de aplicaciones compuestas.

Al igual que con el cambio hacia la orientación al servicio, el desarrollo de aplicaciones compuestas requiere no solo reestructurar el código existente, sino también reconsiderar nuevo código junto a la definición de una infraestructura tecnológica que permita gestionar los servicios a componer [Keyser 2007].

1.4 Infraestructura tecnológica para desarrollar y desplegar aplicaciones compuestas basadas en SOA.

La infraestructura tecnológica está constituida por una serie de componentes y herramientas que se consideran necesarios para el funcionamiento de una organización o desarrollo de una

actividad [Krafzig et al. 2005]. Existen un conjunto de tecnologías que forman parte de la infraestructura para iniciativas orientadas a servicios que pueden ser utilizadas para el desarrollo de aplicaciones compuestas y que trabajan de forma independiente o integrada.

En el mercado mundial, muchas empresas privadas ofrecen herramientas para sustentar un escenario tecnológico que permita obtener los beneficios de la orientación a servicios y de la composición de aplicaciones, pero el precio de las mismas es muy elevado, lo cual representa un inconveniente para Cuba porque además, afecta el objetivo de lograr una soberanía tecnológica. Se hace necesario realizar un compromiso entre las características de las herramientas a utilizar y las posibilidades reales de su aplicación en el entorno real cubano.

En [Davis 2009] se aborda la selección de herramientas de código abierto válidas para formar parte de una infraestructura tecnológica para proyectos SOA. El autor de esta investigación coincide con los criterios expuestos en dicho trabajo aunque cree que la infraestructura para proyectos SOA no tiene que ser necesariamente igual a la utilizada para desarrollar y desplegar aplicaciones compuestas. De hecho en este trabajo de [Davis 2009] no se seleccionan herramientas para los servicios de presentación (sólo se mencionan como elemento necesario) lo cual el autor considera imprescindible para la visualización del resultado de la composición de aplicaciones.

Esta investigación contiene una propuesta que articula el desarrollo tecnológico con las posibilidades reales del país. El autor se basa en una infraestructura de herramientas de código abierto propuesta en el trabajo [Febles and Santos 2010] la cual se ilustra en la Figura 1.5.

Según Ray Ozzie [Ferguson, Pilarinos and Shewchuk 2007], Arquitecto Jefe de Software de Microsoft, para lograr el éxito en el desarrollo de aplicaciones es muy importante que se alcance un conocimiento amplio sobre las tecnologías.



Figura 1.5 Componentes de una infraestructura tecnológica para SOA basado en trabajos de Davis [Davis 2009]. Fuente: Elaboración propia

Si existen fallas en el proceso de construcción, selección de herramientas de acuerdo a los escenarios y no es posible evaluar rápidamente la gravedad de un problema determinado, el valor de la aplicación se ve superado en poco tiempo por el costo de las interrupciones, lo que pudiera conducir a problemas en el mantenimiento asociado con el soporte técnico de la aplicación.

1.4.1 Protocolos, especificaciones y estándares.

La creación de estándares es una tarea compleja. Diferentes grupos e instituciones trabajan en el desarrollo de especificaciones y de estándares en los distintos niveles requeridos, para lograr que el proceso de composición de aplicaciones en entornos orientados a servicios sea homogéneo. Autores, como los que a continuación se citan resultan imprescindibles para el estudio de este tema. [Angus and Lan 2009; Cabrera, Oriol, López, Franch, Marco, Frago and Santaolaya 2010; Davis 2009; Juric and Krizevnik 2010; Kanneganti and Chodavarapu 2007; Mahemoff 2006; Matsumara, Brauel and Shah 2009; Ngu et al. 2010; OASIS 2008; Ogrinz 2009; Olaf Zimmermann 2004; Popadiyn 2009; Rosen 2008; Salter and Jennings 2008; Sarang 2009; Sarin 2010; Sopeña, Camino and Navarro 2008].

1.4.1.1 Servicios Web.

Aunque SOA es independiente de la tecnología, la plataforma tecnológica de servicios Web es una de las máximas responsables de la popularidad de SOA. Los servicios Web tienen una influencia significativa en la orientación a servicios la cual ha reposicionado esta tecnología y la ha formalizado como suya propia [Erl 2005].

Los servicios Web son una de las tecnologías más usadas para alcanzar la interoperabilidad entre aplicaciones informáticas. Se han convertido en estándar para la integración de empresas a nivel de servicio. Pueden ser usados en diferentes plataformas, diferentes sistemas operativos y mediante varios lenguajes de programación [Juric 2010].

La tecnología de servicios Web es similar a la de sus predecesores pero al mismo tiempo difiere en algunos aspectos. Por ejemplo, es la tecnología distribuida que soporta la mayor cantidad de productores de software en el mundo pues cumple con la promesa de la interoperabilidad entre sistemas [Vohra 2012]. Esta tecnología está basada fundamentalmente en especificaciones y estándares como:

- **SOAP** (Simple Access Protocol): provee un formato de mensaje y mecanismo de mensaje que soporta el intercambio estandarizado de XML entre aplicaciones informáticas. Es un utilizado para expresar la estructura de datos basada en XML muy relacionado a la creación de los contratos de servicio [Sattari and Kunjumohamed 2012].
- **WSDL** (Web Services Description Lenguaje): es el estándar definido por el W3C (World Wide Web Consortium) para describir un servicio Web y crear el contrato entre el cliente y el proveedor [Balani and Hathi 2009]. No es un documento obligatorio pero es importante su estandarización para poder acceder de manera dinámica a los servicios. Fue creado para definir la interfaz de los servicios [Tong 2008].

- **XSD** (XML Schema Definition Language): esquema que provee un lenguaje formal para la definición de la estructura y validación de documentos XML. Sirve para describir la estructura detallada de elementos que forman parte del contrato del servicio [Tong 2010].
- **UDDI** (Universal Description Discovery and Integration): estándar de OASIS (Organization for the Advancement of Structured Information Standards) [OASIS 2006], consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de los estándares de comercio electrónico y servicios web. Permite mantener repositorios de especificaciones WSDL simplificando el descubrimiento del servicio web y el acceso a sus especificaciones. Posibilita que una aplicación busque dinámicamente servicios que ofrezcan una serie de características, seleccione el más adecuado y localice servicios alternativos si uno falla [Erl et al. 2011].

Existe un grupo de marcos de trabajo para desarrollar servicios Web que estimulan el correcto uso de estándares y de buenas prácticas [Daigneau 2011]. Entre ellos se encuentran: Axis2 [Tong 2008], Spring [Sattari and Kunjumohamed 2012] [Fisher et al. 2012], Apache CFX [Tong 2010], JAX-WS [Vohra 2012]. Aunque persiguen una misma finalidad, estos marcos de trabajo poseen características que lo diferencian por lo que su selección tiene un alto impacto para el desarrollo de servicios que serán compuestos posteriormente.

1.4.1.2 Servicios RESTful

Los servicios RESTful constituyen una técnica de arquitectura de software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web, escrita por Roy Fielding [Fielding 2000]. Aunque el término REST (Representational State Transfer) se refería originalmente a un conjunto de principios de arquitectura, en la actualidad se usa en el sentido más amplio para describir cualquier interfaz

web simple que utiliza XML y HTTP [Webber et al. 2010]. REST afirma que la web ha disfrutado de escalabilidad como resultado de una serie de diseños claves:

- Un protocolo cliente/servidor sin estado: cada mensaje HTTP contiene toda la información necesaria para comprender la petición [Baun et al. 2011].
- Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE [Burke 2010].
- Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso puede ser direccionado a través de su URI.

REST representó una evolución en cuanto a la manera de intercambiar información entre aplicaciones. Actualmente un grupo importante de organizaciones ofrece funcionalidades a través de este protocolo.

1.4.1.3 WS-BPEL

La tecnología más usada para expresar la lógica de la composición de servicios está basada en WS-BPEL (Web Service Business Process Execution Lenguaje) [Juric 2006]. La incorporación de la orquestación en los procesos de gestión de aplicaciones compuestas basadas en SOA abstrae la lógica de los procesos de negocio y asume la responsabilidad de ejecución de complejas composiciones de servicios [Juric and Krizevnik 2010].

WS-BPEL es el estándar más aceptado para definir procesos de negocio en la composición de servicios que fue concebido y es soportado actualmente por empresas líderes en el mercado de software como Oracle, IBM, Microsoft, SAP, entre otros [Bharadwaj 2010].

WS-BPEL es un lenguaje de alto nivel que lleva el concepto de servicio un paso adelante al proporcionar métodos de definición y soporte para flujos de trabajo y procesos de negocio. El enfoque sobre procesos de negocios modernos guió a BPEL a adoptar los servicios Web como su mecanismo de comunicación externa. Así las facilidades de mensajería BPEL se basan en el uso del WSDL para describir los mensajes entrantes y salientes [Bolie et al. 2006]. Entre sus principales objetivos se encuentran:

- Definir procesos de negocio que interactúan con entidades externas mediante operaciones de un servicio Web definidas usando WSDL y que se manifiestan a sí mismas como servicios Web. Definir procesos de negocio utilizando un lenguaje basado en XML.
- Proveer sistemas de control jerárquicos y de estilo gráfico, que permitan que su uso sea lo más fusionado posible. Proveer funciones de manipulación simple de datos, requeridas para definir datos de procesos y flujos de control.
- Soportar un método de identificación de instancias de procesos que permita la definición de identificadores de instancias a nivel de mensajes de aplicaciones. Los identificadores de instancias deben ser definidos por socios y pueden cambiar.
- Definir un modelo de transacción de largo plazo que se base en técnicas probadas tales como acciones de compensación y ámbito, de tal manera a brindar recuperación a fallos para partes de procesos de negocios de largo plazo [Laws and Combella 2011].
- Usar servicios Web como modelo para la descomposición y ensamblaje de procesos. Construir sobre estándares de servicios Web (aprobados y propuestos) tanto como sea posible, de manera modular y extensible [Rotem-Gal-Oz 2012].

El lenguaje WS-BPEL ha posibilitado modelar y ejecutar interacciones, composiciones, coordinaciones entre servicios que respondan directamente a las necesidades del proceso de negocios [Margolis and Sharpe 2007].

1.4.2 Mashups

La Web ha expandido su presencia de forma extraordinaria durante los últimos años en todos los ámbitos del saber humano, debido a la importancia de su entorno para prestar servicios al usuario y a la necesidad de crear aplicaciones sencillas de utilizar.

Un desarrollo de aplicaciones rápido y fácil ha sido siempre un objetivo de la ingeniería de software. El principio de la reusabilidad de algo que ya ha sido construido y probado es una de las maneras de conseguir este objetivo. Desde la programación estructurada, a la programación orientada a objetos, las plantillas, los servicios web, cada avance en programación ha nacido desde un deseo continuo de reusar material antes que construirlo desde cero [Ogrinz 2009].

La reusabilidad de la gran cantidad de información que se publica en la web actualmente fue un problema avizorado por Tim Berners-Lee quien ofrece una solución en lo que ha llamado “La Web Semántica” la cual describe una plataforma para el intercambio de datos, conocimiento y significado de la información [Segaran et al. 2009]. Mientras se definen nuevos elementos para llegar a esta nueva y prometedora fase en la evolución de la web, los que se encargan de cumplir parte de esta visión hoy en día son las aplicaciones híbridas o mashups.

Según [Yee 2008], los mashups son: “aplicaciones compuestas que combinan datos o funcionalidades de dos o más fuentes externas para crear un nuevo servicio, o un componente web que combina contenido beneficiando las experiencias de los usuarios. Implica fácil y rápida integración frecuentemente usando APIs (Application Programming Interface) y fuentes de datos

como Web feeds (contenido sindicado) y Web scraping (técnica para obtener información de la Web) para producir resultados superiores a los originales.

A los efectos de esta investigación el autor asume como definición de mashups la de: “una aplicación híbrida que combina contenido de varias aplicaciones para crear algo nuevo, una mezcla interesante de contenidos como pudieran ser galerías de los videos más populares, las noticias más relevantes, la localización de un artículo o personas en un mapa”, esto se debe a que facilitan la combinación de información de una manera sencilla, visualmente atractiva y eficaz [Febles et al. 2012].

Algunos autores afirman que “mashup” es una palabra que proviene de un término en inglés asociado a la música, que significa la creación de una nueva canción a partir de la mezcla o pedazos de otras canciones [Bernal 2009; Lynn et al. 2010; Yee 2008; Young 2009].

Los mashups representan un paso evolutivo en cuanto a la reutilización al combinar los distintos servicios y aplicaciones para crear un nuevo contenido. Es opinión del autor que las organizaciones deben prepararse para adoptar el desarrollo de mashups como un método válido y actual de composición de aplicaciones y servicios.

Los mashups tienen una alta presencia en la composición de aplicaciones y recursos de la Web tanto del lado del servidor como del lado del cliente. En la Figura 1.6 se presenta un escenario global donde actúan los mashups.

Según [Ogrinz 2009], atendiendo a sus características, los mashups se pueden agrupar en consumidores, de datos y empresariales. Los mashups consumidores son los más conocidos y los más fácil de ejemplificar, son aquellos que combinan datos de distintas fuentes y presentan una interfaz gráfica por lo general sencilla.

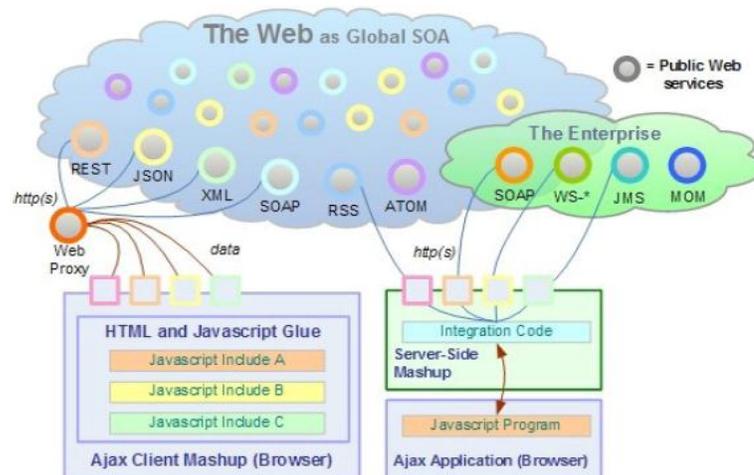


Figura 1.6 Escenario global de Mashups Fuente: <http://web2.wsj2.com>

Los mashups de datos mezclan información similar de distintas fuentes, presentando el resultado con un nuevo entorno gráfico. Los mashups empresariales, por lo general, combinan información tanto de fuentes externas como internas, aportando valor agregado a los datos de la propia institución. Pueden ser enfocados incluso a un proceso o grupo de procesos determinado, agregando una funcionalidad colaborativa y obteniendo como resultado una aplicación de negocio apropiada.

SOA proporciona una atmósfera favorable para que los desarrolladores creen nuevas aplicaciones basadas en el reúso y la combinación de funcionalidades existentes [Crupi and Warner 2008]. El uso de estándares abiertos combinado con los protocolos relacionados con la Web, ha llevado a las organizaciones que adoptan iniciativas de este tipo a obtener grandes beneficios.

El autor de esta investigación considera que las claves del éxito de los mashups en las organizaciones son: su facilidad para combinar información de múltiples fuentes y su filosofía de visualización de los resultados de dicha combinación a través de interfaces enriquecidas. Esta capacidad de visualización, que algunos autores llaman los servicios de presentación, es de vital

importancia y no está suficientemente tratado en la literatura relacionada con las aplicaciones compuestas.

Para la creación de mashups es necesario contar con herramientas que permitan desarrollarlos de forma sencilla [Young 2009]. La presente investigación estuvo dirigida, entre otros aspectos, a lograr la integración de los beneficios de la filosofía de los mashups en el desarrollo de aplicaciones compuestas, lo que enriquece este proceso.

1.4.3 Servicios de presentación.

Los servicios de presentación, representan la parte visible del proceso de negocio para el usuario final, por lo que deben permitir a los usuarios encontrar y completar elementos de trabajo de forma rápida, sencilla y eficiente. Los servicios de presentación para las aplicaciones compuestas desarrolladas en entornos orientados a servicios, pueden ser portales, sitios web, entornos de escritorio o aplicaciones para móviles [Davis 2009].

1.4.3.1 Aplicaciones Enriquecidas

Las primeras aplicaciones de Internet solamente soportaban interfaces básicas basadas en lenguaje HTML. A pesar de que funcionaban aceptablemente no tenían la apariencia de las interfaces de las aplicaciones de escritorio de la época, ni tampoco su funcionalidad puesto que dependían de la rapidez de la conexión a Internet.

El gran avance en la evolución de las aplicaciones web estuvo marcado por la aparición de Ajax. El Ajax (Asynchronous JavaScript and XML) es el nombre de una tecnología, descrita por James Garrett en febrero del 2005 [Garrett 2005], que permite la actualización parcial de las páginas Web sin la necesidad de ser recargadas completamente. Ajax separa la comunicación del lado del cliente con la del lado del servidor haciéndola funcionar de manera paralela, reduciendo así las demoras que experimentaban los usuarios hasta ese momento [Governor et al. 2009]. Con ello se

logró aumentar la calidad de la interacción de los usuarios finales con las interfaces web al incorporar:

- Presentación basada en estándares con el uso de XHTML y CSS.
- Visualización e interacción dinámicas mediante el DOM.
- Intercambio y manipulación de datos mediante XML y XSLT.
- Recuperación asíncrona de datos, con el uso de XMLHttpRequest.
- JavaScript como nexo de unión de todas estas tecnologías.

Ajax desempeñó un rol vital en la evolución de la Web, particularmente en la creación de aplicaciones web incrementando y enriqueciendo la experiencia de los usuarios [Mahemoff 2006].

Las aplicaciones enriquecidas (Rich Internet Applications: RIA) son aplicaciones web con rasgos “enriquecidos” que aportan su funcionalidad de una manera semejante a las aplicaciones de desktop. Ellas son el resultado de las modernas tecnologías de hoy en día que nos permiten crear avanzadas interfaces de usuario [Paul J. Deitel 2008]. Dentro de las características que tienen las RIA se encuentran:

- Las aplicaciones son contenidas dentro de un navegador.
- La interacción con las páginas es rápida debido a que estas no son cargadas completamente, solo el espacio que es modificado o donde se cargan datos nuevos.
- Los controles de la interfaz son más modernos, igualando a los controles de las aplicaciones de escritorio clásicas.
- Para el desarrollo de estas aplicaciones no es necesario grandes plataformas y son compatibles la mayoría de los navegadores.

- Para la creación de este tipo de aplicaciones se utilizan frameworks, librerías y herramientas que adicionan capacidades Ajax a los componentes estándares.

La variedad, riqueza y flexibilidad que proporcionan estas tecnologías transforman la interfaz de usuario en una experiencia visual interactiva, convirtiéndose en una de las formas más poderosas que existen para que las personas interactúen con los sistemas y entre sí.

Actualmente existe una gran cantidad de marcos de trabajo (frameworks) que brindan facilidades para construir aplicaciones con interfaces enriquecidas los cuales poseen características diferentes. Entre ellos se pueden mencionar: JQuery, Vaadin, ZK, GWT, Dojo Toolkit, Ext JS, YUI, MooTools, Silverlight, Richfaces y Prototype.

El autor de esta investigación considera que la selección correcta de un framework de este tipo es de vital importancia para el desarrollo de las interfaces de las aplicaciones compuestas. Esta selección se dificulta por no existir un conjunto de indicadores apropiados que relacionen capacidades del marco de trabajo con necesidades de la aplicación a construir.

1.4.3.2 Portales empresariales

En la actualidad los sitios Web se han convertido en destinos para que las comunidades de usuarios creen y compartan datos complejos y muy variados, para que analicen, evalúen los contenidos publicados, interactúen con otros usuarios y otras aplicaciones. La Web se ha convertido en una plataforma para aprovechar y fomentar la inteligencia colectiva, habilitando interfaces de usuario comunes y enriquecidas, con independencia de dispositivos y mecanismos de gestión e implementación de servicios y software [Paul J. Deitel 2008]. Los portales son ejemplos de uso de la web como plataforma unificada.

Los portales web son muy comunes hoy en día, cualquiera que ha navegado por la red se ha encontrado y ha interactuado con alguno de ellos [Jonas X. Yuan 2010]. Los portales proveen un único punto de entrada a una gran cantidad de información que se encuentra distribuida por la red y ofrece una misma manera de acceder a ella [Sarang 2009]. Existen definiciones diversas:

“Un portal web es un sitio que provee una única función vía página o sitio web funcionando como punto de acceso a la información en la WWW. Presenta información de diversas fuentes de una manera unificada. Aparte de mecanismos estándar de búsqueda, ofrece comúnmente servicios de correo, noticias, precios de productos, entre otros. Facilita a las empresas una vista común para el control, acceso y gestión de muchas de sus aplicaciones” [Sarang 2009].

“Un portal es una aplicación web que permite a los usuarios localizar, crear contenido relevante y usar aplicaciones que comúnmente necesitan para ser productivos” [Sezov 2012].

Los portales no son un remplazo para las aplicaciones web sino que están llamados a extender su funcionalidad. Han sido probados como piezas decisivas dentro de las organizaciones capaces de entregar mediante el uso de estándares, información y funcionalidades a los usuarios finales [Bernal 2009; Yuan 2009].

A pesar de que muchas de las definiciones describen a los portales como aplicaciones configurables que facilitan la gestión de información, olvidan mencionar los beneficios que se pueden obtener mediante la combinación de sus servicios. Ellos promueven la colaboración entre sus usuarios usando muchos de los elementos pertenecientes a la era de la Web 2.0 como son los blogs, los chats, las wikis, el uso de las redes sociales.

Para el desarrollo de un portal se requiere de información actualizada, procesada y adecuada a los usuarios, fácilmente accesible para el que la necesite por lo que el uso de los portales permite

un mayor aprovechamiento de la información. Se convierte en un excelente mecanismo de soporte para la gestión de la información y el conocimiento en una organización.

Una de las tecnologías más actuales para el desarrollo de portales y que forman parte de las características fundamentales que diferencian a los portales web de los llamados portales empresariales son los portlets. Ellos convierten al portal en una plataforma de aplicaciones que pueden ser compuestas.

Los portlets son aplicaciones Web dentro de los portales, son componentes gestionados por un contenedor que al recibir una petición de un usuario responden con contenidos dinámicos. Permiten la personalización, la presentación y la gestión de la seguridad al mismo tiempo [Jeff Linwood 2004] . Sarin define portal empresarial de manera siguiente:

Es una colección de mini aplicaciones web llamadas portlets que soportan la personalización, gestión de contenido, autenticación y configuración. Un portlet no es más que un componente de interfaz responsable de proveer al usuario final la información obtenida de las distintas aplicaciones, sistemas o fuentes de datos [Sarin 2010].

La principal responsabilidad de un portlet es generar fragmentos de marcadores (como HTML, XML y WML), que se visualicen en una página del portal, brindando facilidades para su configuración.

Basado en el estudio realizado, el autor de esta investigación considera que los portales empresariales son elementos importantes que deben formar parte de una infraestructura tecnológica porque habilitan interfaces y funcionalidades que proporcionan una experiencia enriquecida a los usuarios. Ellos permiten no sólo la publicación y gestión de contenido sino que facilitan la visualización de aplicaciones compuestas a través de portlets.

1.5 Sistemas basados en el conocimiento.

Uno de los aspectos asociados al desarrollo de aplicaciones compuestas está relacionado con el uso de las mejores experiencias y su fácil acceso. Existen ejemplos de organizaciones como CBDI [Sprott 2010] que mantienen bases de conocimiento privadas para guiar y depurar el proceso de desarrollo de aplicaciones basadas en servicios.

Existe mucho conocimiento asociado al desarrollo de aplicaciones compuestas pero generalmente este no se aplica (el conocimiento no tiene valor si no se aplica) debido a que no se garantiza su disponibilidad y que no están creadas las condiciones para su fácil acceso. El conocimiento debe verse como un recurso que todos deben utilizar para el beneficio general de una organización.

En un mundo donde la complejidad y los grandes volúmenes de información hacen más difícil el manejo del conocimiento, el uso de los Sistemas Basados en Conocimiento (SBC) se ha convertido en una estrategia importante para las organizaciones. Los SBC constituyen técnicas de Inteligencia Artificial válidas para abordar la construcción de sistemas inteligentes [Bello 2002]. Un SBC es un sistema que ayuda a la solución de problemas empleando una representación simbólica del conocimiento en un determinado dominio. Se puede usar particularmente en aquellas áreas que cuentan, como soporte básico, con el conocimiento de expertos como son la medicina, la industria, la educación, entre otras.

La Inteligencia Artificial tiene por objeto el estudio del comportamiento inteligente en las máquinas. Una de las formas más comunes en que los seres humanos resuelven cierto tipo de problemas es utilizando una especie de razonamiento y aprendizaje basado en analogías. En Inteligencia Artificial se le ha dado el nombre de Razonamiento Basado en Casos (RBC) a un paradigma de solución de problemas que se basa en la utilización de experiencias anteriores para resolver nuevos problemas.

El RBC es capaz de utilizar el conocimiento adquirido en situaciones previas y aplicarlo en una situación presente por lo que brindó un escenario favorable para su incorporación en el modelo propuesto en esta investigación. Un problema nuevo se resuelve buscando en la memoria un caso similar resuelto en el pasado. Al agregar nuevos casos la base se enriquece aún más, lo cual constituye actualizaciones del dominio, aumentando el conocimiento almacenado. El paradigma de razonamiento basado en casos se utiliza en esta investigación con el fin de emplear el conocimiento específico adquirido en situaciones previas y utilizarlo en el presente en la toma de decisiones.

Un nuevo problema se compara con los casos almacenados previamente en la base de casos (Memoria de Casos) y se recuperan uno o varios casos. Posteriormente se evalúa una solución sugerida, por los casos que han sido seleccionados con anterioridad, para tratar de aplicarlos al problema actual [León et al. 2010].

En la Inteligencia Artificial (IA), el RBC, ocupa un importante lugar ya que facilita el uso de la experiencia acumulada para la toma de decisiones sobre las nuevas situaciones que se presenten. La idea principal es construir sistemas cuya función sea actuar como un consejero o una memoria externa del tomador de decisiones.

En el desarrollo de aplicaciones compuestas es necesario almacenar las experiencias obtenidas en cuanto a herramientas, marcos de trabajo y servicios utilizados. Los principios del RBC permiten gestionar y seleccionar eficazmente esta información para facilitar el proceso de composición de aplicaciones.

El RBC es capaz de utilizar el conocimiento adquirido en situaciones previas y aplicarlo en una situación presente por lo que brindó un escenario favorable para su incorporación en el desarrollo de la presente investigación.

Conclusiones del capítulo.

Después de un análisis del estado del arte referido a las aplicaciones compuestas en un entorno SOA, concluimos que:

- Las aplicaciones compuestas en entornos orientados a servicios constituyen una evolución importante en la programación que puede revolucionar la ingeniería de software y la programación.
- Existen varias herramientas, marcos de trabajo y lenguajes que pueden formar parte de una infraestructura tecnológica para desarrollar y desplegar aplicaciones compuestas basadas en SOA. Es importante una correcta selección de la tecnología para evitar problemas de disponibilidad, de rendimiento o de incompatibilidades en este tipo de aplicaciones.
- Un modelo para desarrollar aplicaciones compuestas debe tomar en cuenta las características genéricas que presenten las herramientas que tomen parte en el proceso de composición de aplicaciones y un grupo de criterios que apoyan la selección tecnológica.
- Para la construcción de aplicaciones compuestas es importante la visualización a través de interfaces enriquecidas. Es necesario un conjunto de indicadores apropiados que permitan seleccionar un marco de trabajo RIA para cubrir las necesidades de los servicios de presentación.
- El paradigma de razonamiento basado en caso puede ser útil para la incorporación de las experiencias obtenidas en la composición de aplicaciones en los nuevos proyectos que se acometan.

CAPÍTULO II

MODELO PARA EL DESARROLLO DE APLICACIONES COMPUESTAS.

CAPÍTULO II – MODELO PARA EL DESARROLLO DE APLICACIONES

COMPUESTAS.

En este capítulo se explican modelos actuales para la composición de aplicaciones, se expone el resultado del diagnóstico realizado acerca del conocimiento y puesta en práctica, en instituciones productoras de software, de las aplicaciones compuestas. Se propone un modelo para la composición de aplicaciones, así como se fundamentan las indicaciones metodológicas que orientan a los desarrolladores que pretendan obtener aplicaciones informáticas mediante la composición de servicios.

2.1 Modelos y trabajos actuales relacionados con la composición de aplicaciones basadas en servicios.

En la bibliografía analizada se expone un grupo de trabajos relacionados con el desarrollo de aplicaciones compuestas en entornos orientados a servicios. Entre ellos se destacan:

- *“Building SOA-Based Composite Applications Using NetBeans IDE 6”*[Salter and Jennings 2008] Promueve el desarrollo de aplicaciones compuestas usando JBI (Java Business Integration) en plataforma libre (Open-ESB) perteneciente a Sun Microsystems, con un gran número de conectores y facilidades en el trabajo con el entorno de desarrollo (NetBeans IDE). Posterior a la fusión entre ORACLE y Sun Microsystems esta plataforma culminó su desarrollo y soporte y por lo tanto no continua accesible para los desarrolladores.
- *“An optimal QoS-based Web service selection scheme”*[Angus and Lan 2009], *“A Holistic Method for Selecting Web Services in Design of Composite Applications”*[Bonders and Grabis 2011] y *“An Efficient Qos-Based Ranking Model for*

Web Service Selection with Consideration of User's Requirement”[Mohebi 2012] son investigaciones que se enfocan en la selección basada en elementos de la calidad de servicio para desarrollar aplicaciones. Estas investigaciones no tienen como objetivo integrar los aspectos técnicos generales necesarios para construir aplicaciones compuestas, aunque aportan elementos importantes relacionados con la criticidad de una selección correcta de servicios, los requisitos que se deben tomar en cuenta para esta selección y la importancia del uso de estándares.

- “Rapid Development of Composite Applications Using Annotated Web Services”[Dannecker et al. 2010]. Desarrolla un método para agilizar el desarrollo de aplicaciones compuestas usando anotaciones en los servicios Web que describe su comportamiento y las características de la interfaz de usuario. Utiliza una herramienta propia (ServFaceBuilder) combinada con un modelo (no descrito) que se utiliza como formato de almacenamiento para la transformación al código, donde se interpretan las anotaciones y se generan interfaces de usuario. Aunque este autor considera que no hace un uso correcto de los estándares aceptados a nivel mundial para el desarrollo de aplicaciones compuestas, resulta muy positivo que los resultados de la composición de servicios se combinen con interfaces enriquecidas.

Como puede apreciarse, existen modelos y propuestas de trabajo para guiar el desarrollo de aplicaciones compuestas, pero en algunos de ellos es limitado el uso de los estándares, les falta integralidad y vinculación con el entorno nacional, por lo que fue necesario no solo estudiar la teoría sino hacer una profunda evaluación de la situación del país para elaborar un modelo que constituyera una guía en el propósito de estimular el desarrollo de aplicaciones compuestas y con ello elevar la eficiencia de la informática en Cuba.

2.2 Diagnóstico del desarrollo de aplicaciones compuestas en Cuba.

En la actualidad hay sectores de la industria y los servicios en Cuba que han incursionado en alternativas tecnológicas que incluyen elementos de Gestión por Procesos de Negocio, conocido como BPM (*Business Process Management*) y de SOA. En el caso de las instituciones de la Administración Central del Estado, el Ministerio del Turismo (MINTUR) ha adoptado la Gestión por Procesos como disciplina empresarial en algunas de sus entidades y ha comenzado a utilizar tecnologías como Microsoft SharePoint Server, producto considerado parte de la estrategia de Microsoft en el sector [León, Valdivia, Zaldívar, Santiago and Osorio 2010].

La empresa Tecnomática desarrolla una tecnología propia para proveer servicios a través de internet mediante el proyecto cubano, que cuenta ya con algunos resultados aplicados. Sin embargo en estos trabajos se evidencia la inmadurez del conocimiento en SOA en el contexto nacional el cual se centra en unos pocos especialistas dispersos que han estudiado el tema. Se evidencia la ausencia de centros académicos o productivos que estudien a profundidad estos paradigmas, las tecnologías asociadas a ellos, así como su uso y estandarización [León, Valdivia, Zaldívar, Santiago and Osorio 2010].

En el estudio realizado sobre la aplicación de SOA y del diseño de servicios en Cuba se pudo constatar que en el ámbito empresarial cubano se han comenzado a introducir los beneficios de BPM pero no de SOA y por ende tampoco se introducen los beneficios de las aplicaciones compuestas. En instituciones académicas y de desarrollo de software se han reconocido y utilizado las ventajas de proveer el software como servicio, sin embargo, en casi la totalidad de los casos, los servicios se desarrollan sin cumplir con los principios de diseño de SOA, ni las buenas prácticas de la orientación a servicios [León 2010].

2.2.1 Métodos aplicados en el diagnóstico.

Un diagnóstico es el proceso mediante el cual se lleva a cabo un análisis para recopilar información que ayude a determinar la situación actual de la organización y detectar sus áreas de mejoramiento. Mediante un diagnóstico se trata de focalizar y evaluar un conjunto de variables que juegan un importante papel en la comprensión, predicción y control del comportamiento de un fenómeno determinado [Shull et al. 2008].

Para realizar el diagnóstico de la presente investigación se realizaron entrevistas y se aplicaron encuestas a varios especialistas con experiencia en el desarrollo de proyectos con arquitecturas orientadas a servicios, arquitectos, desarrolladores de software, ejecutivos, profesionales y técnicos relacionados con la actividad del objeto de estudio en Cuba, Guatemala, República Dominicana, España y EU.

El objetivo principal del diagnóstico fue evaluar el estado en que se encuentra la utilización y conocimiento de la tecnología relacionada con el desarrollo de aplicaciones compuestas y el uso de las herramientas que conforman la infraestructura básica en entornos orientados a servicios. También se examinó la capacidad para componer los recursos existentes como activos de software en las instituciones y la interoperabilidad que existe entre estos recursos y los medios que los consumen, así como las causas que provocan los pocos avances obtenidos en la composición de aplicaciones a pesar del reconocimiento que tiene esta disciplina.

Para el diagnóstico se utilizó el método de investigación acción, entrevistas, el análisis de campos de fuerzas, el diagrama causa-efecto, el grupo focal y el estudio de casos.

Los principales aspectos analizados fueron:

- a) Estado de utilización del Registro/Repositorio de servicios. Importancia de la utilización de Registro/Repositorio de servicios para estimular la reusabilidad en el proceso de desarrollo de aplicaciones informáticas.
- b) El cumplimiento con los estándares, principios de diseño de SOA y buenas prácticas en los servicios publicados.
- c) Grado de conocimiento de las herramientas y tecnologías utilizadas para componer aplicaciones en entornos orientados a servicios.
- d) Interés de las organizaciones en adoptar SOA y desarrollar aplicaciones compuestas.

El análisis de estos aspectos se realizó a partir de las problemáticas detectadas al aplicar los métodos y técnicas de investigación planteadas anteriormente, lo cual se describe a continuación:

- **Encuestas:** Se realizaron encuestas aplicando un grupo de cuestionarios previamente elaborados con lo que se conoció la valoración de expertos con respecto a temas relacionados con el objeto de estudio.
- **Entrevista a profundidad:** Fue realizada a varios especialistas y doctores en la especialidad de informática con experiencias en el desarrollo de aplicaciones informáticas.
- **Grupo Focal:** Se aplicó la técnica a un grupo de nueve especialistas empleando una guía de preguntas y con la participación de un moderador.
- **Investigación acción:** Constituyó un método decisivo en el proceso de identificación del problema específico que se trata de resolver mediante la acción. Resultó muy importante su aplicación para detectar las principales causas que limitan la composición de aplicaciones y poder ejecutar acciones para solucionarlas.
- **Observación Participante:** El autor de la investigación fue participante activo en una gran parte de los procesos desarrollados.

A pesar del reconocimiento de la comunidad científica de la importancia y necesidad de desarrollar aplicaciones compuestas, esta concepción no ha logrado imponerse debido a la existencia de un conjunto de dificultades y deficiencias que fueron identificadas en el diagnóstico realizado. A continuación se describen los principales resultados obtenidos:

a) Estado de utilización del Registro/Repositorio de servicios. Importancia de la utilización de Registro/Repositorio de servicios para estimular la reusabilidad en el proceso de desarrollo de aplicaciones informáticas:

Según las encuestas realizadas se pudo comprobar que de un total de 30 especialistas encuestados de tres instituciones productoras de software el 95% afirman que resulta muy importante el disponer de un Registro/Repositorio para propiciar el consumo y publicación de servicios como recursos y así fomentar la estandarización que permita la interoperabilidad en el desarrollo de aplicaciones compuestas.

A pesar de que el 50% respondió que sí ha consumido servicios en el proceso de desarrollo de aplicaciones en algún momento, solo el 10% manifiesta que ha accedido a ellos a través de un Registro/Repositorio. El 85% quisiera reusar funcionalidades expuestas como servicios pero desconoce las herramientas para hacerlo. El 96% de los encuestados destacó la importancia de utilizar repositorios para estimular el desarrollo de servicios así como su adecuada gestión en el desarrollo de aplicaciones.

Almacenar servicios que cumplan con los principios de diseño de SOA, es fundamental para facilitar la composición de aplicaciones. Estos resultados ponen de relieve las carencias en el dominio de aspectos tecnológicos y falta de conocimiento del paradigma orientado a servicios.

b) El cumplimiento con los estándares, principios de diseño de SOA y buenas prácticas en los servicios publicados.

El 93% de los encuestados reconoce que los servicios consumidos no cumplen los **estándares** ni principios de diseño, al mismo tiempo que el 90% manifiesta desistir del consumo de determinados servicios por problemas claros de **interoperabilidad**. El incumplimiento de estos aspectos prácticamente invalida el uso de las aplicaciones compuestas para los desafíos de negocios que se presentan en la actualidad basados en las ventajas que las mismas ofrecen.

La encuesta para determinar el grado de conocimiento de las herramientas y tecnologías utilizadas para desarrollar aplicaciones compuestas, arrojó resultados de interés que se detallan en el Anexo 1.

c) Interés de las organizaciones en adoptar el paradigma orientado a servicios y desarrollar aplicaciones compuestas.

El 100% de los encuestados manifestó el interés de la organización de adoptar el paradigma SOA y el desarrollo de aplicaciones compuestas. Esto demuestra el reconocimiento de la importancia de su adopción a pesar de todas las dificultades y deficiencias que se presentan.

También se llevaron a cabo otras tareas como muestra la Figura 2.1.

- **Determinación de las fuerzas impulsoras y restringentes que inciden en el desarrollo de aplicaciones compuestas.**

Un aspecto fundamental en el que se profundizó durante el diagnóstico fue lo relacionado con el logro de un alto nivel de composición entre aplicaciones. Para ello se aplicó la técnica de Campo de Fuerzas. Para la selección de los doce participantes se tomaron en cuenta varios criterios

basados en la experiencia, los roles y la relación entre las temáticas laborales de estos y el objetivo principal de esta investigación.



Figura 2.1 Tareas desarrolladas durante el diagnóstico. Fuente: Elaboración propia.

Se escogieron profesores que imparten asignaturas como Programación e Ingeniería de Software, así como ejecutivos, gerentes de proyectos, líderes técnicos de proyectos productivos nacionales e internacionales. Además se seleccionaron arquitectos de varios centros productores de la UCI para confirmar el estado de conocimiento del tema a investigar y su nivel de actualidad. Se seleccionaron también expertos de otros centros productores de software como COPEXTEL y SOFTEL. La composición de los expertos involucrados se muestra en detalle en el Anexo 2.

Los resultados de la aplicación de esta técnica se muestran en el Anexo 3. Ello evidencia la jerarquía que le dan los participantes a la composición de servicios, destacándose la importancia de la estandarización, de la preparación de los recursos humanos y la necesidad de disponer de las mejores experiencias en el desarrollo de aplicaciones.

- **Mapa de conocimientos de tecnologías relacionadas con el desarrollo de aplicaciones compuestas.**

La gestión de conocimiento es fundamental en el desarrollo de aplicaciones compuestas. Es importante facilitar el acceso al conocimiento tácito y explícito existente. El conocimiento tácito se encuentra en los ejecutivos, especialistas, arquitectos, desarrolladores, entre otros.

Todos ellos participan en los diferentes procesos relacionados con la gestión de proyectos y la construcción de aplicaciones informáticas, acumulando gran experiencia en su trabajo. Una forma de organizar y socializar tecnologías para el desarrollo de aplicaciones informáticas se presenta en el Anexo 4.

- **Necesidades de aprendizaje asociadas al desarrollo de aplicaciones compuestas.**

Un aspecto considerado en el diagnóstico por la importancia que reviste es la determinación de necesidades de aprendizaje asociadas al desarrollo de aplicaciones compuestas. Uno de los aspectos que se destacan con mayores dificultades es la preparación de los recursos humanos para asimilar este paradigma. De acuerdo a la investigación realizada en el CDAE, donde participó el autor de esta investigación, la cual estuvo basada en el análisis documental y la experiencia acumulada, se definieron necesidades de aprendizaje asociadas al desarrollo de aplicaciones compuestas que se describen en el Anexo 5.

- **Elaboración del diagrama causa-efecto.**

Como parte del diagnóstico realizado se elaboró un diagrama causa- efecto que se muestra en el Anexo 6 agrupándose las deficiencias e insuficiencias destacadas en dos variables causales fundamentales:

- **Recursos Humanos.** Se destacó la poca experiencia en los trabajos realizados anteriormente relacionados con la composición de aplicaciones. Existe poco intercambio de información entre aplicaciones y poca exposición de funcionalidades como servicios lo cual, unido a la falta de interacción, el poco intercambio de experiencias y conocimientos entre, conlleva a la duplicidad de esfuerzos y al no aprovechamiento de las bondades de la composición de aplicaciones por falta de criterios y conocimiento en los desarrolladores.
- **Elementos de Software.** Existe un insuficiente conocimiento de las herramientas y marcos de trabajo existentes en el mundo para aprovechar las ventajas de la composición de aplicaciones y cuando se conocen no se explotan al máximo. Existen en el mercado muchas herramientas para el desarrollo de aplicaciones compuestas pero no existen criterios para su proceso de selección. Además se puede identificar un marcado protagonismo de las herramientas propietarias en el mundo, las cuales son muy completas y competentes, pero Cuba no está en posibilidades de acceder a ellas muchas veces por lo costoso de sus licencias y otras veces por temas legales.

El resultado de este diagnóstico refleja la importancia y necesidad de elaborar una herramienta que sirva de apoyo a los desarrolladores en la creación de aplicaciones compuestas basadas en servicios, que contemple un conjunto de componentes básicos que facilite la toma de decisiones en las diferentes etapas del proceso de composición. Para ello se propone un modelo que se presenta y fundamenta a continuación.

2.3 La orientación a servicios y la composición de aplicaciones en la UCI.

En el año 2009 la Universidad de las Ciencias Informáticas creó el Centro de Consultoría y Desarrollo de Aplicaciones Empresariales (CDAE) cuyo objetivo es desarrollar aplicaciones

empresariales y ofrecer servicios de consultoría. Surge la necesidad de crear una arquitectura de referencia, un proceso de desarrollo, un modelo de madurez, una estrategia para componer aplicaciones y una infraestructura tecnológica. Todo ello basado en la experiencia internacional existente y la necesidad de desarrollo de la industria de software cubana.

Como resultado del 1er Taller de Arquitectura de Software efectuado en el 2007 en la UCI, se planteó como una intención de la universidad, investigar el uso de tecnologías BPM/SOA basadas en software libre, materializándose esto con la proyección en el 2008 de un centro que estudiara y aplicara estas tecnologías y enfoques. En el año 2009 quedó constituido oficialmente el Centro de Consultoría Tecnológica e Integración de Sistemas de la UCI que contempló entre sus líneas de investigación el enfoque BPM/SOA.[León, Valdivia, Zaldívar, Santiago and Osorio 2010]

En muchos de los grandes y medianos proyectos como el ERP Cubano, Guardián del Alba, y otros para la informatización del sistema de salud cubano y venezolanos, así como en proyectos para la automatización de procesos internos de la UCI, se han provisto funcionalidades como servicios para facilitar los mecanismos de integración con otras aplicaciones.

2.3.1 Estudio de casos.

- **Repositorio de servicios de la UCI**

En las organizaciones que emprenden iniciativas orientadas a servicios, la disponibilidad de los recursos y la prestación de servicios satisfactoria e ininterrumpida es un elemento crítico en el éxito del funcionamiento de las aplicaciones.

La UCI cuenta con un repositorio de servicios que fue creado con el objetivo de reutilizar recursos, sin embargo el mismo no ha cumplido su objetivo debido al incumplimiento en los principios de diseño de SOA lo que afecta la interoperabilidad en el consumo de estos.

En el mencionado repositorio se expone información sobre algunos servicios de uso general y de utilidad para los usuarios de la red. Entre estos se encuentra:

1. Gestión académica.
2. Gestión de Identidad.
3. Información telemática.
4. Información de los residentes.
5. Información personal de los trabajadores.
6. Reservaciones de Pase.

El principal problema que presenta este sitio es que carece de la estructura básica definida para repositorio de servicios de acuerdo al **estándar** UDDI y carece también de las funcionalidades básicas para el descubrimiento y publicación de servicios. Además de estas limitaciones presenta también incumplimiento con los principios de diseño de SOA en muchos de sus servicios.

1. Gestión académica (Akademos).

Por ejemplo, la especificación para el servicio de Akademos tiene una ubicación errónea de su contrato de servicio al no estar publicado donde debe (tiene una dirección diferente a la que aparece publicada) lo que atenta contra la disponibilidad. Otros problemas que presenta este servicio son:

- El WSDL no cumple con el perfil WS-I y fue generado automáticamente por una herramienta.
- Faltan numerosas definiciones de los namespaces en el enlace directo al WSDL, mientras que en el sitio sí aparecen, evidenciando una falta de sincronización.

- Se usan referencias a tipos de datos, como los array, que son propios de una plataforma o lenguaje de programación.
- En la definición de los mensajes se usa para especificar la estructura del mismo “type” en vez de “element”.
- En el binding el estilo es RPC, cuando el más aceptado es Document lo cual disminuye la interoperabilidad.
- El valor del use es Encoded en vez de Literal disminuyendo también la interoperabilidad.
- El servicio posee numerosas operaciones, en total son 13, cuando las recomendaciones oscilan entre 4 y 7 operaciones por servicio. Más allá pueden aparecer problemas en la gestión del servicio.

Aunque el diseño no está del todo bien, el servicio funciona y puede ser consumido, pero con evidentes problemas de **interoperabilidad** si se usa una plataforma distinta de PHP por los puntos listados anteriormente [Infante 2012].

Otro problema que existe es el relacionado con la gestión del ciclo de vida de estos servicios, así como con la meta información y las facilidades para descubrirlos y consumirlos.

2. Los servicios Gestión de Identidad e Información Telemática.

Estos presentan problemas similares al caso anterior.

En todos estos casos resulta imposible la composición de aplicaciones usando estándares. Se destacan problemas de interoperabilidad, en el cumplimiento de los principios de la orientación a servicios, en el diseño de los recursos, entre otros. En el desarrollo de aplicaciones compuestas el diseño de servicios desempeña un papel significativo en la previsión de defectos que puedan introducirse en etapas posteriores. Un diseño inapropiado de servicios puede conducir a una pobre reutilización y aplicaciones compuestas innecesariamente complejas.

El modelo desarrollado como parte de esta investigación incorpora en su infraestructura tecnológica un registro repositorio que permite la gestión del ciclo de vida de los servicios y facilidades para su descubrimiento. El modelo también tiene en cuenta los principios de diseño de SOA y las buenas prácticas definidas en [Erl 2007; Erl 2008; Erl, Karmarkar, Priscilla Walmsley, Hugo Haas, Umit Yalcinalp, Canyon Kevin Liu, David Orchard, Andre Tost and Pasley 2010] como elementos necesarios para el uso correcto de estándares y el aumento de la interoperabilidad, como uno de los beneficios principales que promueve la orientación a servicios.

2.4 Modelo conceptual para el desarrollo de aplicaciones compuestas, MIDAC.

El término modelo proviene del italiano “modello” – representación de algo que se debe seguir o imitar. En la literatura también otras definiciones válidas como:

- “un instrumento de la investigación creado para reproducir el objeto que se está estudiando, por tanto es una representación simplificada de la realidad se cumple una función heurística que descubre nuevas relaciones y cualidades del objeto de estudio”
- “en el proceso de modelación el conocimiento parece trasladado temporalmente del objeto que le interesa a la investigación de un cuasi–objeto intermedio, auxiliar: el modelo.”.

Un modelo permite, desde una nueva perspectiva de análisis, una comprensión más plena del objeto de estudio para resolver un problema y representarlo de alguna forma. En el proceso de modelación, el eslabón mediato es el modelo que actúa como función representativa sustituyendo el objeto. A continuación se expone el modelo teórico propuesto teniendo en cuenta su objetivo, principios, cualidades y premisas.

Es objetivo del modelo describir y representar los diferentes componentes de carácter tecnológico, de conocimiento y las interrelaciones existentes entre ellos que contribuyan a su vez a la composición de aplicaciones basadas en SOA.

A los efectos de la presente investigación se considera el modelo como una representación ideal del fenómeno a investigar que contempla aquellos elementos y relaciones que se consideran esenciales y sistematizan el objeto modelado. El mismo refleja la realidad de acuerdo con la intención del investigador.

Integrando los principales fundamentos teóricos establecidos en el capítulo 1 con las características, estado actual y desarrollo de las aplicaciones compuestas en Cuba, el autor desarrolla un Modelo Integrado para el Desarrollo de Aplicaciones Compuestas (MIDAC), que contribuye a minimizar las fuerzas restringentes obtenidas con la aplicación de la técnica de campo de fuerzas, actúa sobre las causas que afectan el desarrollo de aplicaciones compuestas y constituye una solución al problema de investigación identificado, cumpliendo la hipótesis planteada.

El modelo, que constituye el aporte fundamental de la tesis, es definido por el autor de la manera siguiente: Es la representación conceptual que integra herramientas, buenas prácticas, marcos de trabajo y una base de conocimiento de apoyo a la toma de decisiones para facilitar el desarrollo de aplicaciones compuestas.

Para construir el modelo el autor aplicó un procedimiento que utiliza las tendencias teóricas actuales sobre composición de aplicaciones en entornos orientados a servicios y tiene en cuenta el diagnóstico realizado para introducirlos en las condiciones de Cuba.

2.4.1 Procedimiento empleado para la construcción del modelo.

La construcción conceptual del modelo exigió un diseño metodológico que abarca toda la estructura de la tesis y constituye una combinación de teoría y práctica. La vida del modelo sin embargo, lo constituye la dinámica de su funcionamiento y su aplicabilidad. Para ello el autor lo fue utilizando para construir aplicaciones reales en la medida que fue perfeccionando su construcción y desarrollo. Lo anterior explica como el modelo presentará componentes que fueron desarrollados apoyados en la concepción del propio modelo, basado en el principio de recursividad.

En la Figura 2.2 se representa el proceder metodológico de la investigación con el enfoque asumido para el desarrollo del modelo, que se argumenta a continuación:



Figura 2.2 Concepción metodológica seguida para elaborar el modelo Fuente: Elaboración propia

2.5 Principios, cualidades, premisas y componentes para el desarrollo del modelo para aplicaciones compuestas basado en SOA.

Los principios que sustentan la construcción de un modelo propuesto para la composición de aplicaciones son:

- a) La **estandarización** para hacer más eficiente el proceso de composición y reutilización de recursos. Esto se refiere a la utilización de los estándares ampliamente aceptados y utilizados en la práctica mundial para componer aplicaciones basadas en servicios. Dentro de ellos están WS-BPEL, WSDL, SOAP, UDDI que fueron explicados en el capítulo 1 de esta tesis.
- b) **La interoperabilidad** para lo cual es indispensable un diseño adecuado de los recursos, que permita que las aplicaciones sean compatibles y usables sin importar tecnología subyacente.
- c) **La flexibilidad** que se logra utilizando componentes con funcionalidades genéricas y que se adapten a las particularidades de las aplicaciones existentes para el proceso de composición.
- d) La **pertinencia** como garantía de la adecuación del modelo en el contexto de la orientación a servicios.
- e) La **independencia funcional** de los servicios lo que contribuye a su bajo acoplamiento y a su reutilización.
- f) La **actualización** permanente mediante la retroalimentación de la información que nutre al modelo.

Las **cualidades** del modelo son las siguientes:

- a) **Amplitud** que brinda la capacidad de analizar y de emplearse en aplicaciones informáticas de múltiples dominios.
- b) **Enfoque Sistémico:** Se expresa en el modelo propuesto a través de los componentes que interactúan con vistas a perfeccionar el proceso de composición de aplicaciones.

- c) **Integralidad** dada por los componentes del modelo que cubren de manera integrada y coherente la mayoría de los elementos necesarios para la composición satisfactoria de aplicaciones.
- d) **Mejora continua**: El modelo constantemente se mejora con los resultados que se van obteniendo, en particular con la base de conocimiento incorporada en el componente de selección inteligente de la información.

Las premisas con vistas a la aplicación del modelo propuesto son:

- a) La **calificación** de los desarrolladores necesaria para el uso eficiente de las herramientas propuestas para la composición de aplicaciones.
- b) La **voluntad institucional** que apoye la aplicación del modelo y la visibilidad de los recursos necesarios para la composición de aplicaciones.
- c) La existencia de un **repositorio** de servicios debidamente catalogados.

Los principales componentes del modelo son:

- La gestión de conocimiento.
- La gestión tecnológica.
- La gestión de recursos.
- Sistema inteligente selección de información.

En la Figura 2.3 se muestra una gráfica con la interacción de estos componentes y la interrelación que se produce entre ellos y en el epígrafe 2.6.1 se describe el papel que desempeña cada componente.

2.6 Estructura del modelo para el desarrollo de aplicaciones compuestas, MIDAC.

El núcleo del modelo está formado por componentes relacionados entre sí como se muestra en la Figura 2.3.



Figura 2.3. Representación gráfica del modelo. Fuente: Elaboración propia

2.6.1 Descripción de los componentes del modelo

a) Componente: Gestión Tecnológica

Este componente incluye las tecnologías vinculadas al desarrollo de aplicaciones compuestas. Entre ellas se encuentran los marcos de trabajo para el desarrollo de servicios: Axis2, Apache CXF, JAX-WS, JAX-RS, Spring Framework [Fisher, Partner, Bogoevici and Fuld 2012], EJB (Enterprise Java Beans) [Kumar et al. 2009]. Existe también un grupo de herramientas que permiten desplegar aplicaciones que componen servicios como son servidor de servicios, servidor de procesos de negocio, bus de mensajería, servidor de mashups, contenedor de portlets y contenedor de aplicaciones Web. Además incluye un grupo de marcos de trabajo para el desarrollo de interfaces enriquecidas como son: Vaadin, Richfaces, ZK, GWT, Silverlighth, JQuery, Dojo Toolkit, Prototype y Ext JS.

- **Marco de herramientas:** en el componente tecnológico el modelo contiene un grupo de herramientas seleccionadas de acuerdo a su funcionalidad para el desarrollo de aplicaciones compuestas lo que fue comprobado por el autor de esta investigación en entornos tecnológicos nacionales e internacionales.
- **Selección Framework RIA:** El componente incluye un grupo de criterios para el desarrollo de los servicios de presentación. Específicamente apoya la selección de un marco de trabajo adecuado para la construcción de una interfaz enriquecida según las necesidades del usuario, tomando información de una base de casos estructurada y con información clasificada de los marcos de trabajo RIA. Esta información fue basada en el reporte realizado por la consultora Gartner [Neil 2011] acerca de las tecnologías más influyentes para el desarrollo de aplicaciones enriquecidas. Entre los indicadores se encuentran:
 - Soporte de especificaciones de portlets.
 - Tipo de Licencia.
 - Complemento en algún entorno de desarrollo (plug-in).
 - Componentes que soporten la visualización en dispositivos móviles con tecnología táctil.
 - Clasificación (puede ser del lado del servidor o del lado del cliente).
 - Navegadores que soporta.
 - Requiere programación en JavaScript.

b) Componente: Gestión de Conocimiento.

La gestión de conocimiento es un proceso fundamental en el modelo. Se gestiona tanto conocimiento tácito como explícito necesario para la creación de la base de conocimientos, la

selección tecnológica y de recursos. En MIDAC la gestión del conocimiento constituye un pilar para el desarrollo de los procesos que lo integran pues ellos tienen como base el empleo del conocimiento para la toma de decisiones.

Las experiencias que incorpora la base de conocimiento son tomadas de los expertos de instituciones que fueron analizadas en el diagnóstico. La base de conocimientos tiene la posibilidad de enriquecerse mediante una aplicación que usa el paradigma de razonamiento basado en casos para gestionar las experiencias anteriores. Los casos se recuperan atendiendo a la evaluación de los rasgos de la base de casos. Los rasgos de esta base de casos son los siguientes:

- Marcos de trabajo para el desarrollo de servicios.
- Marcos de trabajo para el desarrollo de la interfaz.
- Estándares utilizados.
- Herramientas para desarrollo de la aplicación.
- Lenguaje de programación.
- Lenguaje para la composición.
- Buenas prácticas.
- Ubicación de los contratos de servicio.

Hasta el momento la base contiene 47 casos de desarrollo real y de casos de estudio relacionados con el desarrollo de aplicaciones compuestas.

c) Componente: Gestión de recursos.

El componente de gestión de recursos incluye una herramienta Registro/Repositorio que, como se ha mencionado anteriormente, resulta decisiva para el desarrollo de aplicaciones compuestas en entornos orientados a servicios. Esta herramienta permite la correcta

catalogación y gestión del ciclo de vida de los servicios, así como el descubrimiento de los mismos utilizando estándares aceptados a nivel mundial.

En este componente se seleccionan los servicios a componer en la construcción de las aplicaciones. Uno de los aspectos necesario para seleccionar los servicios es que exista una concordancia entre la funcionalidad descrita en la meta información y la necesidad de la aplicación que se desarrolla. Otro de los aspectos necesario es que los tiempos de respuesta de los servicios satisfagan las necesidades de quien los componga para construir la aplicación. Los tiempos de respuesta son incluidos en la bibliografía dentro del grupo de indicadores de rendimiento (KPI: Key Performance Indicators) y son monitoreados (realizando tareas de BAM: Business Activity Monitor) por las herramientas que forman parte de la infraestructura tecnológica.

d) Componente: Selección inteligente de información.

Este componente se fundamenta en una base de conocimiento expresada en casos y una herramienta (SI-Holmes) para la creación de la base de casos y la realización de diagnósticos como apoyo a la toma de decisiones. La misma se realizó en una de sus versiones utilizando el modelo desarrollado evidenciándose su factibilidad y las ventajas de la aplicación del modelo sobre todo en las facilidades que brinda en la composición de servicios.

SI-Holmes es una aplicación desarrollada en el lenguaje Java. Este lenguaje está muy extendido, encierra su fuerza en su independencia de la plataforma, seguridad, portabilidad y escalabilidad [Febles 2011]. SI-Holmes es una aplicación Web, con una arquitectura tres capas (Capa de Datos, Negocio e Interfaz) lo cual le provee algunas ventajas como la robustez debida al encapsulamiento, la facilidad de mantenimiento, soporte y flexibilidad. Este sistema posee una interfaz amigable y configurable como se aprecia en la Figura 2.4.

Tiene la ventaja de interactuar al mismo tiempo con múltiples bases de casos que se encuentran almacenadas en una base de datos, mostrando de cada una de ellas, sus nombres, cantidades de rasgos y cantidades de casos.

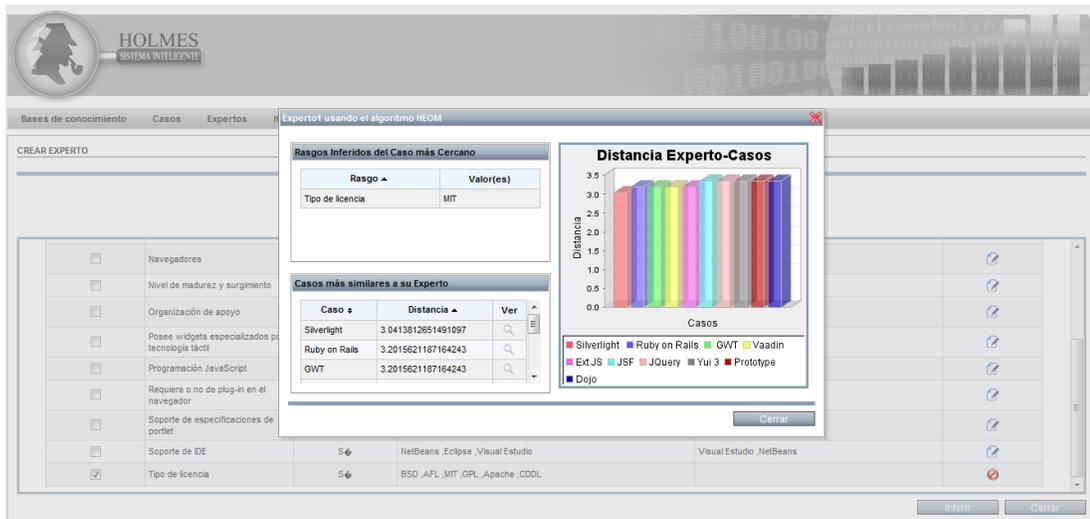


Figura 2.4 Interfaz de herramienta SI-Holmes. Fuente: Elaboración propia

Para obtener una solución de experto se selecciona el o los rasgos a inferir, se llenan los otros que se consideren necesarios, se escoge alguno de los algoritmos implementados para la distancia entre casos (HEOM, GOWER, ARGELIO) y el sistema le indicará el resultado de la inferencia brindada por el experto y los casos más cercanos de manera gráfica y detallada. El valor fundamental del Sistema Inteligente SI-Holmes lo constituye su simplicidad y su ambiente amigable. Se ha comprobado su funcionalidad en aplicaciones a la gestión universitaria y en sistemas de jurisprudencia con resultados satisfactorios.

2.6.2 Funcionamiento del modelo MIDAC.

El modelo MIDAC, como se ha planteado, contempla un conjunto de componentes que facilitan la composición de aplicaciones contribuyendo a la interoperabilidad y estandarización del proceso. El funcionamiento del modelo se representa en la Figura 2.5 y se explica a continuación:

Como entrada del modelo se requiere la siguiente información:

- Tiempo de respuesta mínimo de los servicios a consumir.
- Alcance de la composición.
- Necesidades del marco de trabajo para el desarrollo de la interfaz de usuario.
- Si no requiere de interfaz se debe entregar una descripción candidata del contrato de servicio que se exponen con los resultados de la composición.

Un proceso fundamental es seleccionar la tecnología a utilizar. Para ello se cuenta con una base de conocimiento expresada en casos y la herramienta SI-Holmes, que brinda asesoría para la toma de decisiones sobre qué elementos tecnológicos son aconsejables para facilitar la composición. Esto se basa en experiencias exitosas anteriores.

Para la seleccionar los recursos a componer se toma información del Registro/Repositorio acerca de los servicios existentes, lo cual es una premisa del modelo. A partir de la descripción funcional de los servicios, la concordancia entre los requisitos del primer paso, los resultados arrojados por sus indicadores de rendimiento basado en “Ranking Model for Web Service Selection” [Mohebi 2012] se indica la ubicación de los contratos de los servicios a seleccionar para la composición. Si no existe un servicio que concuerde con las necesidades planteadas entonces será necesario implementarlo, desplegarlo, probarlo y catalogarlo debidamente en el Registro/Repositorio para que sea descubierto y consumido en otras composiciones.

La selección inteligente de información con el apoyo de la herramienta SI-Holmes es la base para la selección tecnológica y de recursos en el proceso de composición de aplicaciones. Existen indicadores de rendimiento (los tiempos de respuesta máximo y mínimo de los servicios)

que son monitoreados de manera sistemática por las herramientas que forman parte de la infraestructura tecnológica. Estos indicadores son la base para la selección de los servicios.

Un componente esencial del modelo es el vinculado a la Gestión del Conocimiento, la cual se refiere al conjunto de procesos desarrollados en una organización para crear, organizar, almacenar y transferir el conocimiento. Actualmente este concepto es muy empleado en las organizaciones que se preocupan por transferir el conocimiento y las experiencias de sus recursos humanos de forma tal que quede disponible y pueda ser utilizado por otros miembros de la organización. Generalmente este proceso implica el empleo de diversas técnicas para capturar, organizar y almacenar el conocimiento del personal de la organización para transformarlo en un activo intelectual que brinde beneficios y se pueda compartir.

En la actualidad las TIC cuentan con herramientas que apoyan este proceso junto con sistemas diseñados para dar mejor uso al conocimiento como es el caso de la herramienta SI-Holmes en la presente investigación. Existen estudios que han determinado que una parte importante del valor de una organización, se relaciona con sus activos intangibles, de los cuales el conocimiento es un activo fundamental.

La organización adquiere conocimiento de una gran variedad de formas, según el tipo de conocimiento que se esté buscando. En la fase de captura y creación del conocimiento, el conocimiento tácito o explícito de la organización se captura a través de una variedad de técnicas entre las que se encuentran las entrevistas a expertos, el análisis documental, los grupos focales, los sistemas inteligentes que usan patrones o estrategias de búsquedas particulares, etc. los que han sido empleados en distintas etapas de la investigación.

2.7 Infraestructura tecnológica para la aplicación del modelo.

El ritmo creciente con el que se adopta SOA y el impacto que ha tenido ha traído un gran número de proveedores de software que compiten por una cuota de mercado en la provisión de la infraestructura tecnológica requerida. Actualmente, los proveedores de plataformas SOA se dirigen principalmente a las grandes corporaciones, por lo que el mercado está dominado por un escaso volumen de ventas y un costo asociado muy elevado.

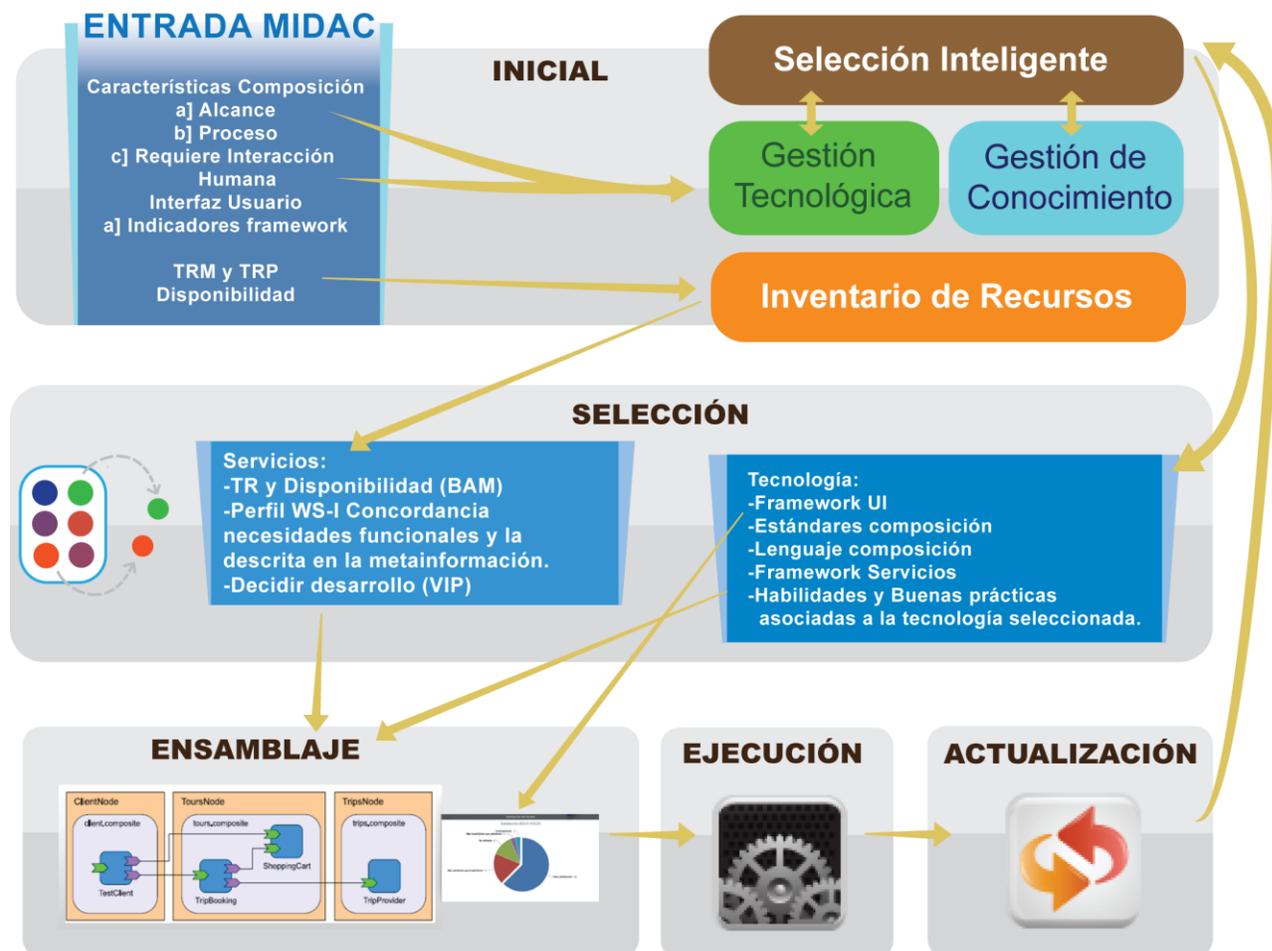


Figura 2.5 Esquema del funcionamiento de MIDAC. Fuente: Elaboración propia

Para proponer una infraestructura para MIDAC fueron analizadas las suites SOA que se sitúan como líderes en el mercado internacional. Entre ellas las suites de Oracle [Panda and Maheshwari 2009], IBM, Software AG y Microsoft. A pesar de que su costo y licenciamiento no son acordes con el entorno cubano, la calidad, la distribución y las características de sus

componentes se tomaron en cuenta para la infraestructura de MIDAC la cual está compuesta por los siguientes componentes:

- **Servidor de aplicaciones:** Permite desplegar y gestionar aplicaciones Web, servicios, procesos de negocio y mashups desarrollados usando varios marcos de trabajo de desarrollo. Tiene soporte total para estándares basados en XML como SOAP y WSDL.
- **Bus de mensajería:** Es la columna vertebral de la infraestructura. Permite la creación de servicios proxy para desarrollar la transparencia de ubicaciones, la transformación, enriquecimiento y ruteo de mensajes, el monitoreo de las comunicaciones, el manejo de eventos y diferentes mecanismos de seguridad.
- **Registro/Repositorio:** Indispensable como estructura de almacenamiento y control de las acciones referentes los elementos fundamentales, o sea, los servicios. Almacena los metadatos de los servicios como pueden ser su WSDL, XSD, políticas, y especificaciones técnicas. Permite gestionar el ciclo de vida de los servicios así como su descubrimiento.
- **Servicios de presentación:** Existen para enriquecer la interacción entre el usuario y el resultado de la composición. Está compuesto por aplicaciones RIA y portlets que puedan ser visualizadas tanto en navegadores convencionales como en dispositivos móviles.

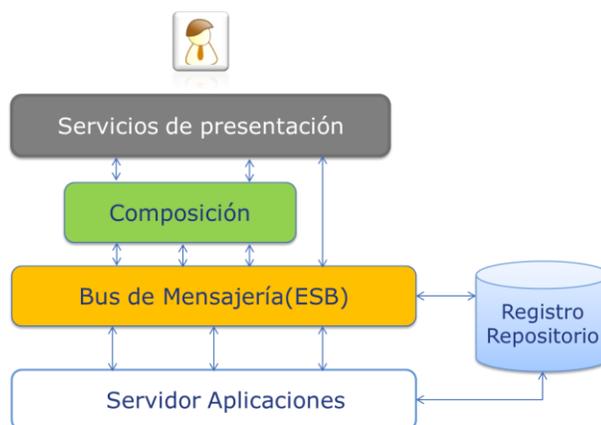


Figura 2.6 Infraestructura propuesta para MIDAC Fuente: Elaboración Propia

El autor de esta investigación tuvo presente criterios expuestos en [Sopeña, Camino and Navarro 2008] para seleccionar herramientas de código abierto que permitiesen el desarrollo de aplicaciones compuestas en entornos orientados a servicios y coincidieran con las características de los componentes de MIDAC. Entre estos criterios se encuentran:

- Referencias documentadas acerca del estado de madurez de la tecnología.
- Continuidad y el ritmo con el que debe evolucionar esta tecnología.
- La disponibilidad de servicios de mantenimiento y soporte adecuados para el producto o gama de productos considerados.
- El tamaño de la comunidad que soporta los componentes que se pretenden utilizar.

La suite WSO2 (*Web Services Oxygen*) posee las herramientas necesarias para soportar el funcionamiento de MIDAC. WSO2 es una compañía que ha desarrollado un conjunto de herramientas de código abierto basado en **estándares** que permite la adopción tecnológica de una arquitectura orientada a servicios. Su plataforma es basada en componentes usando la tecnología OSGI (*Open Services Gateway Initiative*). Ha sido ubicado como visionario en el Cuadro de Gartner entre las suites de SOA.

Entre las razones de mayor peso para su elección se encuentra su nivel de cumplimiento con los **estándares** WS-* para el desarrollo de servicios web, su tipo de licencia (Apache v2), la calidad de su documentación, su característica de multiplataforma, sus facilidades de trabajo en entornos que necesitan de una alta disponibilidad y su buen rendimiento. El autor de esta investigación considera que las herramientas AS (*Application Server*) , ESB (*Enterprise Service Bus*) [Ferguson, Pilarinos and Shewchuk 2007], y GREG (*Governance Registry*) [Dirksen 2012] y Liferay Portal [Sezov 2012; Yuan 2009] son las mejores herramientas libres de su tipo para el desarrollo de aplicaciones compuestas en entornos orientados a servicios.

2.8 Indicaciones metodológicas para la implantación del modelo propuesto.

A partir del análisis de los modelos de implementación propuestos en la literatura y de la experiencia en el desarrollo de soluciones informáticas en la UCI se plantean las indicaciones metodológicas para la implementación del modelo compuesto por 5 etapas o fases (Figura 2.7)

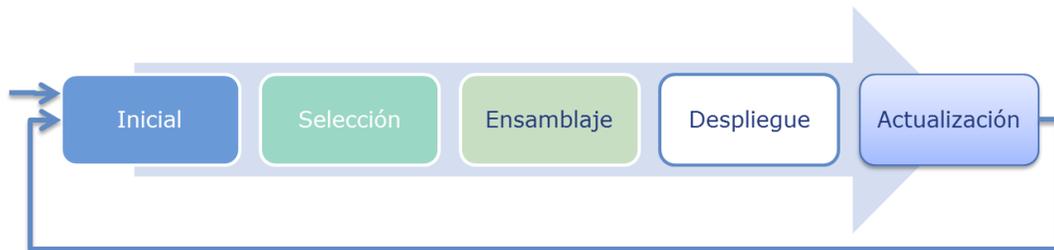


Figura 2.7 Etapas o fases de implementación del modelo.

A continuación se exponen las acciones fundamentales asociadas a cada etapa propuesta:

- **Inicial:** Se identifican las necesidades de la composición, los requerimientos y particularidades de la aplicación que se necesita construir.
- **Selección:** De acuerdo al tipo de aplicación que se necesita construir se sugieren, empleando el componente de selección inteligente de información, las herramientas tecnológicas adecuadas. La toma de decisiones se apoya en una base de conocimientos que constantemente se retroalimenta de los componentes del modelo. Si es necesario se diseñan las interfaces de usuario y la lógica interna de los componentes. También se seleccionan los recursos disponibles en el repositorio de acuerdo a la funcionalidad y a los requisitos no funcionales.
- **Ensamblaje:** Se realizan las interacciones entre los servicios proveedores dentro de las funcionalidades de la aplicación compuesta.
- **Despliegue:** Se despliega la solución de manera que el usuario pueda acceder a los resultados de la composición y realizar pruebas sobre ella.

- **Actualización:** se retroalimenta la base de conocimiento con los casos completados.

Conclusiones del capítulo.

- La adopción del paradigma de composición de aplicaciones en entornos orientados a servicios constituye un elemento de soporte a favor de los importantes esfuerzos de tipo normativo, organizacional y tecnológico del Estado cubano con vistas al mejoramiento de la calidad de los servicios entre los que se incluyen los servicios informáticos y la actualización de la industria cubana de software.
- Para dar respuesta al segundo objetivo específico de la presente investigación, la elaboración del modelo exigió un profundo diagnóstico que reflejó las principales deficiencias e insuficiencias en el desarrollo y composición de aplicaciones dentro de los centros de desarrollo de software en Cuba.
- El modelo que se propone contiene los componentes y relaciones entre los diferentes elementos que conforman el proceso de composición de aplicaciones y establece un conjunto de recomendaciones e indicaciones metodológicas para su implementación como una vía de solución a las principales dificultades detectadas en el diagnóstico. El modelo garantiza la integración de los elementos tecnológicos y cognoscitivos contribuyendo a la adopción de las aplicaciones compuestas en el desarrollo de software.
- La integración al modelo propuesto de un sistema inteligente de selección de información favorece el carácter dinámico del proceso al propiciar elementos de retroalimentación para su perfeccionamiento mediante constantes cambios en los componentes del modelo.

CAPÍTULO III
INSTRUMENTACIÓN Y EVALUACIÓN DEL MODELO

CAPÍTULO III – INSTRUMENTACIÓN Y EVALUACIÓN DEL MODELO

En el capítulo se explica la manera en que fue empleado el modelo en aplicaciones desarrolladas y probadas por el autor para comprobar su factibilidad técnica y evaluar el comportamiento de la interoperabilidad y la estandarización, las variables a cuya mejora se dedicó esta investigación.

3.1 Instrumentación del modelo.

El modelo fue instrumentado con su aplicación total o parcial en las aplicaciones siguientes:

- El sistema SI-Holmes en su versión de aplicación compuesta.
- Herramienta que implementa el escalamiento de Likert y la técnica de Iadov.
- Punto de aplicación de políticas (PEP, Policy Enforcement Point), aplicación para centralizar la gestión de acceso a recursos como parte de una arquitectura XACML (Extensible Access Control Markup Language) [Kanneganti and Chodavarapu 2007].
- Aplicación compuesta para la gestión de tesis.

El proceso de instrumentación se realizó aplicando el ciclo de vida en espiral que plantea la ingeniería de software, con lo cual se fue perfeccionando y depurando la aplicación y extrayendo los elementos necesarios para la experimentación y validación del modelo cuyos cambios tuvieron lugar en una espiral de mayor radio. A continuación se describen algunas de estas aplicaciones basadas en MIDAC, que evalúan las variables interoperabilidad y estandarización.

3.1.1 Versión del sistema Inteligente SI-Holmes basado en MIDAC.

SI-Holmes en una de sus versiones está confeccionada usando el modelo propuesto en esta investigación. El sistema tiene la ventaja de interactuar al mismo tiempo con múltiples bases de

casos y con la información incluida que pueden aparecer almacenadas en sistemas gestores de bases de datos, ficheros, documentos Excel y que son consumidas mediante el uso de servicios de datos (facilitado por las herramientas de la infraestructura tecnológica), brindándole a la aplicación una mayor flexibilidad a su capa de acceso a datos. Estos servicios fueron compuestos en el desarrollo de otras aplicaciones que necesitaron reutilizar la estructura de estas bases de datos [Febles et al. 2012].

La capa de acceso a datos de esta aplicación, como se observa en la Figura 3.1, utiliza también la tecnología de servicios Web, RESTful, así como los estándares asociados a ellas para lograr interoperabilidad en el proceso de composición [Febles, Estrada, Febles and Marquez 2012].

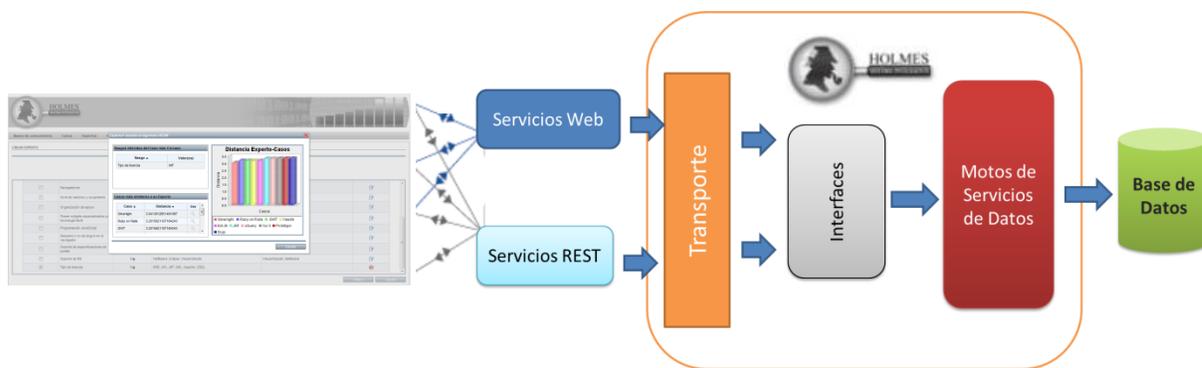


Figura 3.1 Esquema de capa de Acceso a Datos a través de servicios de SI-HOLMES Fuente: Elaboración Propia.

El sistema presenta una interfaz amigable y configurable correspondiente a las aplicaciones RIA que poseen componentes, que brindan un enriquecimiento a las actividades y experiencias del usuario final. El marco de trabajo para construir la interfaz de SI-Holmes (Richfaces combinado con Java Server Faces 2.0) se seleccionó a partir de la interacción de los componentes de Gestión Tecnológica y Selección Inteligente de Información.

SI-Holmes permite crear expertos de cualquiera de sus bases de casos. Para obtener una solución de experto se selecciona el o los rasgos a inferir, se llenan los otros que se consideren necesarios, se escoge alguno de los algoritmos implementados para la distancia entre casos y se obtiene el resultado de la inferencia. En el Anexo 8 se muestran parte de sus interfaces y funcionalidades.

Para presentar los gráficos que permiten mostrar detalladamente la distancia entre los casos con relación a la inferencia realizada por el experto, SI-Holmes utiliza el marco de trabajo Jasper Reports que construye diferentes tipos de gráficas a partir de los datos generales y específicos provenientes de los casos analizados. SI-Holmes está diseñado bajo los principios de interoperabilidad que defiende SOA de manera que sus funcionalidades puedan ser consumidas por otras aplicaciones independientemente de su tecnología.

SI-Holmes comienza a utilizarse como herramienta de apoyo a la toma de decisiones para el manejo de pacientes críticos en unidades de cuidados intensivos (basados en [Rivers et al. 2001]) en el Centro de Investigaciones Médico Quirúrgicas de la Habana (CIMEQ), en la tesis de maestría [Jaquez 2011] y de doctorado [León 2011].

Entre los principales beneficios que se obtienen en la versión de SI-Holmes, como aplicación compuesta, es la posibilidad de incluir las funcionalidades relacionadas con la gestión de bases de casos en los procesos de otras aplicaciones informáticas.

3.1.2 Herramienta para la automatización de la validación.

Basado en MIDAC se desarrolló una aplicación para el apoyo y la automatización de parte de la validación de la presente investigación. Las técnicas de investigación implementadas fueron:

- **Coefficiente de concordancia de Kendall en la gestión de Expertos**

Para fortalecer la calidad de la validación se ejecuta un procedimiento que permite establecer el grado de concordancia de los expertos en relación con todos los aspectos evaluados. Este procedimiento se apoya en el coeficiente de concordancia Kendall (W) (Anexo 9).

- **Escalamiento de Likert y la Técnica de Iadov.**

Para el uso de estas técnicas se desarrollaron dos servicios de datos, uno llamado *IadovDataService* y otro llamado *LikertDataService*. Estos servicios fueron elaborados para insertar, eliminar o actualizar en un sistema gestor de base de datos (PostgreSQL en este caso) el resultado de las encuestas aplicadas para el uso de las técnicas de Iadov y Likert respectivamente. Se implementaron además dos servicios de utilidad: *IadovService* y *LikertService* usando el marco de trabajo Axis2:

- *IadovService*: contiene dos operaciones, una llamada “*obtenerNivelSatisfaccion*” que se encarga de evaluar en una matriz correspondiente con el llamado “Cuadro Lógico de Iadov” previamente creado, las respuestas de las tres preguntas cerradas de la técnica referida y clasificar según su nivel de satisfacción. La otra operación llamada “*obtenerIndiceSatisfaccionGrupal*” calcula el ISG (Índice de Satisfacción Grupal) de acuerdo a las clasificaciones anteriores.
- *LikertService*: posee una operación llamada “*obtenerIndicePorcentual*” que es el resultado del IP (Índice Porcentual) de las encuestas que se encuentren registradas.

Todos estos servicios fueron desarrollados en el WSO2 Developer Studio, basado en Eclipse IDE (Integrated Development Environment). Se diseñaron los contratos de una manera estandarizada antes que la lógica de los servicios, atendiendo a una de las mejores prácticas en el desarrollo de servicios (Contract First) y posteriormente se validaron los contratos como se

muestra en el Anexo 10. Esta buena práctica facilita la independencia de la tecnología subyacente y brinda interoperabilidad al desarrollo de aplicaciones compuestas.

El cálculo del Índice de Satisfacción Grupal se efectuó mediante el lenguaje estándar WS-BPEL, el cual, combinado con un lenguaje de expresiones para procesar documentos XML llamado XPATH, ofreció para esta aplicación la facilidad de orquestar servicios expuestos sin usar código de ningún lenguaje de programación. El ISGProcess fue aplicado en la fase de **Ensamblaje** y su diseño se muestra en la Figura 3.2.

Para elaborar la interfaz de esta aplicación basada en MIDAC, se necesitó un marco de trabajo RIA con soporte de la especificación JSR-168 (especificación que permite desplegar una aplicación web en un portal empresarial en forma de portlet).

En el paso de **Selección** la interacción entre el componente de **Gestión Tecnológica** y el componente de **Selección Inteligente de Información** arrojó el marco de trabajo Vaadin como sugerencia. Este marco de trabajo RIA, usando InviertCharts (add-on para mostrar gráficos interactivos basados en las librerías HighCharts), resultó ser muy conveniente para la visualización de los resultados de la composición.

La aplicación para la automatización de la validación, fue desplegada sobre la infraestructura tecnológica propuesta por MIDAC. En el servidor de aplicaciones se desplegaron los servicios de datos, los servicios de utilidad, el proceso de composición y la aplicación Web para la interacción con el usuario.

Los contratos de los servicios fueron expuestos por el estándar WSDL en sus versiones 1.1 y 2.0. El intercambio de información se basó en el estándar SOAP 1.2 transportado por el protocolo HTTP.

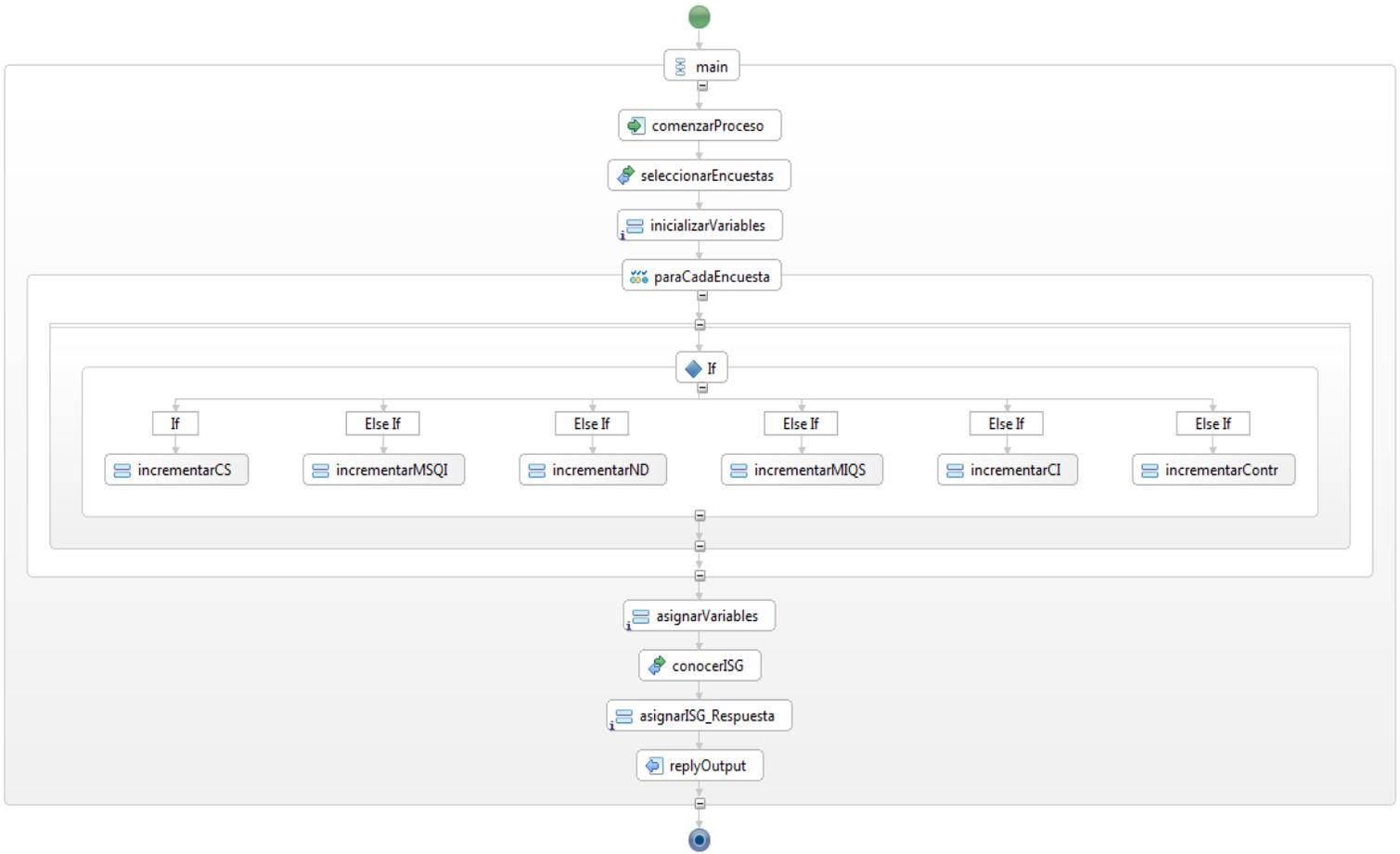


Figura 3.2 Proceso usando el estándar WS-BPEL para calcular el ISG diseñado en WSO2 Developer Fuente: Elaboración Propia.

El Registro/Repositorio gestionó la meta información de los servicios mencionados y permitió su descubrimiento a través del estándar UDDI v3. En el Bus de mensajería se implementaron 4 servicios proxy como una buena práctica para ocultar la ubicación real de los puntos de acceso donde se encuentran los servicios.

3.2 Validación del modelo.

La validación del modelo propuesto se basó en tres pilares fundamentales:

- El estudio de los métodos de validación realizados en las tesis doctorales defendidas en el tribunal de automática y computación en los 5 últimos años.
- Los métodos, procedimientos, técnicas y sugerencias expuestas por doctores en conferencias magistrales, desarrolladas en la UCI y en otros escenarios.
- La literatura referida al tema, especialmente [Shull, Singer and Sjoberg 2008] que aborda específicamente los métodos que pueden ser utilizados para validar investigaciones en el campo de la ingeniería de software.

La validación del modelo incluyó un cuasiexperimento y varias técnicas cualitativas y cuantitativas. Se utilizaron varios profesionales experimentados en el desarrollo de aplicaciones compuestas y en general en la filosofía de la arquitectura orientada a servicios. Las diferentes acciones realizadas se resumen a continuación:

- El escalamiento de Likert para validar los componentes del modelo.
- Un cuasi experimento combinado con el método de comparación para analizar las variables interoperabilidad y estandarización a través de casos prácticos, valorar la facilidad en la composición de aplicaciones y comparar el modelo MIDAC con otros modelos y trabajos que orientan o pretenden orientar la composición de aplicaciones.
- El grupo focal para valorar la integralidad del modelo y para conocer la opinión de expertos sobre la contribución del modelo para facilitar la composición de aplicaciones.
- La técnica de Iadov para evaluar el nivel de satisfacción de usuarios con respecto al modelo propuesto.

Una síntesis de los resultados de cada método o técnica se exponen a continuación.

3.2.1 El escalamiento de Likert

La escala psicométrica creada por Rensis Likert en 1932 (también conocida como método de evaluaciones sumarias) fue aplicada en esta investigación a través de un cuestionario con el objetivo de conocer el nivel de acuerdo o desacuerdo con las características, principios y componentes del modelo MIDAC.

- **Proceso de selección de expertos.**

El proceso de selección de expertos para valorar un tipo de modelo como este, utilizando un proceso de escala, resulta complejo. Para hacerlo se necesitó de especialistas con experiencia en el desarrollo de aplicaciones informáticas y que al mismo tiempo tuviesen conocimientos relacionados con las tecnologías, los estándares, las herramientas, lenguajes y los marcos de trabajo que se aplican en los entornos relacionados con la orientación a servicios.

Se realizó una valoración inicial de los posibles expertos para la validación del modelo. Se contactaron varios profesionales experimentados en la temática con más experiencias en el desarrollo de aplicaciones y servicios, fundamentalmente de Copextel, Softel, de universidades como la UCI, de la maestría de informática de la Universidad “Mariano Gálvez” de Guatemala (específicamente alumnos que combinan los estudios con el trabajo en empresas comercializadoras de software), profesores de la especialidad de Informática de la Universidad Abierta para Adultos (UAPA) de Santiago de los Caballeros, República Dominicana y profesores de la carrera de Ingeniería en Sistemas de la Universidad Central del Este (UCE) de San Pedro de Macorís, República Dominicana.

De una cantidad inicial de 36 expertos posibles, estuvieron dispuestos a participar 27, de los que se desestimaron dos, cuyos índices de competencia fueron bajo el 0,5. De los expertos, 19

obtuvieron un coeficiente de competencia alto y 6 de ellos medio. Seis de los expertos son doctores y quince han cursado maestrías lo que evidencia el alto nivel de los encuestados.

Para demostrar la confiabilidad de las respuestas dadas por los expertos en los cuestionarios, se hizo necesario calcular el coeficiente de concordancia entre ellos. El coeficiente de concordancia de Kendall constituye un método eficiente en estudios de confiabilidad entre expertos de una temática pues está basado en la suposición de que un grupo de expertos pueden llegar mejor a una predicción que una sola persona.

- **Aplicación del escalamiento de Likert**

Para el procesamiento de los resultados se empleó un método que consiste en identificar la frecuencia en cada categoría de la escala de Likert definida en la encuesta (Anexo 11) y se calculan los porcentajes de concordancia de cada categoría de acuerdo a las características propuestas por el autor, luego se calcula en un índice porcentual (IP), que integra en un solo valor la aceptación del grupo de evaluadores sobre las características del modelo.

$$IP = \frac{5(\% \text{ de MA})+4(\% \text{ de A})+3(\% \text{ de Si-No})+2(\% \text{ de ED})+1(\% \text{ de CD})}{5}$$

Dónde: MA-Muy de acuerdo (MA), DA-De acuerdo (A), Si-No- Ni de acuerdo ni en desacuerdo, ED-En desacuerdo (ED), CD-Completamente en desacuerdo

Los planteamientos utilizados fueron los siguientes:

- El proceso de selección tecnológica que forma parte del modelo favorece la correcta composición de aplicaciones.
- El uso en el modelo de inventarios de servicios para la gestión y localización de recursos es fundamental para la composición de aplicaciones.

- El análisis automático de experiencias anteriores como principio de retroalimentación constante es decisivo para el desarrollo de aplicaciones informáticas compuestas.
- El paradigma de razonamiento basado en casos incluido en el modelo es un excelente recurso para seleccionar información de manera inteligente.
- Cubrir las necesidades de aprendizaje que se enuncian en el modelo contribuyen a simplificar la complejidad técnica en el desarrollo de este tipo de aplicaciones.
- La selección correcta de un marco de trabajo RIA, en el componente de gestión tecnológica del modelo, permite a los servicios de presentación enriquecer el valor de las aplicaciones compuestas.
- El chequeo de la aplicación de los principios de diseño de SOA en la Gestión de Recursos es fundamental para la obtención de servicios interoperables, reusables e independientes de la tecnología.
- La concepción de una infraestructura tecnológica correspondiente al entorno cubano realizada para lograr el funcionamiento del modelo es fundamental para el desarrollo de cualquier proyecto informático.
- Los componentes que forman parte del modelo y su interrelación brindan la integralidad necesaria para el desarrollo de aplicaciones compuestas.
- Los principios en los que se basa el modelo facilitan el proceso de composición de aplicaciones en iniciativas orientadas a servicios.

Como se observa en la Figura 3.3 el índice porcentual de los expertos en cada una de las preguntas sobrepasa el valor de 80.

El procesamiento realizado a través del escalamiento de Likert evidencian que tanto los elementos teóricos como las características y las funciones de los componentes del modelo, así

como sus principios, tienen una alta valoración por parte de los expertos, a lo que se unen los criterios emitidos donde se reconoce la importancia del uso de un componente de selección inteligente de información a través del paradigma de razonamiento basado en casos, el cual ofrece una capacidad de retroalimentación constante, necesaria para el apoyo del proceso de composición de aplicaciones.

3.2.2 Validación de satisfacción del usuario.

El conocimiento del estado de satisfacción del usuario respecto al modelo MIDAC, que integra herramientas, buenas prácticas y marcos de trabajo basado en iniciativas orientadas a servicios y que contribuya a facilitar la composición de aplicaciones, es de gran utilidad para la validación de la propuesta.

La Técnica de Iadov constituye una vía para el estudio del grado de satisfacción de los implicados en el proceso objeto de análisis. Para el desarrollo de esta técnica se aplicó una encuesta que permitió conocer el grado de satisfacción con respecto al modelo para la composición de aplicaciones.

La Técnica V. A. Iadov fue creada para establecer el nivel de satisfacción por la profesión de carreras pedagógicas. Luego algunos autores la han modificado en parte y aplicado para valorar la satisfacción en múltiples campos y como parte de diagnósticos y validaciones en diferentes investigaciones. El autor de esta investigación usó esta técnica específicamente para medir la satisfacción de expertos con relación a un modelo para la composición de aplicaciones.

La técnica de Iadov constituye una vía indirecta para el estudio de la satisfacción a la hora de desempeñar ciertas actividades. Se basa en el análisis de un cuestionario que tiene una estructura

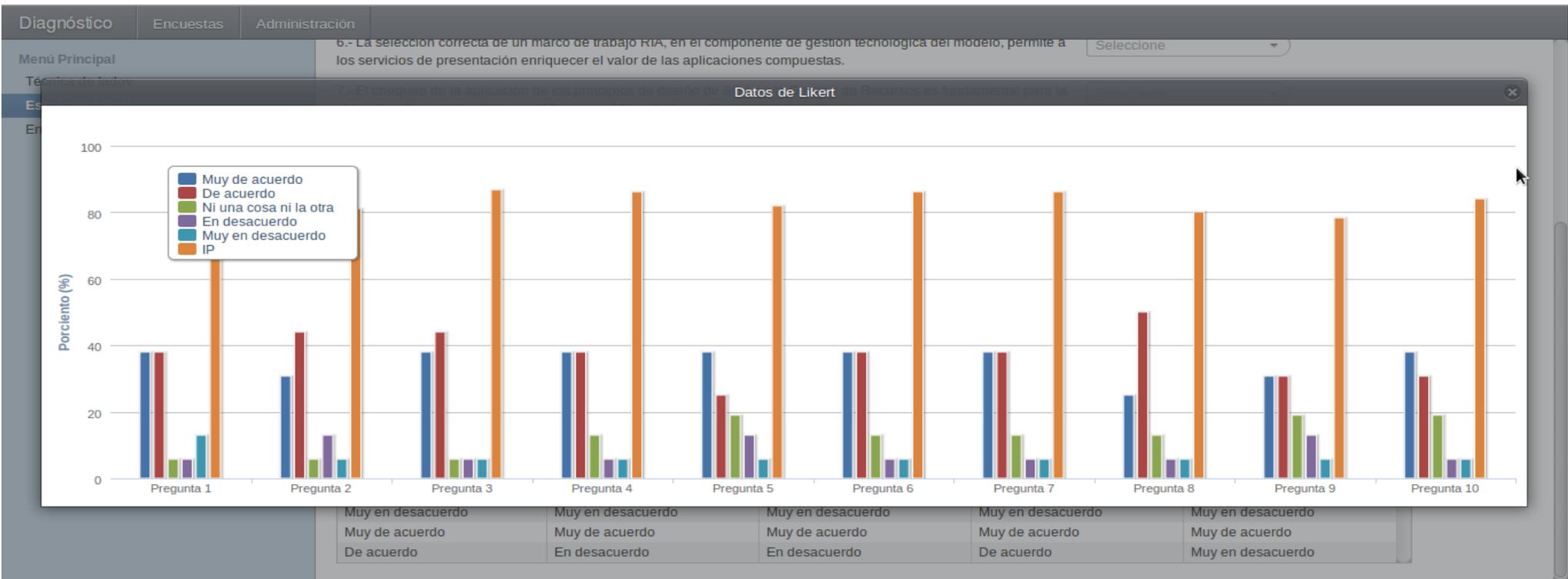


Figura 3.3 Resultados de la aplicación de la escala de Likert. Fuente: Elaboración propia

interna determinada que sigue una relación entre tres cuestionamientos cerrados y el análisis posterior de dos cuestiones abiertas. La relación entre las preguntas cerradas se establece a través del denominado "Cuadro Lógico de Iadov".

¿Le satisface la representación de este modelo para la composición de aplicaciones?	¿Considera usted que se deba desarrollar aplicaciones compuestas sin un modelo coherente?								
	No			No Sé			Sí		
	¿Si usted necesitara desarrollar una aplicación compuesta usaría este modelo?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta Mucho	1	2	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir.	2	3	6	3	3	3	6	3	4

Figura 3.4 Cuadro Lógico de Iadov modificado por el autor.

El número resultante de la interrelación de las tres preguntas indica la posición de cada cual en la escala de satisfacción siguiente:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida

4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

La cual se necesita para calcular el Índice de Satisfacción Grupal (ISG) mediante la fórmula:

$$\text{ISG} = \frac{A (+1) + B (+0,5) + C (0) + D (-0,5) + E (-1)}{N}$$

Que permite reconocer las categorías grupales:

- Insatisfacción: desde (-1) hasta (-0,5)
- Contradictorio: desde (-0,49) hasta (+0,49)
- Satisfacción: desde (+0,5) hasta (1)

Para medir el grado de satisfacción se tomó una muestra de 32 usuarios, teniendo en cuenta los años de experiencia en el desarrollo de aplicaciones y de servicios, entre otros aspectos. Este proceso se llevó a cabo en una aplicación compuesta que se implementó a partir del modelo MIDAC y que sirvió para automatizar y facilitar la validación de esta investigación. El resultado del nivel de satisfacción se muestra en la Figura 3.4.

El *índice grupal* (ISG) arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción. En este caso el valor del ISG fue de 0.703 lo que indica satisfacción con respecto al modelo propuesto.

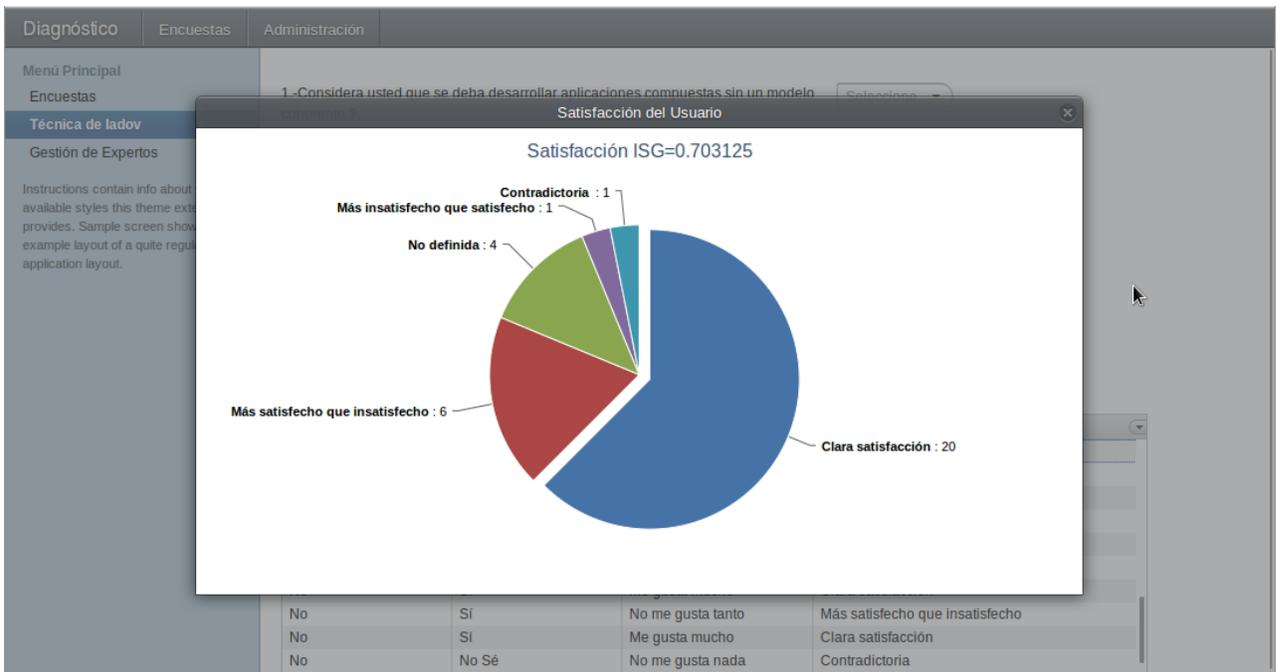


Figura 3.4 Resultados de la aplicación de la técnica de Iadov y valor del índice de satisfacción grupal (ISG) Fuente: Elaboración propia

La técnica de Iadov contempla además dos preguntas complementarias de carácter abierto. Las abiertas son importantes al permitir profundizar en las causas que originan los diferentes niveles de satisfacción y se plantearon sugerencias de utilidad para la presente y futuras investigaciones.

Entre las principales recomendaciones de los expertos estuvieron:

- La posibilidad de medir el tiempo en que los desarrolladores de aplicaciones compuestas podían ahorrar a partir de la sugerencia aportada por el componente inteligente del modelo propuesto.
- El aporte de un mecanismo para la planificación del desarrollo de una aplicación compuesta sin importar su grado de complejidad.

3.2.3 Cuasi experimento para validar el comportamiento de las variables interoperabilidad y estandarización en el modelo propuesto

Se trató, a manera de comparación, de componer aplicaciones utilizando servicios expuestos en <http://uddi.uci.cu> perteneciente a la UCI para componer los servicios que gestionan la identidad, los servicios telemáticos y la información académica. Basado en MIDAC se utilizaron la herramienta SI-Holmes, la aplicación para la automatización de la validación y un componente PEP (Police Enforcement Point) para la gestión de acceso a recursos orientado a servicio.

Las variables que se tuvieron en cuenta para el cuasi experimento se reflejan en la Tabla 1. Por cada uno de estas variables se muestran las características de los recursos que son necesarios para el proceso de composición.

Tabla 1: Variables que se tuvieron en cuenta en el cuasi experimento.

Variables	Aplicaciones desarrolladas sin usar el modelo			Aplicaciones desarrolladas usando el modelo		
	Gestión Académica	Identidad	Información Telemática	SI Holmes	Validación	PEP
Interoperabilidad	Parcial	Parcial	Parcial	Sí	Sí	Sí
Estandarización	Parcial	Parcial	Parcial	Total	Total	Total

El análisis de la interoperabilidad de estos recursos a componer se probó con el consumo de los mismos a través de varias tecnologías. Los tres primeros servicios pudieron ser consumidos solamente con el lenguaje PHP mientras que los servicios expuestos para componer las aplicaciones usando el modelo pudieron ser consumidas con todas variantes que se probaron (la

plataforma Java, la plataforma .Net, el lenguaje PHP, el lenguaje Python y las facilidades para consumir servicios que posee el sistema operativo Adroid, destinado para dispositivos móviles).

Por otro lado, siguiendo los principios del modelo y las buenas prácticas del diseño de servicios, se elaboraron correctamente los contratos, usando estándares relacionados con los servicios Web.

Ninguno de los recursos analizados presentaron problemas con su disponibilidad aunque los pertenecientes al sitio <http://uddi.uci.cu> tienen problemas de localización y descubrimiento.

Los contratos de las aplicaciones desarrolladas usando el modelo exponen correctamente su contrato de servicio en versiones 1.1 y 1.2 de WSDL y diseñados usando las normas y principios de diseño mientras que los 3 primeros servicios poseen elementos en su contrato que no cumplen con los estándares de los servicios Web, afectando su interoperabilidad.

A pesar de que el sitio con dirección <http://uddi.uci.cu> por su nombre se relaciona con el estándar UDDI, este no implementa el estándar con el fin de favorecer el descubrimiento de los servicios mientras el Registro/Repositorio que forma parte de la infraestructura seleccionada para la implementación del modelo utiliza la más reciente versión de este estándar, UDDI v3.

Los marcos de trabajo con que se desarrollaron las aplicaciones basadas en MIDAC son distintos en cada uno de los casos. Esto demuestra estos marcos de trabajo que forman parte del componente de Gestión Tecnológica (tanto marcos de trabajo para el desarrollo de servicios como los utilizados para el desarrollo de interfaces enriquecidas) permiten construir servicios interoperables, componerlos y visualizar los resultados de dicha composición usando los estándares más actuales.

3.2.4 Comparación del modelo MIDAC con otras propuestas anteriores.

La Tabla 2 muestra un análisis comparativo que demuestra el grado de solución que logra el modelo MIDAC con relación a las dificultades principales que actualmente enfrentan los desarrolladores para componer aplicaciones.

Tabla 2 Comparación del modelo MIDAC con otras investigaciones.

Trabajos relacionados con desarrollo de AC.	Independencia tecnológica.	Visualización de la composición.	Facilita Selección Tecnológica.	Posibilidad de almacenar experiencias.	Selección de servicios y uso de repositorios.	Uso de estándares.
[Salter and Jennings 2008]	No	No	No	No	No contempla	Medio
[Mohebi 2012]	Sí	No	No	No	Alto	Alto
[Dannecker, Feldmann, Nestler, Hübsch, Jugel and Muthmann 2010]	No	Sí	No	No	No contempla	Medio
[Bonders and Grabis 2011]	Sí	No	No	No	Alto	Medio
[Angus and Lan 2009]	Sí	No	No	No	Alto	Medio
[Ngu, Carlson and Sheng 2010]	No	Sí	No	No	No contempla	Medio
MIDAC	Sí	Sí	Sí	Sí	Medio	Alto

Como se observa en la tabla anterior MIDAC supera las propuestas anteriores de manera general, aunque queda pendiente el reto de igualar a [Mohebi 2012] y a [Bonders and Grabis 2011] en

cuanto a algoritmos para la selección de servicios almacenados en los Registros/Repositorios. Estas investigaciones mencionadas también tienen un uso más profundo de los elementos pertenecientes a la Calidad de Servicio.

3.2.5 Grupo focal

Para la aplicación de esta técnica se seleccionaron nueve especialistas en temas relacionados con el desarrollo de soluciones en iniciativas orientadas a servicios de la UCI. Dentro de los participantes están investigadores, especialistas del Centro de Desarrollo de Arquitecturas Empresariales (CDAE), doctores en ciencias técnicas, másteres y desarrolladores de servicios que utilizan estándares y tecnologías de referencia vinculadas al paradigma de la orientación a servicios.

Para la conducción de la actividad fue diseñado un guión para conducir el debate (Anexo 13). Para su confección se tuvo en cuenta que la actividad debe comportarse como una entrevista abierta pero estructurada donde se propicie el debate en base a las experiencias personales y al conocimiento que poseen sobre la temática cada uno de los especialistas.

Se seleccionaron además dos personas que no tienen relación directa con la investigación para que participaran como moderador y otro para que llevara la relatoría de una forma imparcial.

Durante el proceso de desarrollo del grupo focal se arrojaron los siguientes resultados:

- Una total coincidencia con respecto al beneficio para el logro de la interoperabilidad que presenta el uso de estándares propuesto en el modelo, lo que beneficia el proceso de composición.
- Hubo concordancia con los componentes del modelo y las herramientas que se seleccionaron para su implementación.

- Todos se pronunciaron positivamente respecto a la contribución del modelo para facilitar la composición de aplicaciones.

Se realizaron algunas recomendaciones que se tuvieron en cuenta en el perfeccionamiento del modelo propuesto en esta investigación. Las más significativas fueron las siguientes:

- Valorar la posibilidad de incorporar el análisis de otro paradigma distinto al de razonamiento basado en casos para el componente de selección inteligente de información.
- Elaborar indicaciones metodológicas que contribuyeran a la aplicación de modelo.
- Incluir en los marcos de trabajo para el desarrollo de servicios, que forman parte del componente Gestión Tecnológica, los frameworks de WSO2 para PHP y C++. Estos frameworks facilitan, a los desarrolladores de PHP y C++ el aprovechamiento de los estándares actuales, así como el despliegue y gestión de servicios en las herramientas propuestas para la infraestructura tecnológica de MIDAC.

3.2.6 Triangulación metodológica

La triangulación metodológica es una técnica usada para tomar múltiples puntos de referencia y localizar una posición desconocida. Es un procedimiento de control implementado para garantizar la confiabilidad en los resultados de cualquier investigación, debido a que disminuye el sesgo que se produce al comparar resultados obtenidos en la cuantificación de variables mediante un método cuantitativo, las tendencias y dimensiones que surgen de la aplicación de métodos cualitativos. Una definición de triangulación metodológica con la que el autor coincide es la ofrecida por Denzin [Sabiote et al. 2006] quién dice que *“es la aplicación y combinación de varias metodologías de la investigación en el estudio de un mismo fenómeno”*.

Para la presente investigación se utiliza la triangulación de métodos, con el objetivo de validar los datos recolectados tanto cualitativos como cuantitativos a través de la aplicación de forma simultánea de los métodos grupo focal, el estudio de casos, la técnica de IADOV y la entrevista a profundidad.

Un resumen de los resultados de la triangulación de métodos se muestra en el en el Anexo 14.

Después de aplicados los métodos descritos se puede concluir que la hipótesis planteada en la investigación fue positiva cumpliéndose los objetivos de desarrollar un modelo que contribuyera a la interoperabilidad y la estandarización en el proceso de composición de aplicaciones.

Desde el punto de vista económico, con la aplicación del modelo propuesto, se logra un impacto importante en cuanto a la productividad del proceso de desarrollo y mantenimiento de aplicaciones compuestas lo que se traduce en ahorro de recursos y reducción de los tiempos de implementación. Además se pueden tomar decisiones más efectivas en cuanto a la selección de las herramientas y marcos de trabajo que deben aplicarse en un proyecto de adopción de SOA en cualquier organización. Según [Greenbaum 2006] por su orientación al reúso de funcionalidades, los proyectos de aplicaciones compuestas basadas en SOA son entre un 13% y un 35% menos costosos que los proyectos tradicionales y los basados en plataformas de integración de aplicaciones (EAI).

La infraestructura seleccionada para soportar tecnológicamente el modelo MIDAC está basada en herramientas de la suite WSO2 que opera bajo licencia Apache v2.0 lo que significa la posibilidad de su uso y modificación sin coste alguno. Los marcos de trabajo tanto para el desarrollo de servicios como para la construcción de interfaces enriquecidas que forman parte de la base de conocimiento también son de código abierto.

Los resultados de esta investigación han sido presentados en Uciencia- 2012 con el trabajo titulado “Interoperabilidad entre sistemas usando aplicaciones compuestas”. Se presentaron también en la Escuela de Verano 2012 donde el autor de la presente investigación impartió el curso “Desarrollo de aplicaciones Web con interfaces enriquecidas usando framework Vaadin” a profesionales del país.

En este curso se hizo particular hincapié en los indicadores para selección de framework RIA y se mostraron las prestaciones del framework Vaadin en las herramientas creadas para automatizar el proceso de validación, al mostrar los beneficios de la composición al usuario final. Parte de estos indicadores están siendo usados por los creadores de Vaadin para comparar su marco de trabajo con otros de su tipo.

Han sido puestos en práctica resultados de esta investigación en la primera fase de un proyecto en conjunto del centro CDAE con el Centro de Inmunología Molecular (CIM) que tienen como objetivo informatizar de una manera homogénea el área de producción y manufactura y donde las aplicaciones compuestas jugarán un papel fundamental.

Conclusiones del capítulo

- Se comprobó la importancia de los componentes elaborados como parte del modelo propuesto así como sus relaciones y funcionalidades.
- Se consideran fundamentales los resultados obtenidos a partir del desarrollo de aplicaciones basadas en los principios del modelo que demostraron la factibilidad de su uso durante la ejecución de los métodos de validación en esta investigación.

- Los resultados obtenidos referentes a la infraestructura tecnológica propuesta refleja la capacidad de las herramientas seleccionadas para soportar funcionalmente el modelo propuesto.

CONCLUSIONES GENERALES

1. Con los elementos teóricos y prácticos más actuales de las ciencias informáticas en lo referido a las aplicaciones compuestas se obtuvo un modelo integrado que fue utilizado para la construcción de aplicaciones informáticas con resultados satisfactorios y que fue validado con métodos científicos. El modelo garantiza que, mediante la integración de herramientas, buenas prácticas y marcos de trabajo para el desarrollo de aplicaciones compuestas en iniciativas orientadas a servicios, se puede ganar en la interoperabilidad y la estandarización del proceso.
2. Se comprobó que es posible incorporar la experiencia obtenida en aplicaciones informáticas desarrolladas con anterioridad si se emplean preceptos del razonamiento basado en casos y se introducen como parte de un modelo integral. Para ello se diseñó e implementó una aplicación basada en el modelo propuesto que permite gestionar conocimiento y apoyar la toma de decisiones. De esa manera se le brinda a las organizaciones un instrumento para el registro de las mejores prácticas y el conocimiento en la gestión y desarrollo de software.
3. En la infraestructura propuesta se incluyen herramientas que fueron seleccionadas para soportar tecnológicamente y garantizar el funcionamiento correcto del modelo. Los elementos pertenecientes a esta infraestructura han sido probados en el entorno nacional e internacional.
4. El conjunto de métodos científicos utilizados para la validación del modelo (Grupo focal, Escalamiento de Likert, Técnica de Iadov, Triangulación Metodológica) permitió comprobar:
 - La utilización de constructos que responden a la teoría más actual de las aplicaciones compuestas, están correctamente utilizados y se ordenan de forma precisa.
 - Las características, la estructura y los componentes del modelo fueron aceptadas y utilizadas con relativa facilidad por desarrolladores cubanos y de otros países.

- Existe una alta satisfacción de los usuarios actuales y potenciales con respecto a la necesidad, utilidad y actualidad del modelo propuesto.

RECOMENDACIONES

- Continuar las investigaciones que permitan elaborar una estrategia de seguridad para el modelo que incluya los componentes de Gestión Tecnológica y Gestión de Recursos. Esta estrategia debe incorporar en las indicaciones de salida del modelo, las medidas necesarias para asegurar el proceso de composición.
- Valorar la incorporación de nuevos algoritmos a la herramienta desarrollada en el componente de selección inteligente de información y que fue muy útil en el uso del modelo. Se deben implementar las posibilidades de conexión entre bases de casos diferentes para enriquecer la toma de decisiones en diferentes dominios.
- Continuar la generalización del modelo con su aplicación en otros entornos.
- Continuar enriqueciendo los Registros/Repositorios incorporando una revisión previa a los servicios para evitar ineficiencias en el proceso de composición.

REFERENCIAS BIBLIOGRÁFICAS

1. ABOULSAMH, M.A. 2009. Towards a model-driven approach for planning a standard-based migration of enterprise applications to SOA In *Proceedings of the Congress on Services - I* 2009.
2. AL-JAROODI, J., MOHAMED, N. AND AZIZ, J. 2010. Service Oriented Middleware: Trends and Challenges. In *Proceedings of the 2010 Seventh International Conference on Information Technology* 2010.
3. ALAHMARI, S., ZALUSKA, E. AND ROURE, D.D. 2010. A Service Identification Framework for Legacy System Migration into SOA In *Proceedings of the IEEE International Conference on Services Computing* 2010.
4. ANGUS, H. AND LAN, C.-W. An optimal QoS-based Web service selection scheme. *Information Sciences: an International Journal* archive, 2009, vol. 179.
5. BALANI, N. AND HATHI, R. *Apache CXF Web Service Development*. Edtion ed.: Packt Publishing, 2009. ISBN 978-1-847195-40-1.
6. BANERJEE, A. What Are Composite Applications? *The Architecture Journal*, 2006, vol. 10.
7. BAUN, C., KUNZE, M., NIMIS, J. AND TAI, S. *Cloud Computing: Web-Based Dynamic IT Services*. Edtion ed.: Springer, 2011. ISBN 978-3-642-20916-1.
8. BELLO, R. *Aplicaciones de la Inteligencia Artificial*. Edtion ed. Guadalajara, Jalisco, México, 2002. ISBN 970-27-0177-5.
9. BERNAL, J. *Web 2.0 and Social Networking for the Enterprise*. edited by K. CLINE. Edtion ed.: IBM Press, 2009. ISBN 978-0-13-700489-8.
10. BERTOLINO, A. AND POLINI, A. 2009. SOA Test Governance: enabling service integration testing across organization and technology borders. In *Proceedings of the IEEE International Conference on Software Testing Verification and Validation Workshops* 2009.
11. BHARADWAJ, N. *BPEL PM and OSB Operational Management with Oracle Enterprise Manager 10g Grid Control*. Edtion ed.: Packt Publishing, 2010. ISBN 978-1-847197-74-0.

12. BIEBERSTEIN, N., BOSE, S., FIAMMANTE, M., JONES, K. AND SHAH, R. *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*. Edition ed.: IBM Press, 2005. ISBN 0-13-187002-5.
13. BLANVALET, S., BOLIE, J., CARDELLA, M., CAREY, S., CHANDRAN, P., COENE, Y., GEMINIUC, K., MATJAŽ B. JURIČ, NGUYEN, T.H., PODUVAL, A., PRAVIN, L., THOMAS, J. AND TODD, D. *Best Practices for SOA-based integration and composite applications development*. Edition ed.: Packt Publishing, 2006. ISBN 1-904811-33-7.
14. BOLIE, J., CARDELLA, M., CAREY, S., CHANDRAN, P., COENE, Y., KEVIN GEMINIUC, MATJAŽ B. JURIČ, NGUYEN, T.H., PODUVAL, A., PRAVIN, L., THOMAS, J. AND TODD, D. *BPEL Cookbook Best Practices for SOA-based integration and composite applications development*. Edition ed. Birmingham: Packt Publishing, 2006. ISBN 978-1-847197-94-8.
15. BONDERS, M. AND GRABIS, J. A Holistic Method for Selecting Web Services in Design of Composite Applications. Institute of Information Technology, Riga Technical University, 2011.
16. BURKE, B. *RESTful Java with JAX-RS*. edited by M. LOUKIDES. Edition ed.: O'Reilly Media, 2010. ISBN 978-0-596-15804-0.
17. CABRERA, O., ORIOL, M., LÓPEZ, L., FRANCH, X., MARCO, J., FRAGOSO, O. AND SANTAOLAYA, R. WeSSQoS: Un Sistema SOA para la Selección de Servicios Web según su Calidad. 2010.
18. COFFEY, J., WHITE, L. AND WILDE, N. 2010. Locating Software Features in a SOA Composite Application. In *Proceedings of the Eighth IEEE European Conference on Web Services 2010*.
19. CRUPI, J. AND WARNER, C. Enterprise Mashups: Bringing SOA to the People. The SOA Magazine, 2008, no. XVIII.
20. CHEN, R., SHEN, V., WROBEL, T. AND LIN, C. 2008. Applying SOA and Web 2.0 to Telecom: Legacy and IMS Next-Generation Architectures. In *Proceedings of the IEEE International Conference on e-Business Engineering 2008*.

21. DAHMAN, K., CHAROY, F. AND GODART, C. 2010. Generation of Component Based Architecture from Business Processes: Model Driven Engineering for SOA In *Proceedings of the Eighth IEEE European Conference on Web Services* 2010.
22. DAIGNEAU, R. *Service Design Patterns*. Edtion ed.: Addison-Wesley, 2011. ISBN 978-0-321-54420-9.
23. DANNECKER, L., FELDMANN, M., NESTLER, T., HÜBSCH, G., JUGEL, U. AND MUTHMANN, K. Rapid Development of Composite Applications Using Annotated Web Services. 2010.
24. DAVIS, J. *Open Source SOA*. Edtion ed. Greenwich: Manning Publications, 2009. ISBN 978-1-933988-54-2.
25. DELGADO. Metodología para desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture). In *XXXII Conferencia Latinoamericana de Informática (CLEI'06), Sesión 4*. Santiago de Chile, 2006.
26. DEXTER, H., PETCH, J. AND POWLEY, D. Establishing an SOA composite applications development process for work- based learning and competency progression management. 2012.
27. DIAS, J.J.L., ALMEIDA, E.S.D. AND MEIRA, S.R.D.L. 2010. The Analysis Activity in a Systematic SOA-based Architecture Process. In *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference Workshops* 2010.
28. DIRKSEN, J. *SOA Governance in Action*. Edtion ed.: Manning Publications, 2012. ISBN 9781617290275.
29. DUGGAN, D. *Enterprise Software Architecture and Design. Entities, Services, and Resources*. Edtion ed.: John Wiley & Sons, Inc, 2012. ISBN 978-0-470-56545-2.
30. DUNLAP, T. IBM's New Portal Server Embraces "Composite Applications". *Intranet Journal*, 2006.
31. ERL, T. *Service-Oriented Architecture: Concepts, Technology, and Design*. Edtion ed.: Prentice Hall PTR, 2005. ISBN 0-13-185858-0.

32. ERL, T. *SOA Principles of Service Design* Edtion ed.: PRENTICE HALL, 2007. ISBN 9780132344821.
33. ERL, T. *SOA Design Patterns*. Edtion ed.: PRENTICE HALL, 2008. 856 p. ISBN 0-13-613516-1.
34. ERL, T., BENNETT, S.G., CLIVE GEE, R.L., MANES, A.T., SCHNEIDER, R., SHUSTER, L., TOST, A. AND VENABLE, C. *SOA Governance*. Edtion ed.: PRENTICE HALL, 2011. ISBN 978-0-13-815675-6.
35. ERL, T., KARMARKAR, A., PRISCILLA WALMSLEY, HUGO HAAS, UMIT YALCINALP, CANYANG KEVIN LIU, DAVID ORCHARD, ANDRE TOST AND PASLEY, J. *Web Service Contract Design and Versioning for SOA*. Edtion ed.: PRENTICE HALL, 2010. ISBN 978-0-13-613517-3.
36. FEBLES, O. Sistema de Razonamiento Basado en Casos. SI-Holmes. UAPA Universidad Abierta para Adultos, ISSN: 84-95193-08-6, 2011.
37. FEBLES, O., CRUZ, M.R. AND RODRÍGUEZ, Y.G. Interoperabilidad entre sistemas usando aplicaciones compuestas. In *Uciencia*. Artículo en memoria de evento, 2012.
38. FEBLES, O., ESTRADA, V., FEBLES, J.P. AND MARQUEZ, I. Aplicación compuesta para la gestión de sistemas basados en conocimiento. ACIMED, 2012.
39. FEBLES, O., PUPO, C., ESTRADA, V. AND FEBLES, J.P. Los Mashups: Aplicaciones compuestas de la Web 2.0. Revista Ciencias de la Informacion., 2012.
40. FEBLES, O. AND SANTOS, C. Propuesta de Infraestructura Tecnológica para el desarrollo y despliegue de una iniciativa con Arquitectura Orientada a Servicios 2010.
41. FERGUSON, D., PILARINOS, D. AND SHEWCHUK, J. Bus de Servicios de Internet. The Architecture Journal, 2007, vol. 13.
42. FIELDING, R. Architectural Styles and the Design of Network-based Software Architectures. In. California: University of California, 2000, vol. Phd Thesis.
43. FISHER, M., PARTNER, J., BOGOEVICI, M. AND FULD, I. *Spring Integration in Action*. Edtion ed.: Manning Publications Co., 2012. ISBN 9781935182436.

44. GARRETT, J.-J. *Ajax: A New Approach to Web Applications*. Edtion ed., 2005.
45. GOVERNOR, J., HINCHCLIFFE, D. AND NICKULL, D. *Web 2.0 Architectures*. edited by O'REILLY. Edtion ed., 2009. ISBN 978-0-596-51443-3.
46. GRAHAM, I. *Requirements Modelling and Specification for Service Oriented Architecture*. Edtion ed.: Wiley, 2008. ISBN 978-0-470-77563-9.
47. GREENBAUM, J. Return on Investment for Composite Applications and Service Oriented Architectures: A Model for Financial Success and Enterprise Efficiency. In.: *Enterprise Applications Consulting*, 2006.
48. GRONOSKY, A., ATIGHETCHI, M. AND PAL, P. 2010. Understanding the Vulnerabilities of a SOA Platform. In *Proceedings of the Ninth IEEE International Symposium on Network Computing and Applications 2010*.
49. GU, C. AND ZHANG, X. 2010. An SOA Based Enterprise Application Integration Approach. In *Proceedings of the Third International Symposium on Electronic Commerce and Security 2010*.
50. GUELMES, Y. Proceso de diseño de servicios para proyectos SOA. In. Habana: UCI, 2010, vol. Tesis para optar por el grado de Máster en Ciencias.
51. HAINES, M.N. AND HASEMAN, W. 2009. Service-Oriented Architecture Adoption Patterns. In *Proceedings of the 42nd Hawaii International Conference on System Sciences*. 2009.
52. HIRZALLA, M. 2010. Realizing Business Agility Requirements through SOA and Cloud Computing. In *Proceedings of the 18th IEEE International Requirements Engineering Conference 2010*.
53. HOLLEY, K. AND ARSANJANI, D.A. *100 SOA Questions: Asked and Answered*. Edtion ed.: Prentice Hall, 2010. ISBN 978-0-137-08020-5.
54. HONGHUI, N. AND XIAOJUN, M. 2010. Research of E-Learning System Based on SOA. In *Proceedings of the 2010 Second International Conference on MultiMedia and Information Technology 2010*.
55. MANERO, B AND FERNANDEZ, B, IEEE Definición de interoperabilidad 1990.

56. INFANTE, J. UDDI o no UDDI. [online] In. <http://blogs.prod.uci.cu/desarrolloSOA/2010/03/uddi-o-no-uddi/> visitado:[Consultado marzo 2012] , 2012.
57. JAQUEZ, J. Diseño e implementación de una base de conocimiento de apoyo a la toma de decisiones en el juicio penal de República Dominicana., 2011, Tesis para optar por el grado de Máster en Gestión de Información. Universidad de La Habana.
58. JEFF LINWOOD, D.M. *Building Portals with the Java Portlet API*. Edtion ed., 2004. ISBN 1-59059-284-0.
59. JIAN, J., XI, H.-F. AND ZHANG, L. 2010. The design for General-purpose Interface of Community Informatization Basic Database Based on SOA. In *Proceedings of the International Conference on E-Business and E-Government 2010*.
60. JONAS X. YUAN, X.C., FRANK YU *Liferay User Interface Development*. Edtion ed., 2010. ISBN 978-1-849512-62-6.
61. JOSUTTIS, N.M. *SOA in Practice*. Edtion ed., 2007. ISBN 10: 0-596-52955-4.
62. JURIC, M.B. *Business Process Execution Language for Web Services Second Edition*. Edtion ed., 2006.
63. JURIC, M.B. *WS-BPEL 2.0 for SOA Composite Applications*. 2010. ISSN 978-1-847197-94-8.
64. JURIC, M.B. AND KRIZEVNIK, M. *WS-BPEL 2.0 for SOA Composite Applications with Oracle SOA Suite 11g*. Edtion ed.: Packt Publishing, 2010. ISBN 978-1-847197-94-8.
65. KANNEGANTI, R. AND CHODAVARAPU, P. *SOA Security*. Edtion ed.: Manning, 2007. ISBN 1-932394-68-0.
66. KAZEM, M. AND FORTE, M.W. 2010. Service Oriented Enterprise Architecture Framework. In *Proceedings of the IEEE 6th World Congress on Services 2010*.
67. KEYSER, C. Composite Applications .The new paradigm. *The Architecture Journal*, 2007, vol. 10.
68. KISTASAMY, C., MERWE, A.V.D. AND HARPE, A.D.L. 2010. The Relationship between Service Oriented Architecture and Enterprise Architecture. In *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference Workshops 2010*.

69. KRAFZIG, D., BANKE, K. AND SLAMA *Service Oriented Architecture Best Practices*. Edtion ed., 2005. ISBN 0-13-146575-9.
70. KUMAR, B.V., NARAYAN, P. AND NG, T. *Implementing SOA Using Java™ EE*. Edtion ed.: Addison Wesley, 2009. ISBN 978-0-321-49215-9.
71. LAWS, S. AND COMBELLACK, M. *Tuscany SCA in Action*. Edtion ed.: Manning, 2011. ISBN 9781933988894.
72. LEÓN, G. Modelo de Gestión del conocimiento para las áreas económicas del sistema de instituciones del Ministerio de Educación Superior. In., 2011, vol. Tesis para optar por el grado de Doctor en Ciencias de la Educación. Universidad de La Habana. Cuba.
73. LEÓN, G., FEBLES, O., ESTRADA, V. AND FEBLES, J.P. El razonamiento basado en casos y la determinación del grado de introducción de la gestión de la información y el conocimiento económico en la gerencia de la universidad cubana. *Pedagogía Universitaria*, 2010, vol. 15.
74. LEÓN, I.Y.G. Proceso de diseño de servicios para proyectos SOA. In. La Habana: Universidad Ciencias Informáticas, 2010.
75. LEÓN, Y.G., VALDIVIA, M.M., ZALDÍVAR, Y.R., SANTIAGO, Y.V. AND OSORIO, J.I. *Perspectiva de Diseño para la Arquitectura de Referencia SOA del CECTIS*. 2010.
76. LI, B. 2010. Research on SOA and Component Oriented Technology in Development of Large System. In *Proceedings of the International Symposium on Computational Intelligence and Design* 2010.
77. LI, Y. AND XIONG, Q. 2010. Enterprise Application Rebuilding Framework based on Semantic SOA and Workflow. In *Proceedings of the Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science* 2010.
78. LYNN, R., BISHOP, K. AND KING, B. *Getting Started with IBM WebSphere sMash*. edited by I. PRESS. Edtion ed., 2010. ISBN 978-0-13-701970-0.
79. MAHEMOFF, M. *Ajax Design Patterns*. Edtion ed.: O Reilly, 2006.

80. MARGOLIS, B. AND SHARPE, J. *SOA for the Business Developer: Concepts, BPEL, and SCA*. Edition ed.: MC Press 2007. ISBN 9781583470657.
81. MATSUMARA, M., BRAUEL, B. AND SHAH, J. *SOA Adoption For Dummies*. Edition ed.: Wiley Publishing, 2009. ISBN 978-0-470-48334-3.
82. MICROSOFT. La Arquitectura Orientada a Servicios (SOA). 2006. Available from Internet:<http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf> [Consultado junio 2011].
83. MOHEBI, A. An Efficient Qos-Based Ranking Model for Web Service Selection with Consideration of User's Requirement. In *Applied Mathematics*. Toronto, Canada: Ryerson University, 2012, vol. Master of Science in the program of Computer Science.
84. NEIL, T. Top Ajax Technologies and RIA Frameworks. In., [Consultado febrero 2011], 2011.
85. NGU, A.H.H., CARLSON, M.P. AND SHENG, Q.Z. Semantic-Based Mashup of Composite Applications. *IEEE Transactions on services computing*, 2010, vol. 3.
86. OASIS. Reference Model for Service Oriented Architecture. 2006, [Consultado 23 octubre 2011]. Available from Internet:<<http://www.oasis-open.org/committees/download.php/28134/wd-soa-rm-cd1ED.pdf..>>.
87. OASIS Reference Architecture for Service Oriented Architecture 2008.
88. OBEIDAT, R. AND ZAATREH, Z. 2010. Motivation for Integrating Supply Chains using Service Oriented Architecture Approach. In *Proceedings of the Fourth International Conference on Next Generation Mobile Applications, Services and Technologies* 2010.
89. OGRINZ, M. *Mashup Patterns. Designs and Examples for the Modern Enterprise*. Edition ed. Massachusetts Addison Wesley, 2009. ISBN 978- 0-321-57947-8.
90. OLAF ZIMMERMANN, P.K., CLIVE GEE Elements of Service-Oriented Analysis an Design, an interdisciplinary modelling approach for SOA Projects 2004.
91. ORACLE. Gestión de aplicaciones compuestas: cómo salvar la brecha de visibilidad de IT en aplicaciones compuestas complejas 2008.

92. PANDA, D. AND MAHESHWARI, A. *Middleware Management with Oracle Enterprise Manager Grid Control 10g R5*. Edition ed.: Packt Publishing, 2009. ISBN 978-1-847198-34-1.
93. PAPAGEORGIOU, A., KROP, T., AHLFELD, S., SCHULTE, S., ECKERT, J. AND STEINMETZ, R. 2010. Enhancing Availability with Self-Organization Extensions in a SOA Platform. In *Proceedings of the Fifth International Conference on Internet and Web Applications and Services 2010*.
94. PAUL J. DEITEL, H.M.D. *AJAX, Rich Internet Applications, and Web Development For Programmers*. Edition ed. Prentice Hall, 2008. ISBN 0-13-158738-2.
95. POPADIYN, P. Patterns and practices in the Composite Application Library 2009, vol. 1.
96. POSTINA, M., TREFKE, J. AND STEFFENS, U. 2010. An EA-Approach to Develop SOA Viewpoints. In *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference 2010*.
97. QH, H., XZ, L. AND ZF, T. 2010. A new Customer Care Information System Architecture Model integrated EA with SOA in Power Supply Enterprise. In *Proceedings of the International Conference on E-Business and E-Government 2010*.
98. QI-WEN, Z. AND QI, X. 2010. Research on Heterogeneous System Integration of Supply Chain Based on SOA. In *Proceedings of the International Conference on E-Business and E-Government 2010*.
99. RAJANI, V., KUMAR, A. AND JANAKIRAM, D. 2010. A Calculus for Service Interactions. In *Proceedings of the IEEE International Conference on Services Computing 2010*.
100. RIVERS, E., NGUYEN, B., HAVSTAD, S., RESSLER, J., MUZZIN, A., KNOBLICH, B., PETERSON, E. AND TOMLANOVICH, M. Early Goal-Directed Therapy in the Treatment of Severe Sepsis and Septic Shock. *The New England Journal of Medicine*, 2001, vol. 61.
101. ROSEN, M. *Applied SOA: Service-Oriented Architecture and Design Strategies*. 2008.
102. ROSEN, M., LUBLINSKY, B., SMITH, K.T. AND BALCER, M.J. *Applied SOA. Service-Oriented Architecture and Design Strategies* Edition ed.: Wiley Publishing, 2008. ISBN 978-0-470-22365-9.

103. ROTEM-GAL-OZ, A. *SOA Patterns*. Edtion ed.: Manning, 2012. ISBN 9781933988269.
104. SABIOTE, C.R., LLORENTE, T.P. AND PÉREZ, J.G. La triangulación analítica como recurso para la validación de estudios de encuestas recurrentese investigaciones de réplica en Educación Superior. RELIEVE, 2006, vol. 12.
105. SALTER, D. AND JENNINGS, F. *Building SOA-Based Composite Applications Using NetBeans IDE* 6. Edtion ed. Birmingham, : Packt Publishing, 2008. ISBN 978-1-847192-62-2.
106. SANGROYA, A., GARG, K. AND VARMA, V. 2010. SAGE: An Approach to Evaluate the Impact of SOA Governance Policies. In *Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications Workshops* 2010.
107. SARANG, P. *Practical Liferay Java-based Portal Applications Development*. Edtion ed., 2009. ISBN 978-1-4302-1848-7.
108. SARIN, A. *Portlets in Action*. Edtion ed.: Manning Early Access Program, 2010.
109. SATTARI, H. AND KUNJUMOHAMED, S. *Spring Web Services 2 Cookbook*. edited by C. RODRIGUEZ. Edtion ed.: Packt Publishing, 2012. ISBN 978-1-84951-582-5.
110. SEGARAN, T., EVANS, C. AND TAYLOR, J. *Programming the Semantic Web*. Edtion ed.: O'Reilly Media, 2009. ISBN 978-0-596-15381-6.
111. SEZOV, R. *Liferay in Action*. Edtion ed.: Manning Publications, 2012. ISBN 9781935182825.
112. SHULL, F., SINGER, J. AND SJOBERG, D.I.K. *Guide to Advanced Empirical Software Engineering*. Edtion ed.: Springer-Verlag, 2008. 394 p. ISBN 978-1-84800-043-8.
113. SILVA-LEPE, I., ROUVELLOU, I., AKOLKAR, R. AND IYENGAR, A. 2010. Seamless Cross-Domain Connectivity for Enabling Domain Autonomy in a Federated SOA In *Proceedings of the IEEE International Conference on Web Services* 2010.
114. SOPEÑA, J.G., CAMINO, F.J.S. AND NAVARRO, J.J.M. Informe de Vigilancia Tecnológica 14 (VT14). Tecnologías software orientadas a servicios. 2008.

115. SOUSA, F.M.L.E. AND NETO, V.S.C. 2012. Conceptual Elaboration of a Geographic Intelligence Based on Service Oriented Architecture (SOA). In *Proceedings of the Seventh International Conference on Information Technology 2012*.
116. SPOTT, D. The CBDI-SAE Reference Framework. *CBDI Journal*, 2010.
117. SRIVASTAVA, B. AND MAZZOLENI, P. Business Driven Consolidation of SOA Implementations. In *IEEE International Conference on Services Computing*. 2010.
118. TANG, L., DONG, J., ZHAO, Y. AND TSAI, W.-T. 2010. A Classification of Enterprise Service-Oriented Architecture. In *Proceedings of the Fifth IEEE International Symposium on Service Oriented System Engineering 2010*.
119. THURAISSINGHAM, B. *Secure Semantic Service-Oriented Systems*. Edition ed.: Auerbach Publications, 2011. ISBN 978-1-4200-7332-4.
120. TOMCZAK, J.M., CIEŚLIŃSKA, K. AND PLESZKUN, M. Development of Service Composition by Applying ICT Service Mapping. *Computer Networks*, 2012.
121. TONG, K.K.I. *Developing Web Services with Apache Axis2*. Edition ed.: TipTec Development, 2008. ISBN 978-99937-929-1-8.
122. TONG, K.K.I. *Developing Web Services with Apache CXF and Axis2*. Edition ed.: TipTec Development, 2010. ISBN 978-99937-929-1-8.
123. VOELZ, D. AND GOEB, A. 2010 What is Different in Quality Management for SOA? In *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference 2010*
124. VOHRA, D. *Java 7 JAX-WS Web Services*. Edition ed.: Packt Publishing, 2012. ISBN 978-1-84968-720-1.
125. WEBBER, J., PARASTATIDIS, S. AND ROBINSON, I. *REST in Practice*. edited by S. ST.LAURENT. Edition ed.: O'Reilly Media, 2010. ISBN 978-0-596-80582-1.
126. YAN, P. AND GUO, J. 2010. Researching and Designing the Architecture of E-government based on SOA. In *Proceedings of the International Conference on E-Business and E-Government 2010*.

127. YEE, R. *Pro Web 2.0 Mashups: Remixing Data and Web Services*. Edition ed.: Apress, 2008. ISBN 978-1-4302-0286-8.
128. YOUNG, M. *Google Maps Mashups with Google Mapplets*. Edition ed.: Apress, 2009.
129. YUAN, J.X. *Liferay Portal 5.2 Systems Development*. Edition ed.: Packt Publishing Ltd, 2009. ISBN 978-1-847194-70-1.
130. YUNLIANG, J., XIONGTAO, Z., QING, S., JING, F. AND NING, Z. 2010. Design of E-Government Information Management Platform Based on SOA Framework. In *Proceedings of the 2010 First International Conference on Networking and Distributed Computing 2010*.
131. ZENDER, R., LUCKE, U. AND TAVANGARIAN, D. 2010. SOA Interoperability for Large-scale Pervasive Environments. In *Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications Workshops 2010*.
132. ZHANG, C., CHEN, H. AND DU, J. 2011. A Tabu Search Approach for Dynamic Service Substitution in SOA Applications. In *Proceedings of the IEEE Asia -Pacific Services Computing Conference. 2011*.
133. ZHANG, L.-J. AND ZHANG, J. 2009. Componentization of Business Process Layer in the SOA Reference Architecture. In *Proceedings of the 2009 IEEE International Conference on Services Computing 2009*.
134. ZHOU, N., ZHANG, L.-J., CHEE, Y.-M. AND CHEN, L. 2010. Legacy Asset Analysis and Integration in Model-Driven SOA Solution. In *Proceedings of the IEEE International Conference on Services Computing 2010*.

ANEXOS

ANEXO 1. Encuesta para determinar el grado de conocimiento de las herramientas y tecnologías utilizadas para desarrollar aplicaciones compuestas

El 90% de los encuestados conoce los marcos de trabajo RIA y de ellos el 70% cree que resulta muy importante su uso para el desarrollo de la interfaz de aplicaciones informáticas. El 60% conoce del paradigma de aplicaciones compuestas aunque solo el 20% ha participado en el desarrollo de alguna. El 40% de los encuestados no tienen conocimiento alguno sobre herramientas para desarrollo de aplicaciones compuestas. El 50% de los encuestados cree que el uso de estándares es definitorio para el logro de la interoperabilidad en el desarrollo de las aplicaciones compuestas. El 60% considera que las aplicaciones compuestas tendrán gran impacto en el desarrollo de proyectos clásicos y orientados a servicios, así como la informatización de las organizaciones. El 70% estima que la selección de herramientas es de vital importancia para el desarrollo de este tipo de aplicaciones.

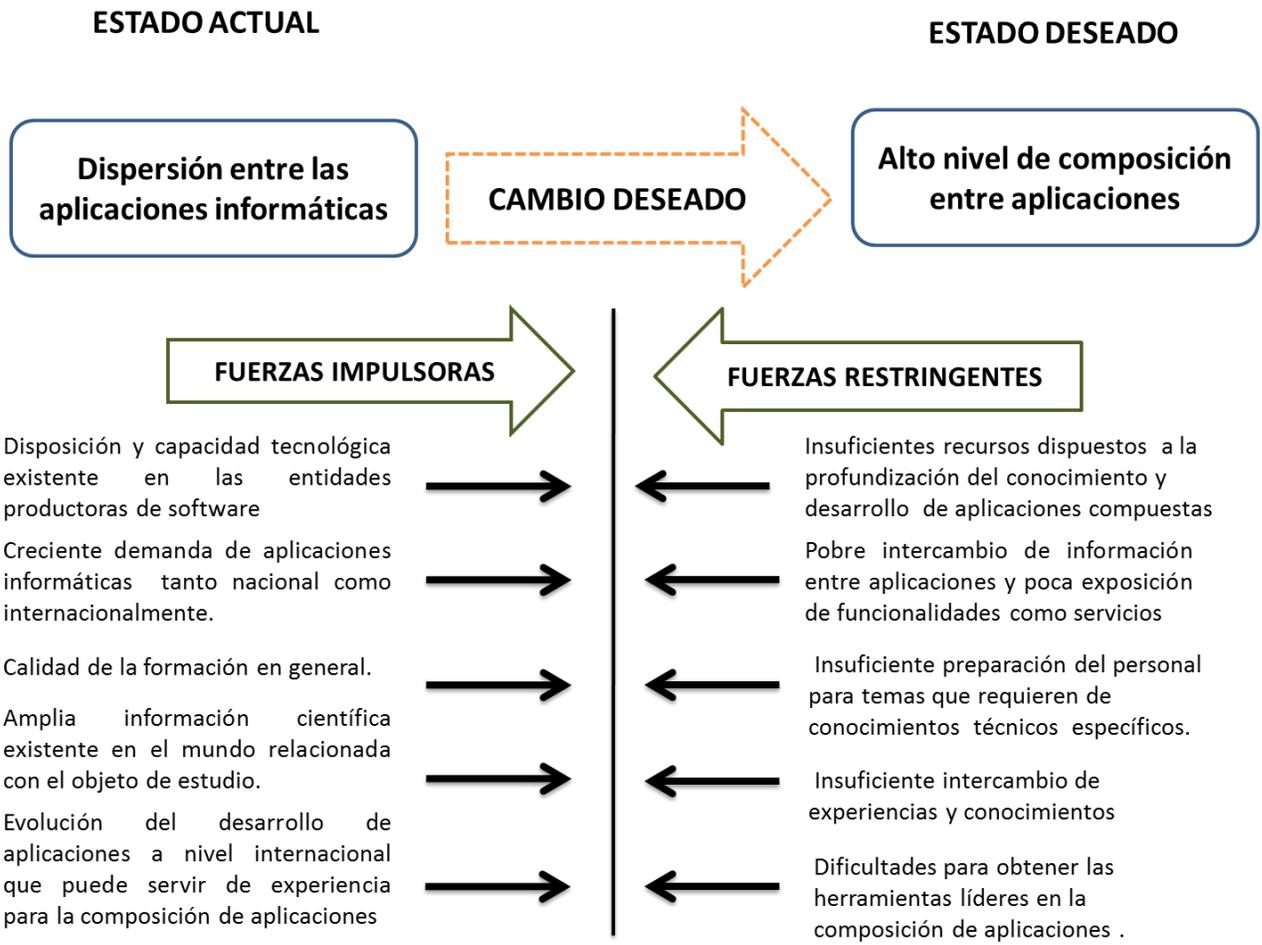
El 60% de los encuestados conoce qué es un *mashup*, el 50% no tienen ningún conocimiento sobre herramientas para desarrollarlos. Solo el 5 % ha desarrollado *mashups* en algún momento. No obstante a ello, el 60% considera que el desarrollo de *mashups* tendría un alto impacto en la producción de aplicaciones y un gran aporte al enriquecimiento de la interacción con los usuarios finales y en el aprovechamiento de las ventajas de la Web 2.0.

El 80% de los encuestados conoce que significa SOA y de ellos el 95% opina que el desarrollo de aplicaciones compuestas es imprescindible en una iniciativa orientada a servicios. Solo el 30 % de los encuestados conoce el lenguaje de orquestación BPEL de orquestación de servicios y de ellos todos reconocen su papel fundamental en el desarrollo de aplicaciones compuestas.

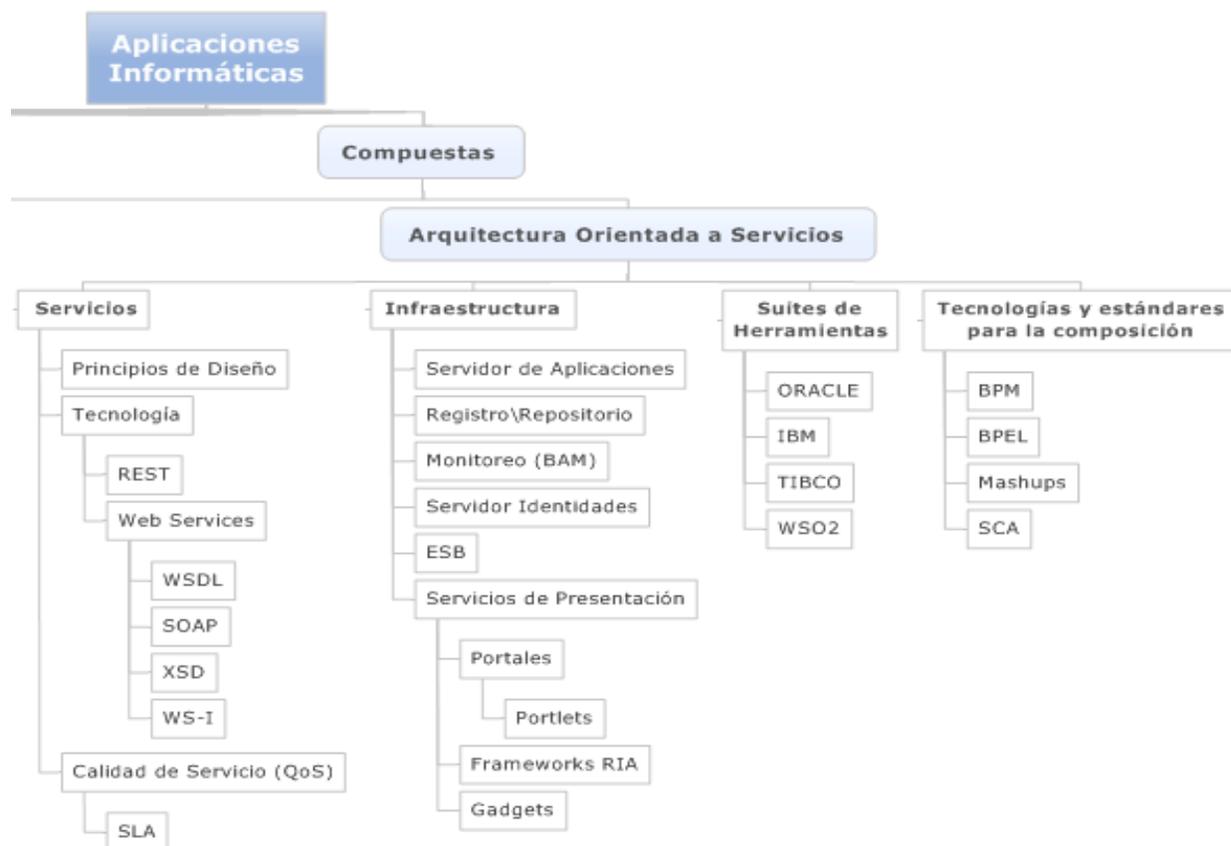
ANEXO 2 Composición de expertos involucrados en el diagnóstico.

Nro	Perfil de los expertos	Cantidad
1	Perfil de Trabajo	
	Desarrolladores	6
	Especialistas en temáticas de la orientación a servicios	2
	Ejecutivos con experiencia en iniciativas SOA	2
	Diseñadores de Software	2
	Total	12
2	Años de experiencia en su actividad	
	Menos de 5 años	1
	Entre 5 y 10 años	10
	Más de 10 años	1
	Total	12
3	Entidad a la que pertenecen	
	UCI	9
	SOFTEL	2
	COPEXTEL	1
	Total	12

ANEXO 3 Determinación de las fuerzas impulsoras y restringentes que inciden en el desarrollo de aplicaciones compuestas.



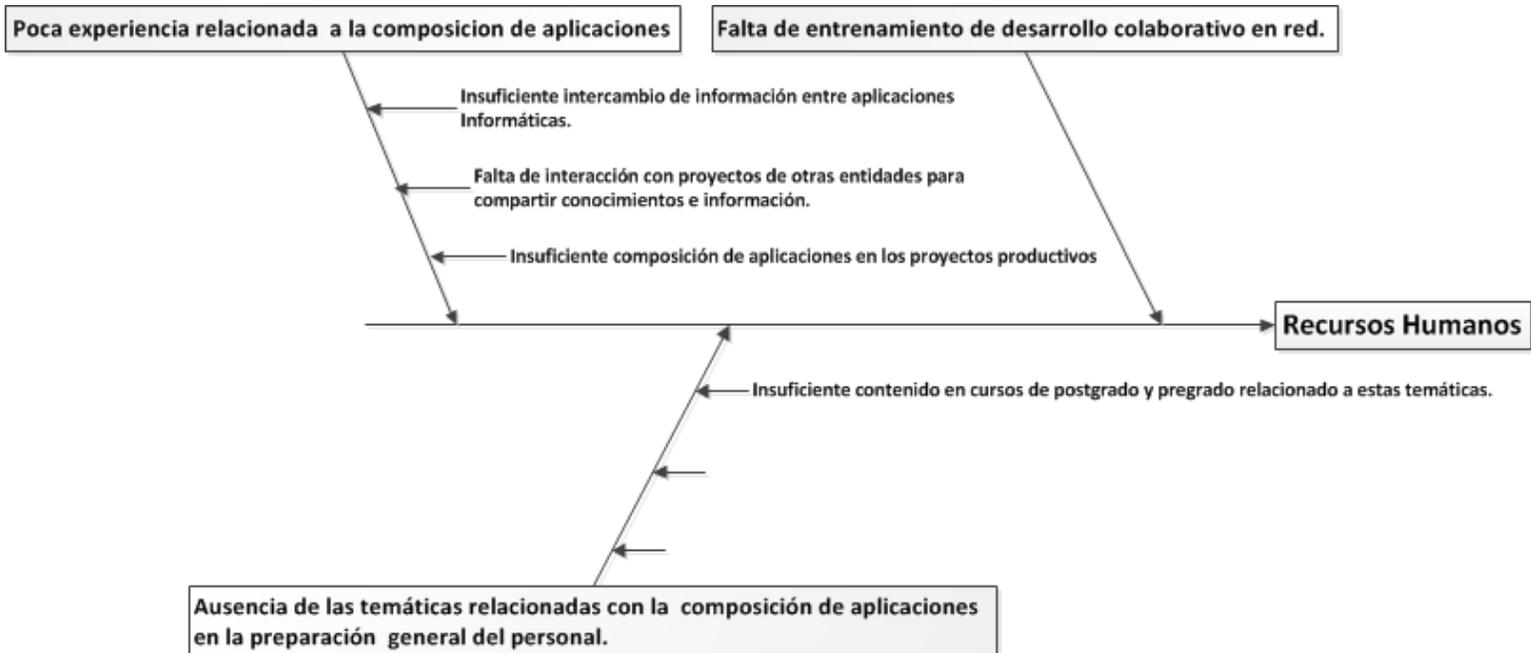
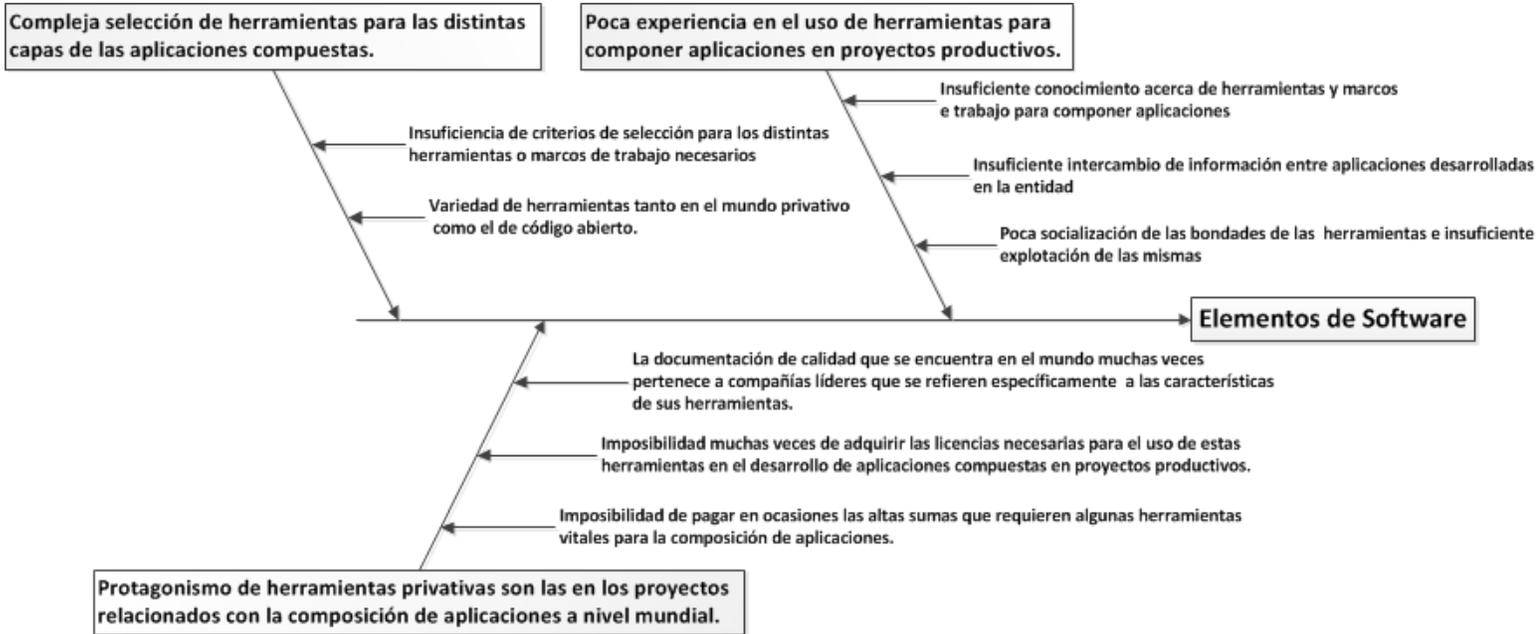
ANEXO 4 Mapa de conocimientos de tecnologías relacionadas con el desarrollo de aplicaciones compuestas.



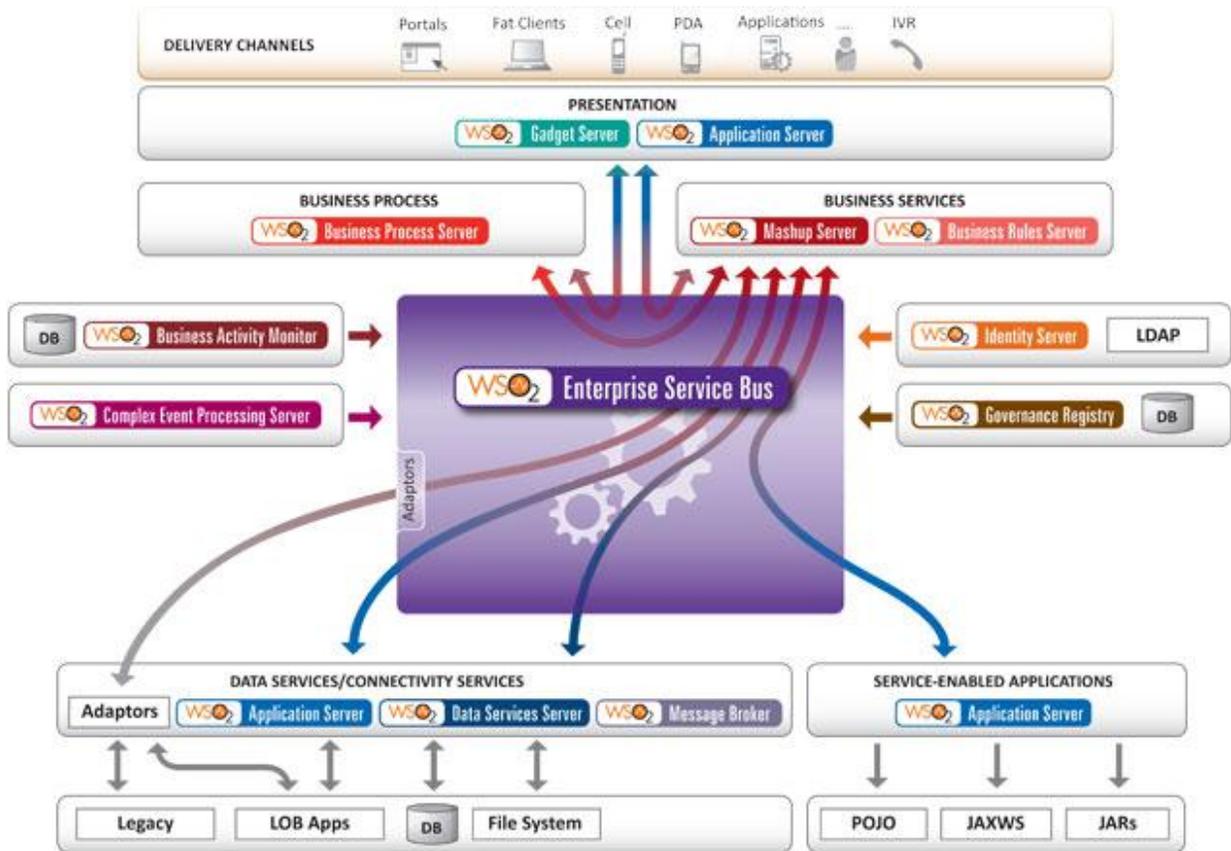
ANEXO 5 Necesidades de aprendizaje para desarrollar aplicaciones compuestas descritas por el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE).

No	Necesidad de aprendizaje	Categoría
1	Integración de Spring Hibernate, inyección de beans, manejo de transacciones.	Junior
2	Dominio de los framework JSF o Vaadin para el desarrollo de la capa de presentación.	Junior
3	Conocimientos de control de versiones en sistemas SVN y uso de subeclipse.	Junior
4	Dominio del trabajo con SGBD, SQL y Modelado conceptual de los datos	Junior
5	Dominio de los conceptos imprescindibles para emprender una iniciativa SOA	Junior
6	Dominio de los lenguajes WSDL, XSLT, XPath	Senior
7	Dominio del framework Axis2.	Senior
8	Dominio del uso de la herramienta WSO2 App Server , BRS , ESB, Mashups Server, BPS y WSO2 Developer Studio	Senior
9	Dominio avanzado de un framework RIA de presentación. Vaadin o JSF	Senior
10	Dominio del diseño y construcción de vistas de presentación.	Junior
11	Dominio del uso de la herramienta Liferay Portal	Senior
12	Dominio del uso de la herramienta WSO2 Gadget Server	Senior

ANEXO 6. Diagramas Causa y Efecto de las variables Elementos de Software y Recursos Humanos



ANEXO 7 Herramientas pertenecientes a la suite WSO2.



ANEXO 8. Interfaces y funcionalidades de la herramienta SI-Holmes

Tabla con Bases de Casos y sus detalles

Bases de conocimiento	Casos	Expertos	Herramientas
Bases Legales	1	0	✓
Bienvenida Holmes	6	0	✓
Goal Directed Therapy	6	2	✓
Salud	3	3	✓
Selección RA	4	6	✓
Selección de Framework RA	13	10	✓

Gestión de valores en determinado rasgo

ADICIONAR CASO

Nombre: Caso 1

Valor(es):

- Crisis aguda de asma bronquial
- Enfermedad Cerebro Vascular Isquémica
- EPOC descompensada
- Enfermedad Cerebro Vascular Hemorrágica
- Enfermedad Cerebro Vascular Isquémica
- Hemorragia Subaracnoidea

Selección del algoritmo para la inferencia

Experto: Experto 1

Algoritmo: HEOM GOWER ARGELIO

Selección del rasgo a inferir

Inferir	Rasgo	Valor(es)	Editar
<input type="checkbox"/>	Edad	20 años ,de 20 a 30 años ,de 30 a 40 años ,de 40 a 50 años ,de 50 a 60 a 70 años ,de 70 a 80 años ,más de 80 años	de 70 a 80 años
<input checked="" type="checkbox"/>	Motivo de ingreso a UCI	No	Infarto del miocardio agudo ,Enfermedad Cerebro Vascular Hemorrágica ,Enfermedad Cerebro Vascular Isquémica ,Hemorragia Subaracnoidea ,Crisis aguda de asma bronquial ,EPOC descompensada ,TEP ,insuficiencia cardíaca ,Shock cardiogénico ,Shock hipovolémico ,Shock séptico ,Trasplante de hígado ,Trasplante renal
<input type="checkbox"/>	Raza	No	Blanca ,Negra ,Amarilla
<input type="checkbox"/>	Sexo	No	Femenino ,Masculino
<input type="checkbox"/>	Talla	No	menor de 1.14 m ,entre 1.40 y 1.50 m ,entre 1.50 y 1.60 m ,entre 1.60 y 1.70 m ,entre 1.70 y 1.80 m ,entre 1.80 y 1.90 m ,entre 1.90 y 2 m ,más de 2m
<input type="checkbox"/>	Temperatura 1h	No	35 ,34 ,36 ,37 ,38 ,39 ,40

Resultado de la inferencia

Experto 1 usando el algoritmo HEOM

Rasgos Inferidos del Caso más Cercano

Rasgo	Valor(es)
Clasificación	Lado del Servidor

Distancia entre los casos más cercanos

Gráfica con la distancia de los 10 casos más cercanos

Distancia Experto-Casos

Casos

- JSF
- GWT
- Vaadin
- Ruby on Rails
- jQuery
- Ext JS
- Silverlight
- Yui 3
- Prototype
- Dojo

ANEXO 9. Procedimiento empleado para determinar el coeficiente de competencia a los candidatos a expertos y resultados obtenidos.

El cálculo de dicho Coeficiente se realiza de la forma siguiente:

$$K_{comp} = \frac{1}{2} (K_c + K_a)$$

donde:

K_{comp}: Coeficiente de competencia.

K_c: Coeficiente de conocimiento: Resulta el promedio de los valores que cada candidato le otorga a cada una de las preguntas, según el conocimiento que considere tenga al respecto.

K_a: Coeficiente de argumentación o fundamentación de los criterios del experto, obtenido como resultado de la suma de los puntos de acuerdo a la siguiente tabla patrón:

Nro	Fuentes de argumentación	Alto (A)	Medio (M)	Bajo(B)
1.1	Estudios teóricos realizados por usted.	0.30	0.20	0.10
1.2	Experiencia adquirida durante su vida profesional.	0.50	0.37	0.30
1.3	Conocimiento de investigaciones y/o publicaciones nacionales e internacionales.	0.05	0.04	0.03
1.4	Conocimiento propio sobre el estado del tema de investigación.	0.05	0.04	0.03
1.5	Actualización en cursos de postgrado, diplomados, maestrías, doctorado, etc.	0.05	0.04	0.03
1.6	Intuición.	0.05	0.03	0.02
	Total	1.00	0.70	0.50

Se plantea entonces que :

La Competencia de experto es de Alta (A): Si $K_{comp} > 0.7$

La Competencia de experto es Media (M): Si $0.5 < K_{comp} \leq 0.7$

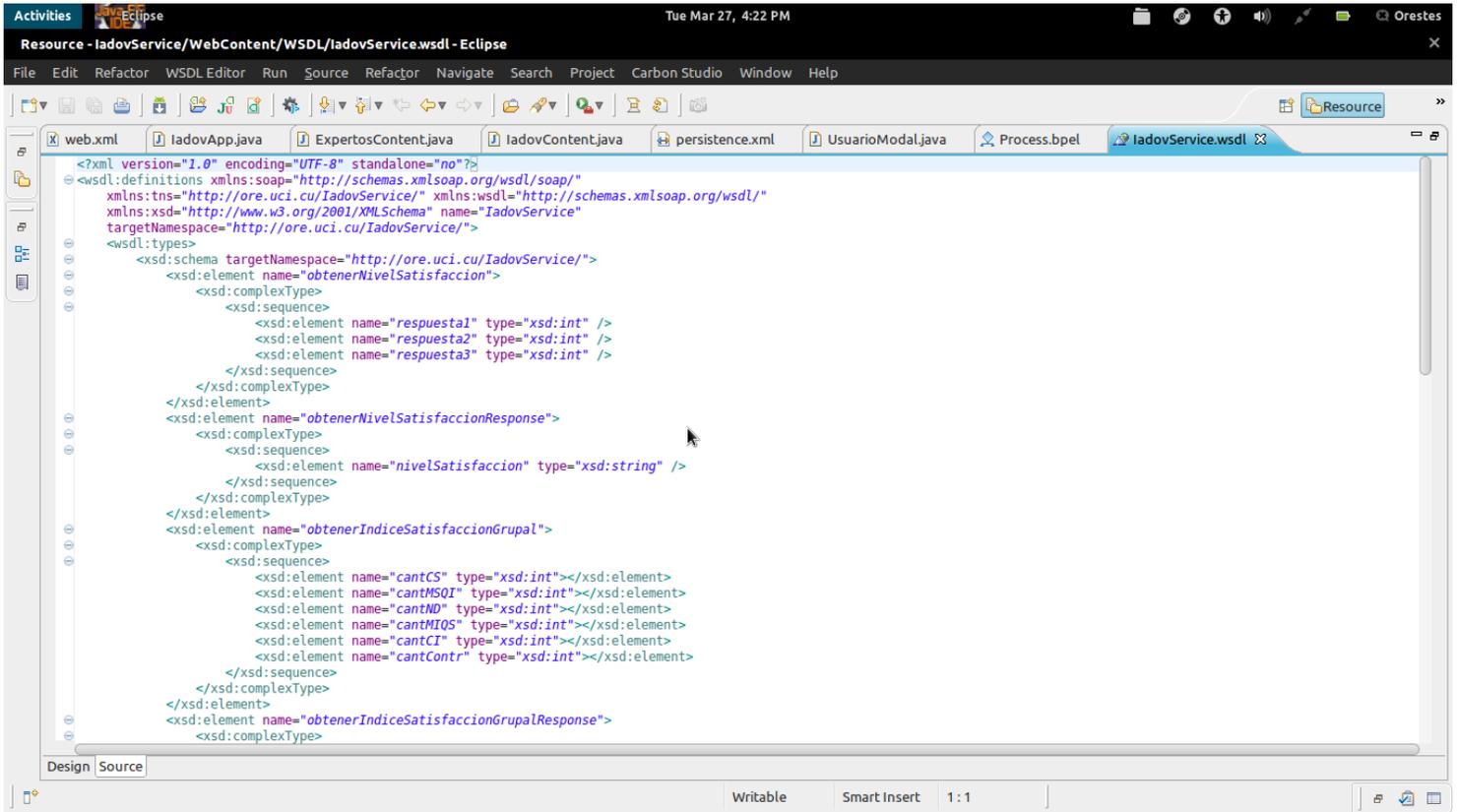
La Competencia de experto es Baja (B): Si $K_{comp} \leq 0.5$

RESULTADOS: Competencia de los 27 expertos que participaron en la validación de MIDAC.

Expertos	Kc	1.1	1.2	1.3	1.4	1.5	1.6	Ka	K	CC
1	0.8	0.20	0.50	0.05	0.05	0.05	0.05	0.90	0.85	ALTO
2	0.6	0.30	0.50	0.05	0.04	0.03	0.02	0.94	0.77	ALTO
3	0.3	0.20	0.30	0.05	0.04	0.05	0.05	0.69	0.50	BAJO
4	1.0	0.20	0.50	0.04	0.04	0.03	0.05	0.86	0.93	ALTO
5	0.5	0.30	0.37	0.05	0.03	0.03	0.05	0.83	0.67	MEDIO
6	0.8	0.20	0.50	0.05	0.05	0.05	0.05	0.90	0.85	ALTO
7	0.8	0.30	0.50	0.04	0.05	0.02	0.02	0.93	0.87	ALTO
8	0.4	0.30	0.37	0.03	0.03	0.02	0.03	0.77	0.59	MEDIO
9	0.5	0.10	0.30	0.03	0.04	0.03	0.02	0.52	0.51	MEDIO
10	0.8	0.30	0.50	0.04	0.05	0.02	0.02	0.93	0.87	ALTO
11	0.9	0.20	0.30	0.05	0.04	0.05	0.05	0.69	0.80	ALTO
12	0.8	0.20	0.50	0.05	0.05	0.05	0.05	0.90	0.85	ALTO
13	0.2	0.30	0.30	0.04	0.05	0.03	0.02	0.74	0.47	BAJO
14	0.8	0.30	0.30	0.05	0.04	0.05	0.02	0.76	0.78	ALTO
15	0.6	0.30	0.50	0.05	0.04	0.03	0.02	0.94	0.77	ALTO
16	0.5	0.20	0.50	0.05	0.05	0.05	0.05	0.90	0.70	MEDIO
17	0.8	0.20	0.50	0.05	0.05	0.05	0.05	0.90	0.85	ALTO
18	0.8	0.30	0.30	0.05	0.04	0.05	0.02	0.76	0.78	ALTO
19	0.9	0.20	0.30	0.05	0.04	0.05	0.05	0.69	0.80	ALTO
20	0.8	0.20	0.50	0.05	0.05	0.05	0.05	0.90	0.85	ALTO
21	1.0	0.20	0.50	0.04	0.04	0.03	0.05	0.86	0.93	ALTO
22	0.5	0.30	0.37	0.05	0.03	0.03	0.05	0.83	0.67	MEDIO
23	0.8	0.20	0.50	0.05	0.05	0.05	0.05	0.90	0.85	ALTO
24	0.5	0.30	0.37	0.05	0.03	0.03	0.05	0.83	0.67	MEDIO
25	0.8	0.20	0.50	0.05	0.05	0.05	0.05	0.90	0.85	ALTO
26	1.0	0.30	0.50	0.05	0.05	0.05	0.05	1.00	1.00	ALTO
27	0.6	0.30	0.50	0.05	0.04	0.03	0.02	0.94	0.77	ALTO

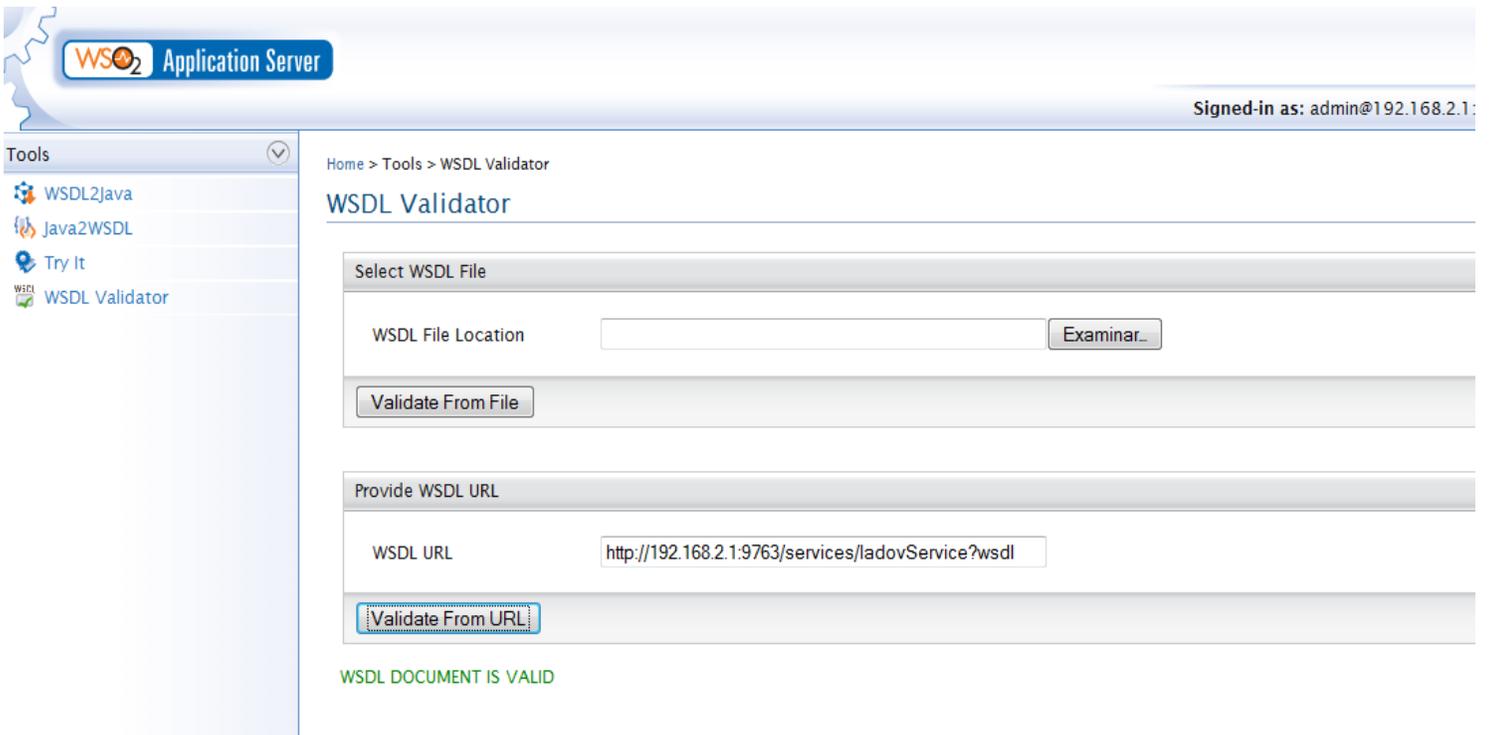
ALTO	19
MEDIO	6
BAJO	2
TOTAL	27

ANEXO 10. Diseño y validación del contrato IadovService.wsdl.



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://ore.uci.cu/IadovService/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="IadovService"
targetNamespace="http://ore.uci.cu/IadovService/">
  <wsdl:types>
    <xsd:schema targetNamespace="http://ore.uci.cu/IadovService/">
      <xsd:element name="obtenerNivelSatisfaccion">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="respuesta1" type="xsd:int" />
            <xsd:element name="respuesta2" type="xsd:int" />
            <xsd:element name="respuesta3" type="xsd:int" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="obtenerNivelSatisfaccionResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="nivelSatisfaccion" type="xsd:string" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="obtenerIndiceSatisfaccionGrupal">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="cantCS" type="xsd:int"/></xsd:element>
            <xsd:element name="cantMSQT" type="xsd:int"/></xsd:element>
            <xsd:element name="cantND" type="xsd:int"/></xsd:element>
            <xsd:element name="cantMIQS" type="xsd:int"/></xsd:element>
            <xsd:element name="cantCI" type="xsd:int"/></xsd:element>
            <xsd:element name="cantContr" type="xsd:int"/></xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="obtenerIndiceSatisfaccionGrupalResponse">
        <xsd:complexType>

```



WSO2 Application Server

Signed-in as: admin@192.168.2.1:

Tools

- WSDL2Java
- Java2WSDL
- Try It
- WSDL Validator

Home > Tools > WSDL Validator

WSDL Validator

Select WSDL File

WSDL File Location

Provide WSDL URL

WSDL URL

WSDL DOCUMENT IS VALID

ANEXO 12. Cuestionario para evaluación de la satisfacción del usuario con respecto a MIDAC.

MIDAC: MODELO PARA EL DESARROLLO DE APLICACIONES COMPUESTAS BASADO EN SOA						
Entidad:						
Área:						
Datos del Encuestado						
Nombre y Apellidos:						
Cargo o Rol:						
Nivel Escolar:		Técnico Medio <input type="checkbox"/>	Universitario <input type="checkbox"/>			
Categoría Docente:		Instructor <input type="checkbox"/>	Asistente <input type="checkbox"/>	Auxiliar <input type="checkbox"/>	Titular <input type="checkbox"/>	
Categoría Científica:		Especialista <input type="checkbox"/>	Máster <input type="checkbox"/>	Doctor <input type="checkbox"/>		
Años de experiencia:						
#	Afirmaciones	Respuesta				
1	El proceso de selección tecnológica que forma parte del modelo favorece la composición correcta de aplicaciones.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
2	El uso en el modelo de inventarios de servicios para la gestión y localización de recursos es fundamental para la composición de aplicaciones.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
3	El análisis automático de experiencias anteriores como principio de retroalimentación constante es decisivo para el desarrollo de aplicaciones informáticas compuestas.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
4	El paradigma de razonamiento basado en casos incluido en el modelo es un excelente recurso para seleccionar información de manera inteligente.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
5	El conjunto de competencias que se enuncian en el modelo contribuyen a simplificar la complejidad técnica en el desarrollo de este tipo de aplicaciones.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
6	La selección correcta de un marco de trabajo RIA, en el componente de gestión tecnológica del modelo, permite a los servicios de presentación enriquecer el valor de las aplicaciones compuestas.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
7	El chequeo de la aplicación de los principios de diseño de SOA en la Gestión de Recursos es fundamental para la obtención de servicios interoperables, reusables e independientes de la tecnología.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
8	La concepción de una infraestructura tecnológica correspondiente al entorno cubano realizada para lograr el funcionamiento del modelo es fundamental para el desarrollo de cualquier proyecto informático.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
9	Los componentes que forman parte del modelo y su interrelación brindan la integralidad necesaria para el desarrollo de aplicaciones compuestas.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>
10	Los principios en los que se basa el modelo facilitan el proceso de composición de aplicaciones en iniciativas orientadas a servicios.	MA <input type="checkbox"/>	NUNO <input type="checkbox"/>	CD <input type="checkbox"/>	DA <input type="checkbox"/>	ED <input type="checkbox"/>

ANEXO 12. Cuestionario para evaluación de la satisfacción del usuario con respecto a MIDAC.

EVALUACION DE LA PROPUESTA DEL MODELO DE COMPOSICIÓN DE APLICACIONES BASADO EN SOA				
Entidad:				
País:				
Datos del Encuestado				
Cargo o Rol:				
Nivel Escolar:		Técnico Medio ___	Universitario	
Categoría Docente:		Instructor ___	Asistente ___	Auxiliar ___ Titular ___
Categoría Científica:		Especialista ___	Máster ___	Doctor ___
Años de experiencia:				
No	Preguntas	Respuestas		
1	¿Cree usted que los componentes escogidos son prioritarios para el proceso de composición de aplicaciones?	Sí _ No __		
2	¿Ha participado en el desarrollo de aplicaciones compuestas de algún tipo?	Sí _ No __		
3	¿Conoce usted algún proyecto donde se compongan aplicaciones?	Sí _ No __		
4	¿Considera usted que es aconsejable componer aplicaciones en un entorno SOA sin un modelo coherente?	Sí _ No __		
5	¿Considera que la utilización de este modelo estimulará la reusabilidad de funcionalidades en una organización?	Sí _ No __		
6	¿Le satisface la representación gráfica de este modelo para la composición de aplicaciones?	Me gusta mucho ___ No me gusta tanto ___ Me da lo mismo ___ Me disgusta más de lo que me gusta ___ No me gusta nada ___ No sé qué decir ___		
7	¿Qué relevancia le concede usted a la vinculación del proceso de composición de aplicaciones con una herramienta de selección inteligente de información?	Entre 0 y 10 _____		
8	¿Qué impacto cree usted que produciría el uso de un modelo para componer aplicaciones en organizaciones que desarrollen proyectos con iniciativas orientadas a servicios?	Entre 0 y 10 _____		
9	¿Cree usted que las aplicaciones compuestas puedan aportar en cuanto a productividad a los usuarios finales?	Entre 0 y 10 _____		

10.- ¿Qué elementos considera positivos de este modelo?

11.- ¿Qué elementos considera negativos de este modelo?

12.- ¿Qué sugerencias tienen para el desarrollo e implantación de este modelo?

ANEXO 13. Grupo Focal. Guía de desarrollo.

No de participantes: 9 participantes compuesto por investigadores, especialistas del departamento Soluciones SOA del Centro de Desarrollo de Arquitecturas Empresariales (CDAE), doctores en ciencias técnicas, másteres y desarrolladores de servicios que utilizan estándares y tecnologías de referencia vinculadas al paradigma de la orientación a servicios.

Fecha: 17 de enero del 2010

Lugar: Salón 301

Hora: 10:a.m

Apertura

- Describir lo que constituye un grupo focal
- Explicar el objetivo de la reunión
- Explicar procedimiento

Presentación de MIDAC

1. Objetivos

Objetivos de la Investigación
Elaborar un modelo basado en la integración de herramientas, buenas prácticas y marcos de trabajo para el desarrollo de aplicaciones compuestas en iniciativas orientadas a servicios que contribuya a la interoperabilidad y la estandarización del proceso.
Objetivos del grupo focal
Comprobar la apreciación y el criterio que tiene un grupo de especialistas en el objeto de estudio con respecto al Modelo Integrado para el Desarrollo de Aplicaciones Compuestas (MIDAC).

Nombre del moderador: Ing. Orestes Febles Díaz

Nombre del observador: Dra. Ailyn Febles Estrada

Guía de preguntas

- De las tendencias teóricas actuales en la adopción de SOA y el desarrollo de aplicaciones compuestas. ¿Qué elementos consideran que deberían incorporarse en MIDAC?
- ¿Considera que la infraestructura contienen los elementos necesarios para soportar tecnológicamente el proceso de desarrollo de aplicaciones compuestas?
- ¿Considera que el razonamiento basado en casos puede ser factible para el análisis de las mejores experiencias en con el desarrollo de aplicaciones compuestas?
- ¿Cree que los estándares usados por las aplicaciones compuestas basadas en MIDAC son los más reconocidos y aplicados a nivel mundial en entornos orientados a servicios?
- ¿Sería necesario un grupo de indicadores para seleccionar marcos de trabajo para desarrollar interfaces enriquecidas?
- ¿Considera que un Registro/Repositorio es un elemento clave que facilita el intercambio de información entre los consumidores y proveedores de servicios?

ANEXO 14. Resultados de la triangulación metodológica.

Objetivo a evaluar	Métodos cuantitativos	Método cualitativos	Conclusión
<p>Desarrollo de un modelo que brinde interoperabilidad y estandarización al desarrollo de aplicaciones compuestas</p>	<p>IADOV, para comprobar el nivel de satisfacción de los usuarios (ISG = 0,703)</p> <p>El escalamiento de Likert para comprobar la aplicabilidad de los principios del modelo y su funcionamiento.</p> <p>(IP = 80 en cada una de las preguntas)</p>	<p>Grupo focal para evaluar las características de los componentes del modelo y la funcionalidad de la infraestructura tecnológica seleccionada para su implementación.</p> <ul style="list-style-type: none"> • Aprobación mayoritaria de las características y funcionalidades del modelo. • La propuesta de competencias básicas para los desarrolladores de aplicaciones compuestas fue aceptado favorablemente <p>Método de comparación</p> <ul style="list-style-type: none"> • Se destaca el aporte del modelo en cuento al mejoramiento del nivel de interoperabilidad de los recursos. 	<p>Según los resultados tanto cuantitativos como cuantitativos se puede concluir que existe una correspondencia satisfactoria entre los resultados de la variable evaluada que:</p> <ul style="list-style-type: none"> • Evidencia tanto el nivel de satisfacción de los usuarios como de aprobación de los componentes e infraestructura tecnológica del modelo los cuales promueven el uso de estándares y mejoran el intercambio de información entre aplicaciones. • Se comprueba que es provechoso utilizar el modelo en entidades desarrolladoras de software que adopten el paradigma orientado a servicios. <p>Por tanto, se comprueba la validez de la propuesta del modelo para el desarrollo de aplicaciones compuestas basado en SOA que contribuye favorablemente a la estandarización y la interoperabilidad.</p>

ANEXO 15 Composición de los expertos involucrados en la validación

Nro	Perfil de los expertos	Cantidad
1	Perfil de Trabajo	
	Profesores	7
	Desarrolladores	12
	Especialistas en la temática	8
	Profesionales con labor científica	5
	Total	32
2	Categoría científica	
	Doctores	6
	Másters	15
	Ninguna	11
	Total	32
3	Instituciones	
	Universidad “Mariano Gálvez”	5
	Universidad Abierta para Adultos (UAPA)	4
	Universidad Central del Este (UCE)	7
	ORACLE	2
	UCI	9
	SOFTTEL	3
	COPEXTEL	2
	Total	32

ANEXO 16. Interfaces y funcionalidades de la aplicación compuesta desarrollada para automatizar la validación.

Gráfica del ISG como resultado de la aplicación de la Técnica de ladov

Gestión de expertos para la validación

The screenshot displays two main components. On the left, a pie chart titled 'Satisfacción del Usuario' shows the results of a user satisfaction survey. The chart is divided into five segments: 'Clara satisfacción: 20' (blue), 'Más insatisfecho que insatisfecho: 6' (red), 'No definida: 4' (green), 'Contradictoria: 1' (purple), and 'Más satisfecho que insatisfecho: 1' (orange). The overall ISG score is 0.703125. Below the chart is a legend with Likert scale options: 'No', 'Sí', 'No me gusta tanto', 'Me gusta mucho', 'No me gusta nada', and 'Clara satisfacción'. On the right, a form titled 'Yausel Marme Ruiz' allows for the management of experts. It includes fields for 'Nombre', 'Entidad', 'Área', 'Cargo', 'Experiencia', 'Correo Electrónico', 'Teléfono', 'Nivel Escolar', 'Categoría Docente', and 'Categoría Científica'. A list of experts is visible in the background, including names like Luis Enrique Vega, Jorge Infante Osorio, and Yausel Marme Ruiz.

Gestión de los cuestionarios realizados para aplicar el escalamiento de Likert

The screenshot shows a questionnaire interface. On the left, a navigation menu includes 'Diagnóstico', 'Encuestas', and 'Administración'. The 'Encuestas' section is active, showing a list of questionnaires. The main area displays a questionnaire with 10 items, each followed by a Likert scale dropdown menu. The dropdown menu is currently open, showing options: 'Muy de acuerdo', 'De acuerdo', 'Ni una cosa ni la otra', 'En desacuerdo', and 'Muy en desacuerdo'. Below the questionnaire, there is a 'Registrar' button and a table showing the responses for each question. The table has five columns labeled 'RESPUESTA 4', 'RESPUESTA 5', 'RESPUESTA 2', 'RESPUESTA 3', and 'RESPUESTA 1'. The rows show the selected response for each question, such as 'Muy de acuerdo' or 'Muy en desacuerdo'.

RESPUESTA 4	RESPUESTA 5	RESPUESTA 2	RESPUESTA 3	RESPUESTA 1
Muy de acuerdo				
Muy en desacuerdo				
De acuerdo				
Ni una cosa ni la otra				
Muy de acuerdo	De acuerdo	Muy de acuerdo	Muy de acuerdo	Muy de acuerdo
De acuerdo				
Muy de acuerdo				
Muy de acuerdo				

ANEXO 17. Gestión, despliegue y estadísticas de los servicios por el servidor de aplicaciones propuesto para MIDAC.

Tools

- JAX-WS/JAX-RS
 - List
 - Add
- Axis2 Services
 - List
 - Add
 - AAR Service
 - Jar Service
 - Spring Service
 - EJB Service
 - Data Service
 - Generate
 - Create
 - Upload
- Modules
 - List
 - Add
- Applications
 - List
 - Add
- Repositories

Service Type: ALL Service:

Select all in this page | Select none [Delete](#)

Services							
<input type="checkbox"/>	echo	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service	Download
<input type="checkbox"/>	HelloService	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service	Download
<input type="checkbox"/>	ladovDataService	data_service	Unsecured	WSDL1.1	WSDL2.0	Try this service	
<input type="checkbox"/>	ladovService	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service	Download
<input type="checkbox"/>	Saludo	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service	Download
<input type="checkbox"/>	Version	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service	Download
<input type="checkbox"/>	wso2carbon-sts	sts	Unsecured	WSDL1.1	WSDL2.0	Try this service	Download

Select all in this page | Select none [Delete](#)

Estándares de contrato de servicio

Quality of Service Configuration

✔ Active [[Deactivate](#)]

Security	Policies
Reliable Messaging	Transports
Response Caching	Modules
Access Throttling	Operations
MTOM Optional	Parameters

Specific Configuration

[Edit Data Service \(Wizard\)](#)

[Edit Data Service \(XML Edit\)](#)

Statistics

Request Count	84
Response Count	80
Fault Count	2
Maximum Response Time	235 ms
Minimum Response Time	< 1.00 ms
Average Response Time	9.857 ms

Estadísticas de requisitos no funcionales de los servicios

Average Response Time(ms) vs. Time(Units)